State-of-The-Art Worldwide Widespread ERP-Borne Misuse of Data

Haekyung Rhee[1]

[1]*Associate Professor, Department of Computer Games, Yongin-Songdam University*

Abstract: Anything cannot be simply qualified as data, since that sort of absent-minded design attitude can enormously increase the degree of data obesity in corporate. However, as ERP-based information systems still prevail in most enterprises, data is created in a fashion of adopting the design patterns in ERP, although someone who attempts to have his/her own data model apart from ERP tends to copy those ERP-inherent design patterns. In this paper, we take a thought-to-be one of the best practice in real-world corporate data modeling and then we scrutinize its value fairy theoretically on basis upon disciplines and norms associated with legitimate data modeling. The result is astounding in that nearly half of data in ERP-based model turn out to be valueless and therefore they do not deserve to be included in the data part of information system, rather they are strictly recommended to relocate to the its program code part.What more appall us is the design outcomes of ERP is fetched much far from so called the standard normal forms in database community. They are classified as simple file structures rather than relational data table structures. We then show the method to rectify the anomaly inherent in ERP-based model so that every data model automatically abides with the design outcome of at least the third normal form. Such refinement leads us to have data models optimized in term of data redundancy by kicking out unnecessary data attribute replication and useless attributes. The result revealed that 40 percent of redundancy in average can be removed.

Keywords: Data Redundancy, Data Obesity, ERData Model, Third Normal Form, Data Misuse

1. INTRODUCTION

1.1 Background

As more information systems are forged and introduced in the world of increasing digitization, we hear a lot of outcry of complaints with regard to the retardation of system response and the quality of data answered from the system. In this essay, we explore why response speed and data quality are compromised in a way of demoting overall system performance. For this type of investigation, we searched a data model which is thought to be the one of the best practice in real world information systems. Searching such a data model took nearly a decade as many enterprises have never pleased to open their enterprise-wide data model for the purpose of academic study. After finally finding out such a data model, we validated that model to check its speed and quality. Quantitative analysis rather than qualitative approach has been attempted in this process.

The significance of evaluating the speed and quality of a given data model is more than crucial as its speed and quality determines the performance of the entire information system in which the data model is contained, since it is not just a tiny subset of entire data models of the information system but the overall and basic philosophy of data modeling is represented by that specific data model. Although the issue of evaluating data model is profoundly important, attempting that kind of evaluation was not easy as devising approaches to deal with the evaluation seems infeasible. But in this essay, we were very successful in carrying out this analysis.

1.2 Motivation

Although any information system should be composed of data and program, clear division and distinction between and programs appears to be very difficult as it is very often observed that a variable to be coded in program part is treated as data, and thus subsequently mistakenly included as a datum in database part. For example, *Verification Flag*, which means whether something such as approval has been verified or not, is intrinsically not a datum and therefore should not be allowed to be data, since its value can be ambiguous. The value of data must be numeric for the understanding of nature of data. In case the value cannot be numeric, the form of a value of data turn out to be a type of *Yes* or *No* or a long string of description. Then in this case the data must be expressed in the program part as a variable or parameter rather than as a datum. If it intends to be a genuine datum it should have taken a form of *Date_of_Verification*. With this data, checking to see whether something has been verified or not in the program part will be easy. In this case, it is evident that once the date of verification has been adopted as data there is noneed to forge a superficial data like *Verification Flag*.

What we mean by superficial is that the data was not supposed to be treated as data. In other words, it is not a real and genuine data. In another way, we can designate such a data as non-real or fake data in a way of expressing that it is essentially not born to be a datum. So in this paper, superficial data means a datum that is very much

dependent upon some real data and at the same time should be dealt with in the program part but is obviously forged artificially for the only sake of probably convenience.

