Student Database Design Using A Relational Approach

Asep Deddy Supriatna, Raden Erwin Gunadhi Rahayu, Dini Destiani Siti Fatimah

Abstract: Database design using relational approaches consists of the following stages: describing business processes, identifying and tapping entities, completing entities with attributes, creating relationship diagrams between entities equipped with cardinality, carrying out normalization processes, making structure language queries (SQL) and translation into programming languages. Managing student data manually, for various information needs at different management lines both horizontally and vertically, will cause several problems. These problems are related to accuracy and response time. This is because each identified file is independent so that when the search for records or data is made difficult, and the accuracy of the data is not maintained due to data redundancy. This research aims to design databases, thereby increasing accuracy and response time by eliminating data redundancies. Based on the condition above, in this study, a database of student data management will be conducted using the relational approach. From the simulation shown, it can be concluded that the database design of student data management using the relational approach has been able to accommodate data and information needs that are relevant to business processes that occur within the student data management system.

Index Terms: Conventional; Database; Redundancy; Relational.

1 INTRODUCTION

Manually managing data has caused many problems. This problem is mostly caused by errors due to human error factors. Some of the main problems caused by manual management are data redundancies, lots of paper use, and relatively large storage space [1], [2]. So that data management manually will reduce the level of data accuracy, relatively using longer time in accessing data or making reporting. To improve the accuracy of data usage, one of them can be done by eliminating redundancies. A method that can be used to eliminate data redundancy is to use a database. The database is a collection of files that are connected in the form of tables to eliminate the occurrence of data redundancies [1], [3]. The tables are linked to each other through the name-field name field. Each data connection will have a degree of connectedness known as the cardinality. There are several kinds of degrees of connectedness, namely one to one, one to many, many to one and many to many. From the results of the analysis of student data management carried out manually, it often raises problems with the situation if the student data is much the same. The similarities can include name, place, and date of birth, address, parent's name, study program name, faculty, or even the origin of the college [4]–[6]. This situation is exacerbated if the number of students who have to be managed is relatively large. One alternative that can be done is by developing from manual management to database use. There are several approaches to database design, including relational, object-oriented, hierarchical, and network approaches. However, from the design stage, the relational approach provides several conveniences that are shown to be more relevant between business processes, relationship diagrams between entities, and menu presentation. So that this relational approach, in certain conditions specifically for designing student data management, is easier to use the relational approach.

- *Asep Deddy Supriatna, Department of Informatics, Sekolah Tinggi Teknologi Garut, Indonesia. E-mail: asepdeddy@sttgarut.ac.id*
- *Raden Erwin Gunadhi Rahayu, Department of Informatics, Sekolah Tinggi Teknologi Garut, Indonesia. E-mail: erwingunadhi@sttgarut.ac.id*
- *Dini Destiani Siti Fatimah, Department of Informatics, Sekolah Tinggi Teknologi Garut, Indonesia. E-mail: dini.dsf@sttgarut.ac.id*

One approach that can be done in the database design is to use a relational approach. Conceptually this approach is based on the connection of each table identified through its name - field. All identified tables are connected so there is no independent and avoidance of data redundancy. This situation will improve data accuracy and response time in accessing data and making reports [7]–[10]. Database design using the relational approach is the conventional categorized approach. The main characteristic of the conventional approach is that all modeling uses conventional modeling. It starts with modeling business processes using flowchart modeling, then continues with modeling data using modeling relationships between entities until normalizing [11]–[13]. The stages and modeling carried out are very different from the approaches that use object-oriented. In the controlled scope and development of relatively static business processes, this approach is much easier to design and understand. Based on this description, in this study, the database design will use a relational approach. This is very different from the object-oriented approach, where various modeling activities will end in modeling using class diagrams. There is a relatively close relationship between the depiction of entities in the relational approach and class constraints in the object-oriented approach. That is both using attributes. However, in class design, it is added that the attribute is complemented by operating behavior or methods. That explains that the class can do activities. So it can be concluded that the relational approach is static if compared to the more dynamic object-oriented approach. Based on existing developments, many database designers have turned to object-oriented approaches on the grounds that database designers are easier to make modifications due to the dynamics of business processes. However, for certain things the relational approach is still feasible to be used as a database design method. The main reasons that can be considered are that the existing business processes are relatively static and the scope of the existing business processes is relatively controlled.

