

A Comparative Study on Filtered, Vertical and Horizontal Inheritance Mapping in Database

Ala Ahmad Lasasmeh

*Jordan, al-karak, Mutah University
Ala laasasmeh.yahoo.com*

Abstract

The inheritance is one of the most basic ideas of object technology that can be useful for database. This paper discusses and elaborates one of the most common mapping algorithms about mapping of inheritance structure to relational database. The inheritance mapping in database contains three various approaches: filtered, vertical and horizontal inheritance mapping. This paper review the fundamental measures used to comparative among three inheritance mapping in database through determine the strengths and weaknesses for each on: ease of access to the data, the speed of data access, Ad hoc reporting, Ease of implementation, Coupling, Support for polymorphism to the user and the development of modern commercial applications to protect the time and effort taken advantage of them while working. By tow method Based on the algorithms and rules, Object Relational Mapping (ORM) tool.

Keywords: *Object-oriented, Object Relational Mapping*

1. Introduction

The relational model [1, 5, 8, and 9] has undoubtedly been the most widely used and commercially successful way to date of modeling data. Its characteristics are very different from the earlier models. To begin, data entities are represented by simple tabular structures, known as relations. Entity relationships and data integrity are defined by primary keys and foreign keys. The design of relational database is based on the idea of normalization, the process of removing redundant data from your tables in order to improve storage efficiency, data integrity, and scalability, the relational model was proposed in 1970 by Edgar Codd to represent the natural structure of data without the database user needing to know about the machine representation. It promoted the idea of end-user programming and interactive querying of a database. This was a big step forward in the usability of databases.

The object-relational data model [1, 3, 8] is an extension of the relational model, with the following features: a field may contain an object with attributes and operation, complex objects can be stored in relational tables. At roughly the same time that the object data model was being proposed, the problem of storing objects in databases was being approached from a different angle. The object-relational model extends the capabilities of the relational model to allow objects to be stored in the columns of a relational database. An object relational DBMS is sometimes referred to as a hybrid DBMS. The hybrid DBMS approach was proposed by Michael Stone Barker in 1990, and has been implemented in some commercial RDBMSs, including Oracle. The motivation was a desire for databases that could store more complex entities and rules but that retained all the strengths of the second-generation databases.

The object-oriented approach has become very popular in application development and systems analysis and design, but object-oriented database management systems never gained widespread acceptance. Instead, relational database management systems continue to maintain their dominant role as the primary data management technology [4, 5]. Therefore, it is essential that relational databases be used effectively with object-oriented application development approaches

The inheritance [1, 2, 3, 6, and 7] is one of the core concepts in object-oriented programming. Most of the fundamental ideas of object technology are not supported in the relational database. The relational model as used by most RDBMSs has no concept equivalent to inheritance. As a result, the mapping can become quite involved. There are several possible strategies, and there is no single best way to do it. The optimum strategy depends on the precise nature of the inheritance tree.

The mapping inheritance [1, 2, 3, 6, and 7] is one of the most common mappings algorithms. When mapping objects to tables, a number of pattern languages are designed based on the problem structure for each three problems: mapping aggregation, inheritance and associations. This paper will focus only on the inheritance. As we mentioned earlier, inheritance is not supported by the relational database and special algorithms and rules are needed to transform them into tables. Different strategies and approaches have been suggested in the previous researches for mapping inheritance structure of classes to the relational tables. Basically there are three fundamental approaches: filtered, vertical and horizontal inheritance mapping in database.

The filtered inheritance mapping (FM) [6, 12, and 7] is also referred to as one table per hierarchy inheritance or one inheritance tree one table. The strategy uses one database table to store all the information for an entire inheritance hierarchy. Therefore the table contains a column for every attribute of every class in the hierarchy. To determine the class, a record belongs to, the table introduces a discriminator column. This column stores an identifier that allows mapping a record to a certain class type. If an instance of a class gets stored in the table, only the columns for the attributes of the class and all super classes will be filled. All columns, which do not belong to the class or its super classes, will be filled with NULL values.