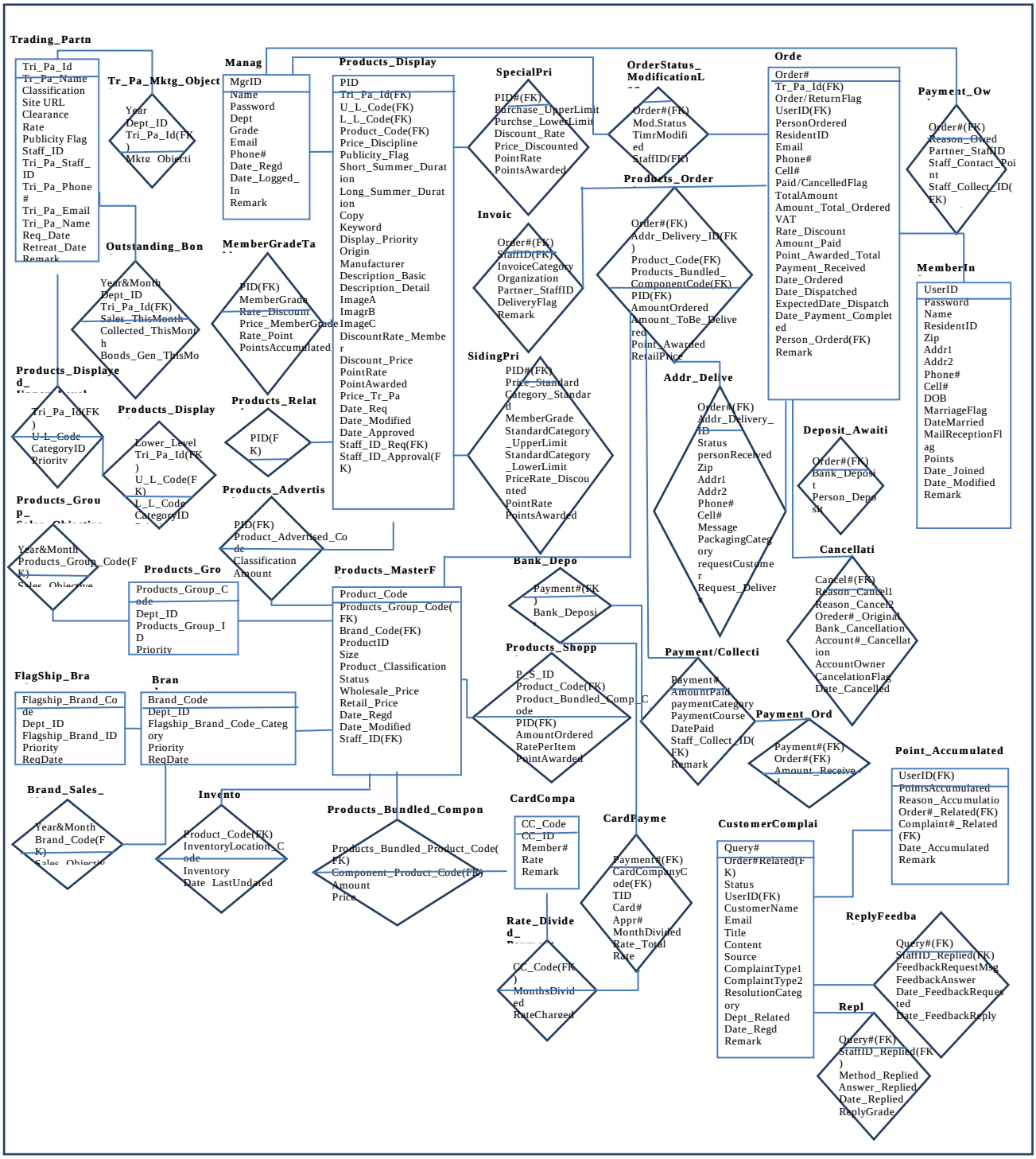

Fig. 1. Entity-Relationship Model for Electronic Commerce Applications

The phenomenon of permitting the existence of such superficial data in the database becomes so prevalent in real corporate world. Careless use of superficial data leads the degree of data redundancy unexpectedly increased when it is difficult to particularly identify the origin of data. Any data should have its origin and the real origin of data is where the data is born as data as its first appearance. Once the data appeared in any places in the database other than its origin, it is clearly a replica and in this occasion the data should obey the rule of replicating itself. Namely, replication of data is only allowed when the data was originally and basically a primary-key attribute data. Such a replicated data plays a role of foreign-key attribute data and the fulfillment of such a role should be strictly

