

Unique Data Model (UDM) for Data Warehouse: Characteristics, Benefits and Limitations

Monika Rathore

Research Scholar

Suresh Gyan Vihar University, Jaipur, Rajasthan, India

Dr. Vijay Singh Rathore

Director

Shree Karni College, Jaipur, Rajasthan, Jaipur

Dr. Ripu Ranjan Sinha

Professor

Suresh Gyan Vihar University, Jaipur, Rajasthan, India

Abstract- Objective of this paper is to propose a data model which can provide suggestion to remove the problems which occur due to the limitation of Relational database .The UDM(Unique data model) data model is a model for database Which has the capacity to overcome the problems in data warehousing applications. In contrast with the relational model, which is about having multiple records and dealing with entities and attributes, UDM with entities those having discreet independent presence, and the relationships among those entities are represented as Links.

Keywords – Relational Database, UDM, Data Warehousing

I. PROBLEMS ASSOCIATED WITH RELATIONAL DATA MODEL

1. Each new relational application requires a new set of programs created from beginning, which is time taking, costly and inefficient.
2. Relational applications are not simple applications which can be altered as per the requirement of huge number of individual users.
3. Relational application is not able to keep record of a particular thing's information which is not related to each other thing of the same type and it causes the difficulty in fulfilling the customer needs.
4. Information for the same things in the real world is stored in different way in each relational database, so it is problematic and costly to merge two databases. To overcome above mentioned problems this paper propose UDM model mentioned below

II. UDM (UNIQUE DATA MODEL)

The UDM based on a subject-connector-target arrangement including sentences constructed in English or could be in any other language. Examples of expressions that are appropriate for the UDM can comprise:

Blue is a Color

Rashmi is a Doctor

Cook makes food

Rides are in a park

A Park is in a Jaipur

By take in the above example it could be seen that the connector is a medium to connect. The connection's (Link) aim is to find the relationship(connection) between the subject and the object.

The UDM consist two structures, First is group of objects and a group of connectors that are used to connect them with each other entries must hold a unique indication , a type, and a name with the object structure. Entries in the connectors have a unique indicator beside with indicators for the associated source, subject, target and connector.

In UDM model database contains two types of data structures:

Object- Each object has unique identifier: A name and a type.

LINK- Each LINK has a unique identifier, and it consist three other things, that signifies the source, connector and target of a fact that is stored about the source in the database. Each of the three things identified by the source, connector and target may each be either a LINK or an object.

The UDM model would use these two structures to store the piece of information. For example "Train MR1983 arrived at JAIPUR STATION on 18-Aug-14 at 12:55Pm". There are seven items: the four nouns Train MR1983, JAIPUR STATION, 18-Aug-14 and 12:55Pm and there are three connectors arrived at, on and at. To store data there is requirement of three connectors they are

Train MR1983 ,

Arrived JAIPUR STATION

On 18-Aug-14 at 12:55Pm.

For the first LINK it makes use of the connector arrived at to link the items Train MR1983 and Jaipur station .Now second LINK is the first and second lines joint. It uses the connector on to link the first LINK and the object 18-Aug-14.The third link contains all three lines. It uses the connector at to link the second LINK and the object 12:55Pm. At times when writing LINKs, As a replacement (instead) for to use new lines to show each LINK it is more suitable to go with the long string .it can be good practice to put brackets around each LINK. By doing so, mentioned example will appear like following:

Train MR1983 reached at Jaipur station on 18-sept-14 at 12:55Pm. In the UDM model it will be possible to store an unique database in just two tables:

One table for object and one table for LINK. Now after giving each object and LINK a meaningless number as an identifier, which will act as a primary key.

Object

Identifier	Name
85	Train MR1983

10 Jaipur station
 43 18-Aug-14
 56 12:55Pm
 23 arrived at
 78 on
 11 at

LINK

Identifier	Source	connector	Target
74	85	23	10
03	74	78	43
64	03	11	56

By viewing above example can have idea that these are not proper relations, since every entry in the Source, connector and Target columns can be an identifier of an object or could be a LINK and the Name column can consist just about anything at all.

III. MODIFICATION

In relational model, transactions modifies databases modifying (creation , alter, deletion) tuples in relations the word “modification” to represent any change of a database’s state, comprises creation , deletion and changes) but in case of UDM , in normal case data in a database is not at all physically removed or modified. It is not like there in relational model where values are physically altered to different values in UDM model it does not happen. In fact , modification in an UDM are resulted by logically removing the suitable links and then inserting some new links. A link which is logically removed by including of another link, called a end link and the source of stop link will be deleted link .Thus, data is altered in an UDM by a single process: By adding of new links.

Suppose there is a firm whose information are recorded in case of relational database it update its address, whenever the record is altered in database the new address stores again the old one, and the old address is not able to be seen to the user. So we can say that new data overwrite and wipe out the previous data.