The vertical inheritance mapping (VM) [6, 12, and 7] is also referred to as class table inheritance or one inheritance path one table. The strategy uses one database table for each class in the inheritance hierarchy. Each table contains columns for each attribute defined for the specific class and a primary key, which is shared between this class and its super classes. All values for inherited attributes will be stored in the tables for the super classes. Thus, the information for one class instance gets distributed between the table for the class itself and the tables for all of its super classes.

The horizontal inheritance mapping (HM) [6, 12, and 7] is also referred to as concrete class table inheritance or one class one table inheritance. It uses one database table for each class in the inheritance hierarchy. Each table contains columns for the attributes of a specific class in the hierarchy and also all attributes from the classes it inherits from.

This paper will comparative among various approaches of filtered, vertical and horizontal Inheritance in mapping database. That study include two-way in order to map from inheritance to relational database for these approaches: based on the rules, simple tool object relational mapping (ORM), activation of some of the important factors for comparing these approaches: ease of access to the data, the speed of data access, Ad hoc reporting, Ease of implementation, Coupling, Support for polymorphism, space consumption. To help the user in the development of modern commercial applications by choosing one of the approaches provides the speed and time and are highly efficient.

2. Over View of Block Diagram for Comparative Three Various Approaches of Mapping Inheritance

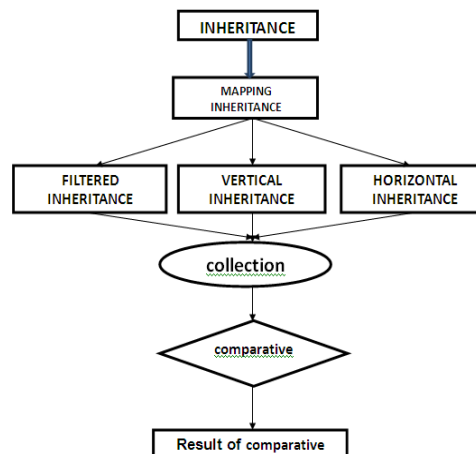


Figure 1. Block Diagram for Comparative Three Various Approaches of Mapping Inheritance

As we mentioned earlier, is not originally the adoption of inheritance by relational databases. This is why there are specialized models for translation structures inheritance. Is a kind of hierarchy, where the inheritance tree to store the database and, of course, it is possible to return, Mapping inheritance gave us three approaches: Filtered, Vertical and Horizontal Inheritance Mapping in Database the study of these Included two ways to mapping: Based on the algorithms and rules, ORM tool to know the advantages, disadvantages of these approaches.

The main factors used it in my study to comparative among these approaches: ease of access to the data, the speed of data access, Ad hoc reporting, Ease of implementation, Coupling, Support for polymorphism, space consumption. To help the user in the development of modern commercial applications by choosing one of the approaches provides the speed and time and are highly efficient, the following subsections describe each process in details.

2.1. Details about Block Diagram for Comparative Three Various Approaches of Mapping Inheritance

2.1.1. Inheritance: Inheritance is one of the basic concepts of object-oriented programming. Most of the basic ideas of the technology faces and are not supported in the relational database [6, 12, 7]. The model relational as most used RDBMSs has no equivalent to the concept of inheritance. Consequently, it can be set completely engaging. There are several possible strategies, and there is no better way one to do it. The best depends on the exact nature of the inheritance tree strategy.

2.1.2. Mapping Inheritance: Inheritance mapping is one of the most common appointments algorithms. When mapping tables to objects and a number of languages model is designed based on the structure of the problem for each of the three problems: inheritance [11, 12, and 17], aggregation inheritance and associations. This paper will focus only on the inheritance. As we mentioned earlier, is not supported by inheritance relational data base and special algorithms are needed to rules to be converted into tables. Proposed strategies and different styles in previous research structure inheritance mapping of classes to relational tables. Basically there are three basic approaches: filtered, and mapping the vertical and horizontal inheritance in the database.

2.1.3. Approach for Mapping of Inheritance: As previously mentioned before, originally inheritance is not supported by relational databases. This is why there are specialized structures for translation types of inheritance. It is a kind of hierarchy, where to store the inheritance tree to the database and, of course, it is possible to restore it [6-8]. There is two way using to give us the solution for the different approach of mapping inheritance for example in Figure 2, that show the some information include the super, sub class and there attribute to be the standard in our work.