observed within the rule of adjacency of the location of replicated data and the location of its original data. According to the basic and mandatory rule with regard to allowance of this adjacency in the entity-relationship model, there must be a direct connection from the data component that holds the original data towards the data component that contains its replica. Any indirect connection via more than one link is in principle illegitimate and therefore illegal, since such careless connections would lead to dataredundancy, which is absolutely unnecessary.

But it is quite astonishing to note that in real corporate world about half of data attributes in the entire database part is unfortunately actually turn outto be illegitimate or illegal. Let me show one typical example of such abnormal data design. During my experience of teaching student in the subject of database design for the past two decades, one time Iapproached by some experienced student and what himself as one of database design experts. The Fig. 1 is the entity-relationship model for some electronic commercial website the student provided for the analysis of its quality.

1.3 Objectives

We attempted to investigate the quality of the data model in Fig. 1 by figuring out how much it is valid and legitimate on basis of the philosophy of ER model [1]. The exact degree of validity will be measured through the degree of unnecessary data redundancy and the degree of meaningless or useless replication. What we mean by data in micro level it is data attribute at the very finest level rather than macro-level entity or relationship. So we will compute the degree of data redundancy at the level of data attribute. This is because from the legacy and history of data modeling theories it is absolutely crucial to abide with the principle of minimizing the amount of data replication.

As we can see from the comparison between say the third normal form and Boyce-Codd normal form [1], the advent of BCNF is accrued by the idea of reducing the degree of data redundancy in 3NF by extraditing those portion of data that cause apparent repetition of a certain attribute. Therefore, the basic attitude towards what type of normal from we are going to pursue is that we take BCNF as the better than 3NF in the sense that more optimization has been made in BCNF and less optimization done in 3NF.

We presented our every analysis quite quantitatively, for example to the point where exact percentage is provided. For this, we counted the number of unnecessary attributes and unique attributes as well as the total number of attributes in Fig. 1. We then revealed how it looks like when all the unnecessary and undeserved data, regardless of whether they are one of entities or relationships or even attributes. Abnormality of that such an outcome was analyzed then, and we will investigate if there is a possibility and way to rectify the deficiency found in Fig. 1.

II. RELATED WORK

There have been a number of researches on how to extract or classify data objects into entity, relationship or attribute, but there has been very few researches on data obesity[2]. However, it is worthwhile to review the methodologies with regard to the classification of data objects as they deal with the issue of data redundancy to some extent.

2.1 Approaches with Imprecise Job Descriptions

A methodology of entity extraction from job requirement descriptions [3, 4] is the one of the showcases in which data objects such as entities or attributes are excavated from unstructured text type of job descriptions. Under the assumptions that there will inevitably be inconsistency in describing data objects and imperfection with regard to trying to not miss any parts of descriptions on job process behaviors in the entire job descriptions, this study requires users who had written down the job descriptions to respond a slew of questions raised for the purpose of getting rid of or clarifying obscurity associated with ambiguities in the entire descriptions. However, in this approach semantics and domain knowledge have not been utilized as much as possible for the purpose of minimizing the degree of data redundancy.

2.2Approaches with Precise Job Descriptions

Since achieving a perfect accuracy in job descriptions by repeatedly modifying them in a way of going through numerous interactions between the automated system and users, initiated absolutely by the systems side, may be tedious and cumbersome, there has been an attempt focused mainly on a transformation of job descriptions into a conceptual model in a fully automated way by avoiding any efforts for improvement of job descriptions [5, 6].