2 METHOD

In conducting database design using a relational approach, the main thing that must be considered is understanding business processes that are directed at identifying entities until designed relationship diagrams between entities. Furthermore,

from the relevance, the menu structure will describe the needs of data and information from the business processes presented. Therefore, David N. Kroenke (2002) in broad outline database design using the relational approach is as follows [14]:

1. Describe the business process in a pointer that is completed by who and who performs and the documents related to the business process;
2. Identifying and grouping entities. It is classified based on several categories, namely strong entity, weak entity, where the existence of a weak entity is very much determined by the existence of the strong entity;
3. Complement the entity with its attributes;
4. Make a diagram of the relationship between entities that are equipped with relations and cardinalities. Where each symbol of an entity or relation will be projected into a table. These tables are independent, some are the results of transactions from other tables;
5. Do the table normalization process from the relationship diagram between entities at least until the stages of the Normal Form Boy code;
6. Creating a structured query language (SQL) grouped into data definition language (DDL) and data manipulation language (DML);
7. Translating into programming and simulation languages.

3 RESULTS AND DISCUSSION

Based on the stages of database design, the database design of student data management is arranged as follows:

3.1 Description of business processes in a narrative manner

1. Identify student data based on their identity
2. Display a list of students based on certain groupings
3. Display student reports in quantity based on certain groupings
4. Display reports on student sequences in quantity
5. Make student conclusions in the highest or lowest quantity

3.2 Identifying and grouping entities

1. Student entities
2. Student-parent entity
3. Study program entity
4. Faculty entity
5. The incoming year entity includes its status

3.3 Completing the entity with its attributes

1. Student entities (Name of student; Place and date of birth of students; Student gender; Student religion; Address of student)
2. Student-parent entity (Name the student's parents; The work of student parents; Address student parents)
3. Study program entity (Code of the study program; Name of the study program)
4. Faculty entity (Faculty code; Faculty name)
5. The year entity enters as a new student (Year of entry; Student Status)

These entities are identified based on business processes that occur. What must be considered is the relevance of business processes that occur with the results of the identification of entities and relationships that must be available. From the analysis of business processes that occur, student identity is a collaborative entity of several small entities such as student-parent entities, faculty origin entities, study program entities, college year entry entities and student status entities that describe whether as new students or continuing / transfer.

3.4 Diagram of relationships between entities and their cardinality

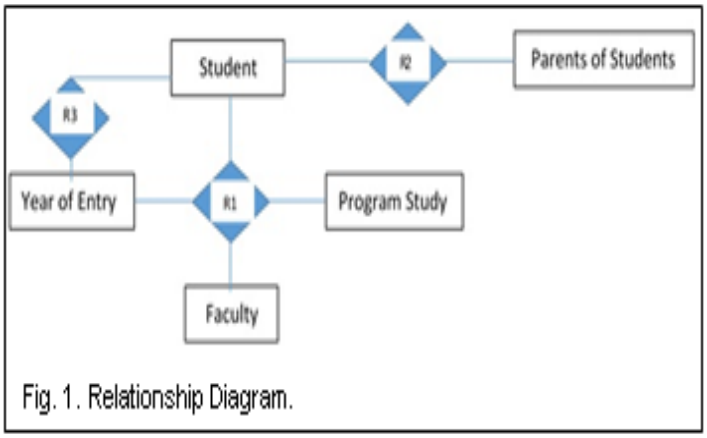

Fig. 1. Relationship Diagram.

Information :
R1 = Student Registration Number
R2 = Has
R3 = Become

From the ER diagram, it can be explained that the entities identified will relate to each other in accordance with existing business processes. It is hoped that data modeling using ER diagrams will represent data and information needs that are relevant to business processes. The appearance of the ER diagram design results will be presented in the main menu structure where it will first experience the normalization process.

3.5 Carry out the normalization process up to BCNF

Figure 2 illustrate the results of the normalization of both the entity and the relation. The results of the normalization will be used to complement the needs of the main menu. From the main menu, you will see the tables that are categorized as data and categorized as information.

Student Table

Student name	Place and date of birth	Gender	Student religion	Address of student

Table of Parents of Students

Name of Parents of Students	Work Parents	Address Parents

Study Program Table

Study Program Code	Name of Study Program

Faculty Table

Faculty Code	Faculty Name

Table of Year of Entry

Student Status	Year Number

Student Number Relationship Table

Student code	Student name	Study program name	Faculty name	Number of years

Student Relations Table has parents

Name of Student	Name of Student Parents

Table of Student Relations Year Enter

Name of Student	Year Number

3.6 Make SQL

1. **Data Definition Language (DDL)**
   a. Create Student Table (contains data about students with various elements that are considered relevant)
   b. Create Table Parents of Students (contains data about parents of students with various elements that are considered relevant)
   c. Create Table Study Program (containing data about study program identity)
   d. Create a Faculty Table (contains data about faculties in the college environment)
   e. Create Table Year of Entry (describes the year of entry of students as new or transfer students)