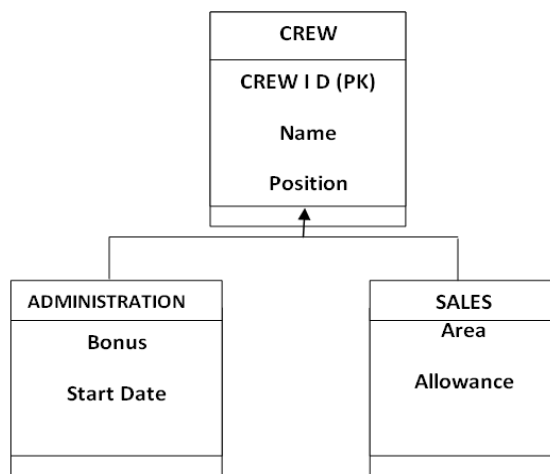


Figure 2: Inheritance Hierarchy for CREW

2.1.3.1. Based in Rules and Algorithm: In this method we just use some rules for each approach to translate to relational data by this: vertical mapping map each class of the inheritance to a separate table, horizontal mapping: Map each subclass class into a separate table containing the attribute of the sub, super class without table for super class, filtered mapping: Map the whole inheritance structure into a single Table [6, 7, 15].

2.1.3.1.1. Filtered Inheritance: Filtered mapping map the whole inheritance structure into a single table. Is a first approach in this paper is also known as the simplest approach, which is mapping the whole inheritance hierarchy to a single table, the solution of example in Figure 2 by filtered mapping.

CREW (CREW_ID, name, position, Bonus, Start Date, area, allowance, skill, is ADMINISTRATION, is sales)

2.1.3.1.2. Vertical Inheritance: Vertical mapping map each class of the inheritance (abstract or concrete) to a separate table. The draw of vertical mapping abstract know by using Figure 2 three class will map in different tables the subclass(crew) with its attribute and the tow subclass in different table by with the attribute and the primary key of sup class.

1. CREW (CREW_ID, name, position)
2. ADMINISTRATION (CREW_ID, Bonus, Start Date)
3. SALES (CREW_ID, area, allowance)

2.1.3.1.3. Horizontal Inheritance: Draw horizontal maps is to map each specific category in table. This independent approach is to convert each category separately for concrete table. All superior features will be added along with all the qualities of the sub-categories in the table know we will see how we use the horizontal inheritance by using the information in Figure 2.

1. ADMINISTRATION (CREW_ID, name, position, bonus, Start Date)
2. Sales (CREW_ID, name, position, area, allowance)

2.1.3.2. ORM Tool: Object-relational schema [21], in the purest sense, is a programming method that supports the conversion of incompatible types in programming languages and object-oriented, specifically between the data store objects and programming. You can use ORM framework that objects continue to model a relational database and retrieved, and ORM framework will take care of the data transfer between two non-compliant otherwise. Most ORM tools rely heavily on descriptive data for each of the data base and things, so that objects have to be to know anything about the database and the database does not need to know anything about how to structure the data in the application. ORM clean separation of concerns in the application of the data provides a well-designed, and the database and the application can work with all of the data in its original form designed. There are three step to create tables form Steps to inheritance structures.

2.1.2.2.1. Define Super Class and its Attributes: All subclasses will inherit attributes from super class. First, the system will prompt to user to enter super class name followed by the number of its attributes as shown in figure 3. After that, user has to define the attribute (variable name), data type and the length of the attribute as shown in the following figure.

| FINDING CLASS NAME | | |
|--------------------|---------|--------|
| VARIABLE NAME | TYPE | LENGTH |
| CREW_ID | INT | 3 |
| name | VARCHAR | 25 |
| position | VARCHAR | 50 |

GENERATE Reset

Figure 3. Define Super Class

| FINDING CLASS NAME | | |
|--------------------|---------|--------|
| VARIABLE NAME | TYPE | LENGTH |
| CREW_ID | INT | 3 |
| name | VARCHAR | 25 |
| position | VARCHAR | 50 |

GENERATE Reset

Figure 4. Define Attributes (Variables) of Super Class

2.1.2.2.2. Define Subclasses and their Attributes: The next step is to define the subclasses and their corresponding attributes as shown in Figure 5 and 6.