III. ANALYSIS OF DATA MODEL USED FOR ELECTRONIC COMMERCE APPLICATIONS

From the birds-eye point of view, it looks proper as a conceptual data model and therefore it seems to be within the framework of the normal ER model, but there are a lot of arguments that makes this type of model far from the normal model. The basic and strict rule in manufacturing ER model is that every relationship should have two entities, one at its left and the other at its right. It is

astonishing that there are frequent violations to this basic norm. First of all, there are two cases of directly connecting a relationship to another relationship, which forces a relationship to be a relationship between another relationship and a certain entity. This sort of representation is not only illegal but also misleading in interpreting the meaning of the relationship.

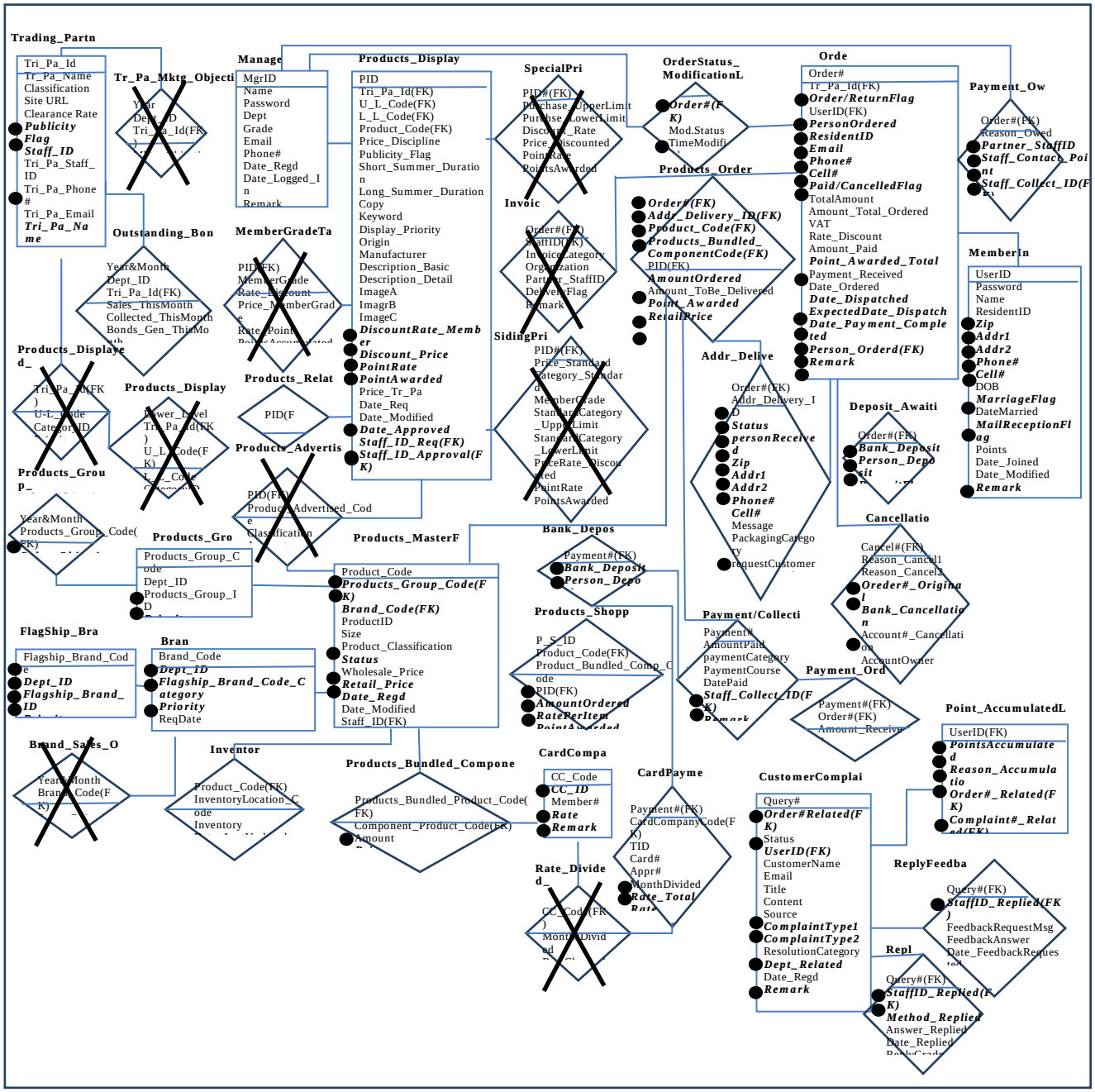

Fig.2. Electronic Commerce Data Model with Redundant and Unnecessary Attributes Removed

3.1 Confusion between Entities and Relationships

There are two cases of treating entity as relationship. For example, entity Products is depicted as relationship. There are 10 cases of treating a simple attribute as entity. For example, dataAddress*_for_Delivery* must be an attribute but it is depicted as entity. There are five cases of treating relationship as entities. For example, an action *Award_BonusPoint* should be treated as relationship but it was depicted as entity.

3.2 Problem of Key Attributes

It is mandatory that once a foreign key is introduced in any relationship it should appear in the key attribute field so that the foreign key is a part of the entire primary key of the relationship. But there are 23 cases of foreign key attributes appearing in the non-key portion of either entities or relationships. Another problem adversely contributing to the data redundancy ratio is whether-or-not type of flag. There are 4 cases of flag-type attribute in the ER model in Fig. 1.

One of the most illegitimate and serious form of violation is, as readers can clearly check in Fig. 1.that there are cases in which entities or relationships having no primary keys at all in them. In case such a design is unwittingly introduced, the resultant data tables will never belong even to the first normal form category.Null or void attributes in key field clearly vindicated that the data model is itself a backlog type of conventional file structure rather than relationship-oriented or behavior-oriented data model.