2. **Data Manipulation Language (DML)**
   a. Select Table Student Relations with Parents Students From Student Tables and Table Parents Students
   b. Select Table Student Relations Student Number From Student Table; Table of Study Programs; Faculty Table, Table of Year Entry
   c. Select Table Student Relations with Year Enter From Student Table; Table of Year of Entry

Based on the stages of the design, there is consistency between the business processes identified with the tables presented. The tables are designed in such a way that they present the menu structure of a database design related to student data management. These tables will be grouped into input and output data in the form of information or reports. From the identified table, it is clear that the input data is presented in the form of student tables, parents of students, study programs, faculties, and years of entry. While the output produced will inform the list of students based on the year of entry, study program, and faculty. And maybe also information that states the number of parents who have more than one student. Translations into structure query languages will group tables that are categorized as data definition languages and data manipulation languages. Furthermore, based on the SQL it will be translated back into a programming language that is considered relevant to the design results and needs.

4 CONCLUSIONS

4.1 Conclusion

Based on the results and discussion, conclusions can be drawn as follows: Database design using the relational approach makes it easy to create a database; The use of a relational approach shows good relevance between business processes identified with the results of designing the database.

4.2 Recommendation

Database design using the relational approach is recommended for use. However, to find out more about the results and capabilities, it is also recommended to design by using another approach.

5 ACKNOWLEDGMENT

Authors wishing to acknowledge Sekolah Tinggi Teknologi Garut that supports and funds this research publication.

6 REFERENCES

[1]. V. Varga, K. T. Jánosi-Rancz, and B. Kálmán, “Conceptual design of document NoSQL database with formal concept analysis,” Acta Polytech. Hungarica, vol. 13, no. 2, pp. 229–248, 2016.
[2]. W. Spoth et al., “Adaptive Schema Databases,” CIDR 2017, 8th Bienn. Conf. Innov. Data Syst. Res. Chaminade, CA, USA, January 8-11, 2017, Online Proc., 2017.
[3]. Y. Septiana, D. Kurniadi, A. Mulyani, and W. Baswardono, “Design of decision support system for blood analysis,” MATEC Web Conf., vol. 197, p. 03018, Sep. 2018.
[4]. O. Lukyanchikov, E. Pluzhnik, S. Payain, and E. Nikulchev, “Using Object-Relational Mapping to Create the Distributed Databases in a Hybrid Cloud Infrastructure,” Int. J. Adv. Comput. Sci. Appl., vol. 5, no. 12, Jan. 2015.
[5]. A. Mitrovic and S. Ohlsson, “Implementing CBM: SQL-Tutor After Fifteen Years,” Int. J. Artif. Intell. Educ., vol. 26, no. 1, pp. 150–159, Mar. 2016.
[6]. A. Abdel-Salam Al-Btoush, “Extracting Entity Relationship Diagram (ERD) from English Sentences,” Int. J. Database Theory Appl., vol. 8, no. 2, pp. 235–244, 2015.
[7]. C. Pinkel et al., “RODI: Benchmarking relational-to-ontology mapping generation quality,” Semant. Web, vol. 9, no. 1, pp. 25–52, Nov. 2017.
[8]. D. Kurniadi, A. Mulyani, Y. Septiana, and H. Aulawi, “Estimated software measurement base on use case for online admission system,” IOP Conf. Ser. Mater. Sci. Eng., vol. 434, no. 1, p. 012062, Dec. 2018.
[9]. D. Batory and M. Azanza, “Teaching model-driven engineering from a relational database perspective,” Softw. Syst. Model., vol. 16, no. 2, pp. 443–467, May

2017.
[10]. D. Kurniadi, E. Abdurachman, H. L. H. S. Warnars, and W. Suparta, “A proposed framework in an intelligent recommender system for the college student,” J. Phys. Conf. Ser., vol. 1402, no. 6, p. 066100, 2019.
[11]. D. Kurniadi, A. Mulyani, Y. Septiana, and G. G. Akbar, “Geographic information system for mapping public service location,” J. Phys. Conf. Ser., vol. 1402, no. 2, p. 022073, 2019.
[12]. A. Mulyani, Y. Septiana, D. Tresnawati, and R. Setiawan, “Design of culinary information system based on android using multimedia development life cycle,” J. Phys. Conf. Ser., vol. 1402, p. 022074, Dec. 2019.
[13]. D. Kurniadi, A. Mulyani, Y. Septiana, and G. G. Akbar, “Geographic information system for mapping public service location,” J. Phys. Conf. Ser., vol. 1402, no. 2, p. 022073, 2019.
[14]. E. F. Codd, “Relational Database: A Practical Foundation for Productivity,” Readings Artif. Intell. Databases, pp. 60–68, Jan. 1989.