FINDING CLASS NAME

Created the super_ class => Crew

Enter sub Class name :

Number of variable :

Figure 5. Create Sub-Class Name

DEFINE SUB_CLASS

| VARIABLE NAME | TYPE | LENGTH |
|------------------|-------|--------|
| Bonus | FLOAT | 5 |
| <u>StartDate</u> | INT | 10 |

Figure 6. Define Variable(s) of Sub-Class

2.1.2.2.3. Transform the Inheritance into Tables: The important step is to transform the given inheritance structure into relational database tables. At this point, user will have three choices; to create a single table (Filtered Mapping) or; to create one class for each table (Vertical Mapping) or one class per concrete table (Horizontal Mapping).

Generate Table

CREW

- CREW_ID name
- Position
- Bonus
- StartDate
- Area
- Allowance
- skill
- is_ADMINISTRATION
- is_sales

Figure 7. Filtered Mapping (One table)

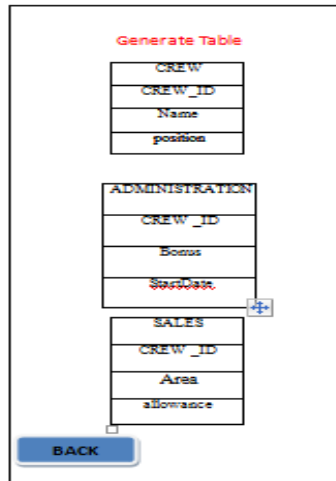


Figure 8. Vertical Mapping (One class one table)

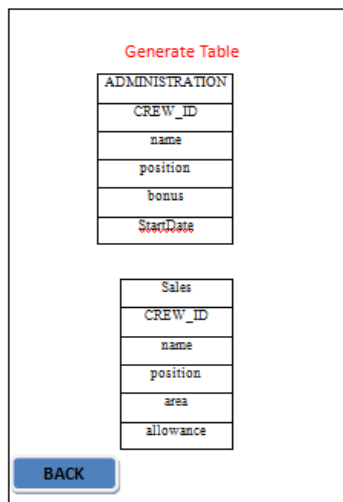


Figure 9. Horizontal Mapping (Concrete table)

2.1.3. Collection: In this process we collect all the advantage and disadvantages for each approach in inheritance mapping include the tow way for each approach first Based on the algorithms and rules, second object relational mapping (ORM) tool.

2.1.3.1. Filtered Mapping: These are the advantages and disadvantages: First the advantages: is considered as the simplest technique and required less maintenance and mostly good for a small database system, any attributes of a new subclass will automatically insert into the same table, there is no join tables needed, this approach yields the maximum performance for polymorphic queries. Second the disadvantages: waste a lot of space with the NULL values, need to add type identification for each subclasses, heavy traffic for the table since all subclasses are crowded in the same table, coupling within the class hierarchy is increased because all classes are directly coupled to the same table, any modifications in one class can affect the table which can then affect the other classes in the hierarchy, table can propagate quickly for huge hierarchies.

2.1.3.2. Vertical Mapping: The advantages and disadvantages of it, First the advantages: is vertical Mapping is considered as the most efficient and acceptable approach by most of the software developer since it will maintain the data

normalization, transforming any other subclass will not affect the existing tables, new subclass will automatically create another table. Second space consumption is realistic because each class involves one table. Nonetheless it might produce too many tables for a large database system, read, write and update data using this technique will take longer because we need to access and join multiple tables, not suitable to perform ad-hoc reporting.

2.1.3.3 Horizontal Mapping: The advantages and disadvantages of it, first the advantages: is the space consumption is at the best because there no redundant attributes, furthermore, this approach produce a good performance in terms of accessing a single object's data. Second the disadvantages is altering a class will affect not only that particular table but it also will change the table of any of its subclasses, this method will not support multiple roles and hard to conserve data integrity.

2.1.4. Comparative: In this include the comparative among three approach (FM), (VM), (HM) based in Main measures to let the user know the strengths and weaknesses for each one to choose the best to development of modern commercial applications to protect the time and effort taken advantage of them while working. By tow method Based on the algorithms and rules, Object Relational Mapping (ORM) tool.