3.3 Alterations Made

For readers to have a quick glance with regard to the degree of violations occurred in Fig. 1, let us have a close look at the modifications taken in Fig.2. The attributes dotted and the boxes marked 'X' are judged to be illegitimate or illegal on basis of the norms that should be observed in ER model. So, roughly half of the entire data design in Fig. 1 is observed to be just obsolete. The examples in Fig. 1 and Fig.2 clearly shows the sour reality that about half of the actual data design in real world done by real-world workforce just fall outside the norm of discipline of ER model. It is also quite surprising to notice that there are 44 cases of violations in this tiny sort of small-scale example of electronic commerce data design.

3.4 Problem of Candidacy for Data

Anything we wish to grant a certain object as datum is many occasions not really a datum at all. When it comes to see data at the level of attribute, it must possess a certain numerical value in it as its exact and meaningful value. For instance, *Score* should contain, say 95, as its value.What is contained as value ought to be some numerical value in a way of possibly sorting the values of attributes in ascending or descending order for the purpose of achieving a quick search for a certain value, away from descriptions of string typeof any length depicting some particular point of score. Otherwise, searches will take enormous amount of time as no sorting algorithms can never be applicable. In this sense, in case the value of some attribute cannot be identified to be a numeric, it is considered to fail as a candidate datum. Those data that cannot possess numerical values therefore should be treated and processed within the boundary of computer application program codes as certain variables or parameters.

3.5 Overall Evaluation: 55% Redundant

The case of dots marked in front of attributes denotes unnecessary replications due to a reason of either syntax or semantics. As they can individually be seen and counted in Fig.2, there are 85 cases of invalid misrepresentation of attributes including those ones replicated. In sum, in terms of attribute data redundancy, there are 150 cases of replication, either syntactically or semantically. As there are 280 attributes in total in the ER model in Fig. 1, the redundancy ratio is 150/280, which is just nearly 55 percent.

3.6 Lessons Learned from Abnormal Data Model Design Practices

The loss or lack of key attributes in the primary key field of entities or relationship leads us to aware of the stance and nature of ER model its designer has taken. Therefore, the final conclusion about the quality of data design in the ER model in Fig. 1 is that it is not actually ER model, although it took all thenotations and linkage between all the components in the model for the purpose of formally displaying data paths among the components. So, the Fig. 1 cannot be deemed to be an ER model. It may be judged to be a file structure in which a lot of data attribute replication is allowed and a file structure that intentionally resembles the shape of ER model. If we omit all the illegitimate part of design in Fig. 2, the resultant data model looks like the one in which there are abundant array of solely unary relationships.

IV. ODDS-AT-WORK FOUND IN DATA MODEL AFTER REPEALING ALL THE ABNORMALITIES

4.1 Prevalence of Unary Relationships

It is very astonishing to note that every relationship in the resultant data diagram is actually absolutely all unary. Out of 18 relationships, we can find none of binary relationship, which is the very fundamental and basic in devising ER models. We should as well notice that in principle unary relationship is only specially permitted within the same entity. For instance, relationship Manage can be made unary as there are some employee *A* that manages some other employees and at the same time *A* can be managed by some other employee within an enterprise. Therefore, unary relationship is very special self-reflexive one in that it specifies relation between entity instances within the same entity type.

4.2Binary Behavior-ignoring Data Model

Among the 18 relationships in Fig. 2, however, unfortunately none falls within the category of self-reflexivity. All of them should have been designed as binary relationships instead, some of them probably extended to ternary ones of course, since the existence of every relationship is meaningful only between two or three different separate entities. So this violation is very critical in that the model in Fig. 2, actually meaning Fig. 1, definitely 100 percent invalidates the notion and

philosophy behind the ER approach.

V. WHERE DOES THIS FORM OF ILLEGITIMATE DESIGN ACTUALLY ORIGINATED FROM?

Since the data model in Fig. 2 is exceedingly and outrageously far from the classical ER approach, questions subsequently arise where this type of abnormal data design does accrue from.

5.1 Origin of Misuse of Data

We tried to find out previous data models similar to the one in Fig. 1, but it was not easy as that model does seemingly contain some form of resemblance to binary relationship, which eventually turned out to be superficial one. So, we tried to find out previous data models in the literature very similar to the one in the resultant data diagramwith all the illegitimate ones totally excluded instead. And this time we were able to retrieve a bunch of such examples from various websites [7, 8]. Readers can follow the *url* links inthe Reference to see a lot of images through Internet searches.

5.2 Methodology for Rectification

5.2.1 Devising Relationships First

If we stick to the fundamentals of ER approach, it isfairly straightforward to devise binary or ternarybehavior-oriented ER data models in a way of first of all identifying such relationships more than anything else at the very beginning of data modeling.

5.2.2 Devising Connections to Entities

Then we need to check whether there is some entity missing for any relationship to its immediate left andto its immediate right. This is very simple but itguarantees the authenticity and perfectness of ER modeling**.**