2.1.4.1 Ease of Implementation: Is the physical realization of the database and application designs This feature is very important for the user because the user needs to select one of the approaches to work with it and use it in the development of projects and other applications in the case was one of the easy methods Ease of implementation This requires that the user be used.

2.1.4.2. Ease of Data Access, Speed of Data Access: A set of high-level productivity tools to allow applications to be developed, maintained, and deployed with relative ease and speed of data access this allow to the user developer to Deal with his work in simple way when you can get the data you want in simple ways and quickly This makes you very comfortable in handling with the data you have.

2.1.4.3. Coupling: Coupling mostly describes the extent to which the components and classes depend on each other and divided into loose coupling and tight coupling. Loose coupling is usually associated with a strong cohesion and vice versa. It is very important for the user who uses maps inheritance in requests development.

2.1.4.5 Ad hoc Reporting: Allows end users to easily build their own reports and modify existing ones with little to no training. Ad hoc reporting create reports that deal with individual information you need quickly and easily, allowing end users to dynamically modify and drill through report data for powerful information analysis. This kind of flexibility frees up valuable IT resources and gets information to end users early, empowering them to interact with the information about the data.

2.1.4.6. Space Consumption: This is the amount of disk space required to store the database files [20]. You may wish to minimize the amount of disk storage used because if he was the availability of this large area is striking that the project will be successful and earn a lot of physical and other gains. In the finale we collect all the result for each inheritance mapping depends on many factors, but using all these factor we don't arrive to best approach because

every one good in specific domain just everyone has its point also good or bad this result show in Figure 10.

| Factors Consider to | VERTICAL INHERITANCE | HORIZONTAL INHERITANCE | FILTERED INHERITANCE |
|--------------------------|----------------------|------------------------|---|
| Ease of data access | Simple | Medium/Simple | Simple |
| Speed of data access | Fast | Medium/Fast | Fast |
| Ad hoc reporting | Medium/Difficult | Medium | Simple |
| Ease of implementation | Difficult | Medium | Simple |
| Coupling | Low | High | Very high |
| Support for polymorphism | High | Low | Medium |
| space consumption | is realistic | best | Waste a lot of space with the NULL values |

Figure 10. A comparative on Filtered, Vertical and Horizontal Inheritance Mapping in DB

2.1.5 Result of Comparative: The result of comparative between among inheritance mapping is everyone has a good point to let the approach be the best to use for the user develop, but in the same time the approach has a bad point let the user go away from it. We can't choose know any of them, even we determine the application that will use the approach, because even application different from the database sometime big, small we can't determine the Appropriate approach

3. Limitations

There are three common ways of mapping inheritance in databases; each one of them has its advantages and disadvantages. It is very important to be familiar with those strategies. The point to go away with is that none of these approaches is perfect; each has its strengths and weaknesses. Preference is only one of them through the user domain means as needed.

Related work

The searchers in these papers [1, 3, 6, 7] define many concept related to relation database and how in paper [9, 21, 17] take the object oriented and relational technologies to Connecting by the idea of inheritance in object oriented.

There are many advantages and disadvantages for the filtered mapping that purposed in [6, 7]. There is only one table to worry about. There are no joins in data retrieval. It does not require any re-sale of the debt, which pays the highest areas or down the hierarchy to change the database schema. On the negative side, cannot be involved in all areas to be recorded, which may confuse people, who work directly with the table. Columns used only by sub lead to wasted space in the database. The table may end up to be a big, and require many indicators frequent lock. Furthermore, there is only a single namespace for the fields, so you have to be sure you do not need the same name for different areas. It is simple to add new categories, because you only need to add new columns to the table. He also mentioned that this strategy is very suitable to access data quickly, especially for ad hoc reporting. On the negative side, of Ambler indicates that coupling with the class hierarchy increases, because all

classes are associated directly to one table. Any change in the class may affect the table, which then affects the other layers in the table. Refers to the type becomes complicated, when significant overlap exists between the types and level of interdependence is suffering, because they are several concepts are stored in a single table.