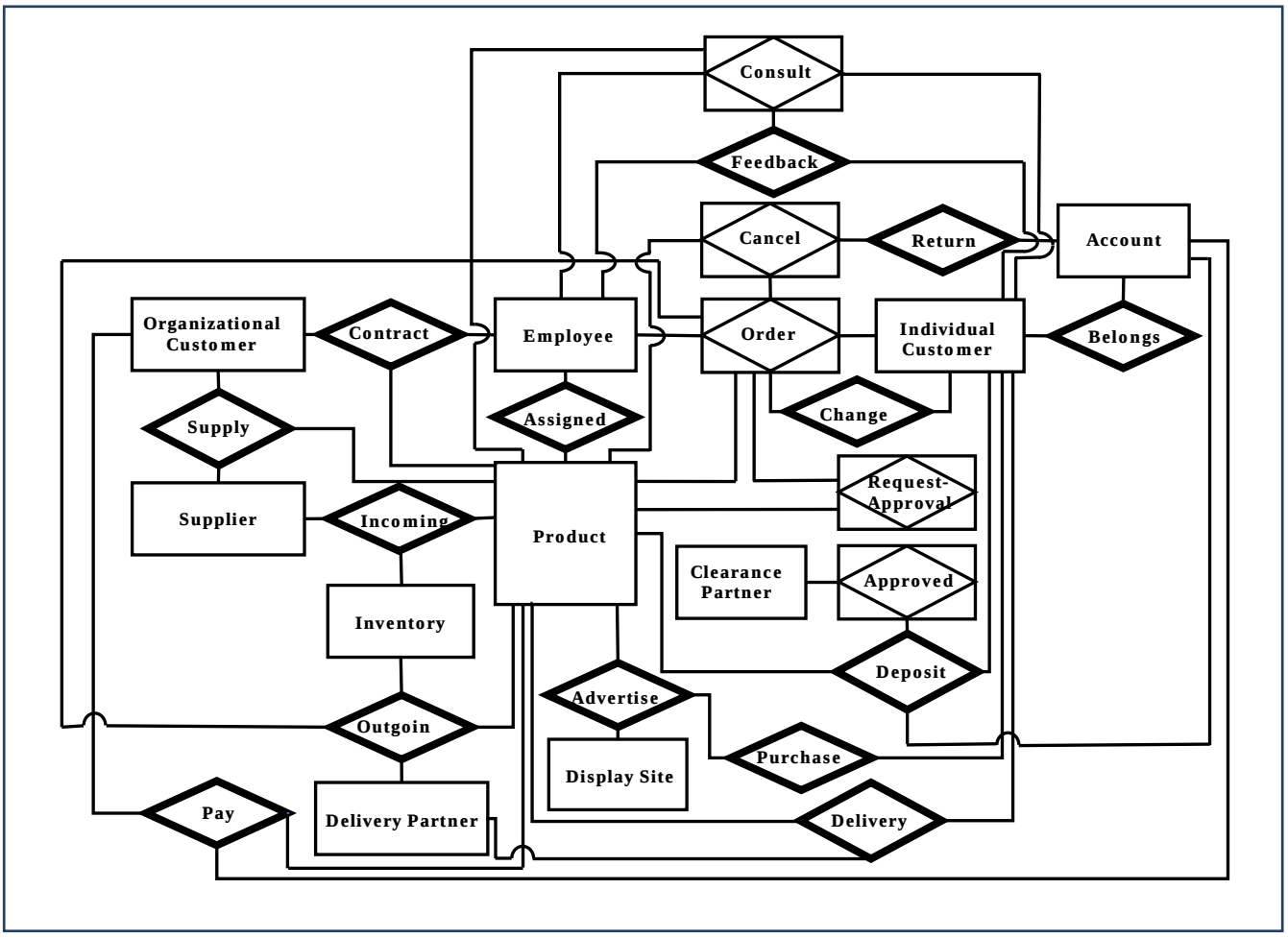

Fig. 3. Behavior-oriented Electronic Commerce Data Model Optimized to Reach 3NF Outcome

VI. HOW WOULD IT LOOK LIKE IF ALL THE NORMS ARE OBEYED

6.1 Generation a Perfect ER Model for Electronic Commerce Applications

Accordingly, we figured out 19 different behaviors as you can see in Fig. 3. Note that in Fig. 3 all such relationships are associated to two different entities, and thus directly connected to them, one at its very adjacent left and the other at its very adjacent right.

6.2 Improvements Attained

There are apparently 15 entities and 14 relationships in the rectified version of Fig. 3 after all. Note thatthere are 5 occasions of entity upgraded fromrelationship as they denote inseparably inter-dependent immediate predecessor or immediate successor to their sequences of behavior. In Fig. 1, there are 11 entities and 25 relationships altogether.

Thus, in terms of modeling box, either entity or relationship, there are 36 of them in Fig. 1. The count is 29 in the case of Fig. 3. Therefore, there is 32 percent of improvement in box-related modeling. More importantly, we should focus on the reduction ratio of data redundancy. It is well-known that the behavior-oriented data modeling [1, 9, 10, 11, 12] guarantees that the data redundancy rate of the final modeling outcome is 15 percent and the normal form type attained is fortunately and desirably automatically 3NF.