In this paper we mention many advantages and disadvantages of vertical mapping determines in [6, 7] from this point of view, and the strengths of this strategy is that all the columns in the table are concerned, which prevents the waste of space, and that the relationship between the model area and the outline of a clear and explicit database, which makes it easy to understand. Weaknesses in the strategy are the need to perform joins the load object. Refactoring of the top or bottom areas of the hierarchy causes the database changes. Super type tables may become a bottleneck, because they have to be frequently accessed and a high degree of normalization may make it difficult to understand queries ad hoc. In addition to best practices Fowler, states that the TPC approach allows adding subsections very easy, because you only need to add a new table. On the negative side, he adds that this strategy needs many tables in the database, one for each category (in addition to the tables to keep relations).

Many writers talk about the good and the bad point of horizontal mapping for working [6, 7]. Each table is a stand-alone and has no areas is irrelevant. It can use this table by other applications that do not use objects. There are no joins to do when reading data from the tables. Access each table only when access to that category, which spreads load over multiple tables. On the negative side, push the areas up and down the hierarchy needs to change the table definitions. Changes to the fields in super classes lead changes in each table contain this area, because it is repeated superior fields across tables. Search the super class forces to check all the tables from the corresponding subclasses, which leads to multiple database up or join strange, because all the necessary data for one category were found on the object to change its role. In this case it is necessary to copy the data in the appropriate table. Similarly, it becomes difficult to support multiple roles and the data still maintain integrity in one table. There is an important point of this strategy is when.

The object-relational mapping ideas of object models are more attention to the principles of software engineering such as coupling and cohesion, and inheritance and packaging. On the other hand, mathematical principles such as referential integrity and normalization are more bound to the relational model [3]. Not resist the match when these models are both used in the different points of view and try to access and manipulate data occurs. So far, various attempts have been made in a way to eliminate the impedance mismatch. Besides, the developers have met to talk several issues when it comes to the appointment of relational databases object. However, we will discuss only two outstanding issues because they are closely linked to our goals: mapping the structure of inheritance mapping feature [1- 4].

Many tools has been enhanced for different purposes for [5, 8, 9] papers. In [5] the normalization of relational databases confined to the conceptual level by Dr. M. S. Patwardhan, *et.al*. The FDs were added on the ER-diagram has achieved a great achievement. To facilitate the process more, released program implemented on the integrated development environment (IDE) and named the AER IDE. The extended design Graphical Editing Framework (GEF) and Draw2d plug-in functionality of Eclipse. The AER IDE helped in the drawing, Storage, validate, and the normalization of the AER diagram. Therefore easy making the process more visual process easy to understand and translate the process rather than theory. Work in [8] explain the way to draw diagram a semi-automatic for marking on the semi-automatic diagram. Also through the use of a schematic drawing tool which extracts contextual information by selecting the graph matching components in the student schemes. This tool will help to change the traditional drawing to make the evaluation process more suitable for the marks to be semi-automatic. That ambiguity disappears in the contextual meaning for each component of the process of drawing during the marking. Which increases the opportunity to understand the student's ideas and taking feedback? In [9] describe architecture of ER-draw, and the details of its implementation and translation of the XML. This tool consisted of XML, OO and relational database theory. Use the ERML to facilitate

interoperability of various practical tools for the ER diagram. Also draw connotations confirm that the validity for ER diagram.

5. Conclusion and Future Work

The concept of inheritance cast in many exciting variations interesting when you save objects to a relational database, which is one of the basic concepts in object-oriented technology, and revealed how it can be represented in this structure inheritance relational database. Issue boils down essentially to learn how to compare all filtered, vertical and horizontal mapping inheritance in the database. The way in which to solve this challenge can have a significant impact on the development of modern business applications and users by using one of the methods. This paper has chosen very important criteria: : ease of access to the data, the speed of data access , ad hoc reporting , ease of implementation ,coupling , Support for polymorphism to the user and the development of modern commercial applications to protect the time and effort taken advantage of them while working. By tow method based on the algorithms and rules, object relational mapping (ORM) tool. Response time and this is the elapsed time to complete a single transaction. From the user point of view, you want to minimize the response time as much as possible. However, there are some factors that affect the response that you may have no control over, such as system load times or communication time, approach you use. I suggested here that the user in selecting Start project, especially the amount of data that will be used for that in this case can select one of the curriculum because some of which support small businesses and big *etc.* in the end, medium.

In the future I suggest away to solution the limitation her by create a new approach of inheritance mapping include all the good point in the different approach. That the user developer can choose it to increase the quality of its work with no bad point.

References

- [1] M. Fowler, Patterns of Enterprise Application Architecture, Boston: Addison-Wesley, (2003).
- [2] C. S. Ambler, "Agile Database Techniques: Effective Strategies for the Agile Software Developer", Wiley, (2003).
- [3] W. Keller, "Mapping Objects to tables", A pattern language, Proceedings Euro Plop, (1997).
- [4] S. W. Ambler, "Mapping Objects to relational Databases", Ambysoft Inc., (1997).
- [5] F. Lodhi, and M. A. Ghazali, "Design of a Simple and Effective Object-to-Relational Mapping Technique", SAC'07, ACM, (2007), March 11-15, pp. 1445-1449.
- [6] C. Murdaca, "An Object-Relational Compiler", World Congress on Computer Science and Information Engineering, IEEE, (2009).
- [7] Dr. M. S. Patwardhan, P. S. Dhabe, A. A. Deshpande, S. G. Londhe, M. L. Dhore and H. K. Abhyankar, "diagrammatic approach for complete automation of relational database normalization at conceptual level", International Journal of Database Management Systems (IJDMS), vol. 2, no.4, (2010), pp. 132-151.
- [8] F. Batmaz and C. J. Hinde, "A diagram drawing tool for semi-automatic assessment of conceptual database diagrams", IN: Danson, M. (Ed.), 10th CAA International Computer Assisted Assessment Conference, Proceedings of the Conference on 4th and 5th July 2006 at Southborough University, Southborough: Lou borough University, (2006), pp. 71-84.
- [9] S. Xu, Y. Li, S. Lu "ERDraw: An XML-based ER-diagram Drawing and Translation Tool, In Computers and Their Applications, pp. 143-146.
- [10] S. Ambler, "Mapping Objects to Relational Databases: O/R Mapping in Detail", Available at www.agiledata.org/essays/mappingObjects.html, Accessed, (2009) September 19.
- [11] F. Lodhi and M. A Ghazali, "Design of a Simple and Effective Object-to-Relational Mapping Technique", SAC'07, ACM, (2007) March 11-15, pp. 1445-1449.
- [12] E. Ralph, J. B. Foote, "Designing reusable classes. Journal of Object-Oriented Programming", (1988), p. 22-35.
- [13] M. L. Fussel, "Foundations of Object-relational Mapping"
- [14] <http://www.chimu.com/publications/objectRelational/objectRelational.pdf>, (1997).
- [15] W. Scott, "Ambler President, Ronin International Mapping objects to relational data bases what you need to know and why", (2000) July.
- [16] M. Gar, "Hibernate Tutorial 08 Inheritance Mapping", (2006) September.
- [17] Diploma Thesis, object-relational mapping", Bratislava (2006).
- [18] M. Lorenz and J. Albrecht, "Object-Relational Mapping Strategies revised".

- [19] A comparison of Row- and Column-oriented Database Systems, Engineering and Technology (ICCIET'2014) (2014) July 17-18, Phuket (Thailand).
- [20] E. Ralph, J. and B. Foote, "Designing reusable classes. Journal of Object-Oriented Programming", (1988), p.22-35.
- [21] A. Dennis, B. Wixom, and D. Tegarden, "Systems Analysis and Design with UML Version 2.0: An Object-Oriented Approach (Third Edition.)", New Jersey, John Wiley & Sons, (2009).
- [22] T. Turcu, "Space consumption, an important factor in the development of transport systems", Posted on, (2011) December 20.
- [23] A. Lipitsäinen, "ORM – Object Relational Mapping. Dpt. of Business Information Technology", Haaga-Helia University of Applied Sciences, Helsinki, Finland.

Author

Name: Alaa Ahmad Lasasmeh
Address: Mutah-alkarak-jordan
E-mail: alaa.ahmad465@yahoo.com