So, the mathematics is quite simple to calculate what we have succeeded. We achieved 40 percent reduction of data redundancy as the redundancy in Fig. 3 is 15 percent and that in Fig. 1 is 55.

VII. CONCLUSIONS

The analysis result of showing 55 percent of data redundancy in real-world data modeling practices almost coincides with the previous work on data obesity [2], in which approximately half of data of the corporate database is observed to be redundant.

Whilst there might be some positive side of looking at ERP, there seems to be absolute demerit of employing ERPs in the particular sense of what degree of optimum data redundancy rate we are going to pursue and interested in. There surely is a down-side of ERP as it was evident that abnormal data modeling practices of real world workforce are unwittingly and unconsciously very much influenced and affected by prototypical data models very popularly used in the ERP communities.

The issue of data obesity is very serious and significant given that the target data model we have taken in this easy is one of the best practices conducted by a real world workforce expert in data modeling. The data modeled this way usually constitutes mainly the structured data part of so called big data. Given as well that the unstructured part of data of big data provokes more data redundancy than the structured part of data, the issue of data obesity cannot simply be ignored at this point in time in the history of computing for us to be prepared for the future efficient and optimized computing in the coming era of big data.

REFERENCES

[1] S. Moon, Data Architecture, Hyung-Seol Publishing Co., 2014.
[2] H. Rhee, “Corporate Data Obesity: 50 Percent Redundant”, Global Journal of Computer Science and technology, July 2010.
[3] N. Kim, S. Lee, S. Moon, "Formalized Entity Extraction Methodology for Changeable Business Requirements", Journal of Information Science and Engineering, vol. 24, no. 3, pp. 649-671, May 2008.
[4] H. Choe, S. Moon, "A Methodology for Accurate and Redundancy-free Business Requirements Description Using Ontology", 2011 International Conference on Information Science and Applications, ICISA 2011, Jeju Island, KO, pp.246-252, April 2011.
[5] S. Lee, N. Kim, S. Moon, "Context-Adaptive Approach for Automated Entity Relationship Modeling", Journal of Information Science and Engineering, vol. 26, no. 6, pp. 2229-2247, November 2010.
[6] S. Lee, N. Kim, S. Moon, "Investigation on the Side Effects of De-normalizing Corporate Databases", Journal of Information Technology Applications and Management,vol. 16, no. 2, pp. 135-150, June 2009.
[7] Olikview, “SAP Data Model”, retrieved in https://community.qlik.com/docs/DOC-4294, May 2013.
[8] “ERP Design Data Model- Database Answers”, retrieved in www.databaseanswers.org/data_models/erp/index.htm, February 2003.
[9] S. Moon, "Reprisal of Unnecessary Data Replication”, Samsung Tomorrow, April 2015.
[10] S. Moon, "Misconception of Big Data”, Samsung Tomorrow, October 2010.
[11] S. Moon, "Data-Driven Human Resources Management: HR in the Era of Big Data", Proceedings of 9th HRD Conference 2015, Seoul, Korea, pp.39-44, September 2015,
[12] S. Moon, “Anything That Granted as Data May Not Usually Deserve to Be Genuine Data,” Economic Review,February 2014.

AUTHOR’S PROFILE

Haekyung Rheeis currently a professor at Department of Computer Games at Yongin-Songdam University in Korea since 2000. She was a professor at the Cairne’s School of Business and Economics at the National University of Ireland, Galway in 2016. She finished her MS in Computer Science at University of Illinois at Urbana-Champaign in 1985 and since then she was an Assistant Professor at the Department of Computer Science at Cheonan Technical University. She finished PhD in Computer Science at Sungkyunkwan University in 2000. She was Academic Dean of School of Information Technology at Yongin-Songdam University. She published more than 20 papers in the area of database management and cyber privacy. Her research areas include distributed processing, corporate data management, privacy, and social media data.