Before:

Customer Address

ABC Ltd C-102 JAIPUR

After:

Customer Address

ABC 25 INDRAPRASTH VIHAR

When this function will be executed in an UDM data- base, the link to the previous address is flagged and does not remain present, and then a new link to the newer address is formed. The previous link and the previous address both reside in the database, and can be accessible using suitable software abilities. And the data is not overwritten or damaged.

Before:

ABC Ltd address C-102 Jaipur

After:

ABC Ltd address C-102 Jaipur

deleted by Transaction 98291...

ABC Ltd address plot no. 25 Indraprasth vihar created by Transaction 98297..

By using UDM database It will be possible for transactions will be able to be seen at a time at which they were received by the database:

ABCLtd address c-102 Jaipur deleted by Transaction 98291...

ABC Ltd address plot no. 25 indraprasth Vihar created by Transaction 98287...

Transaction 98287.. Processed 2013/1/12 12:08:55 ... Effective 2014/02/25 05:00:21 not like the relational model, the UDM model does not have need to detach present snapshot and historical journal. Even though, in the nonappearance of full mirroring, these ability is required to make sure that the database will be recovered in case of media failure. This all can be suggested because data is not altered or actually deleted, It is even feasible to view the condition of the database, At a specified moment in time.

IV. METACODE

As the higher level languages booming up in software Era so it has become desirable to introduce more abstraction into programming languages .The UDM shows this new level of abstraction. In UDM there is no need for the programmer to be aware of whichever the physical or the logical structure, and it permits them to put attention in exclusively on logic and methods by which users get involve with databases, and databases get involve with each other. In relational model each new relational database application requires a new set of programs which are written from scratch, because as mentioned earlier a program which is written for particular application may not be used again for another. So this may lead to the requirement of continuous deliver of new programs those are time-taking and costly. This aspect of software development is commonly acknowledged and in a relational database, every relation is formed in a different way. Every one has a dissimilar number of columns, and every column has a dissimilar column heading and a dissimilar domain also. Due to this, It bounds to design programs in the region of the relations and by Using conventional programming languages, it is unfeasible to write a competent program that is able to access a relation whose organization or structure is not known when the program was written, so it is not possible to make a solution that will solved any problem because Every program should be written by someone who is having good information of the relations and it will make use of, along with a program that make use of one group of relations cannot be make use of by means of a different set. In any business data processing application, each company entity is signified by at least one relation, and mostly commercial applications includes between

1000 and 5000 company entities, so each time new application requires somewhere between 1000 and 10,000 new programs selected to written from scratch. And this may take several weeks even by using latest technolog

V. BENEFITS OF UDM

In case of UDM model, Metacode permits to write the programs that may run on every business body without changes. This definitely minimizes the total amount of newly created programs required for one different application. Although because the requirement arises to deploy more applications the part of few different requirements that is completed by current programs improves, so automatically the need to write few new programs minimizes. But currently programmers have to rewrite the programs in continue manner to work with new tables. But with the help of Metacode it may decrease the total expenses of computing.

The reusable feature of Metacode allows that many simple applications can be programmed using existing programs. This helps end-users in the creation of applications. As they become familiar with a important repertoire of Sentences programs, by using that many end-users can be able to develop and deploy simple applications without taking help of any specialist.

Dissimilar to an UDM database may save the data that is totally new to particular thing of kind ,one customer, one product .Also, by using the metacode strength , applications that are using UDM databases may simply present, update and query information which is absolutely new to single specific thing.

To solve the problem with UDM model take the example of two tables that operate the same function: the two customer tables. It is known that it is not possible to just include the rows from one customer table to the different customer table, because to make it done each row in a relation should have the same columns, and there should be minimum two columns that are not exactly in each table. So it is mandatory to examine both tables and match up the corresponding columns and when have the columns which have the functions which clearly match, generally they would be developed on dissimilar domains. Now for example if a particular designer has taken up decision to get the customers from their numbers, and some other customer from an alphabetic string. Then it is required to chose one or the other, and then assign a new keys to the one that generally not chosen, and then replacing the whole database foreign keys to show the keys that is replaced already. All the operations motioned above are related with only single column in one pair of tables which matches, this is different with the relational database design procedure in that several tables in single database could not be directly similar or have relevance in the other

VI. LIMITATIONS OF UDM

1. The UDM model doesn't use records. From the basic storing methods to the object/relational models, the essential unit of data storage always been a record that includes all the parts of information about an object or an entity, stored closely. The main issue is efficiency.
2. Efficiency could be the major issue of any database model and it is in case of UDM also because UDM models discard the process of storing the data on record-based method which is generally used by all used by different data models in order of accumulating data items independently.
3. In the UDM model, Each occurrence of an entity type needs a name. There is no equivalent concept in the relational world, and it satisfies requirement for some people who prefers the naming, wishing that a thing's name, when it requires one, must be one of its attributes. Naming things is an essentially main feature of human discussion, so it is unreasonable to give out with it when we use databases. In the UDM model, the entity name theory keeps away from the requirement to select which attributes will be used to recognize something visually but for every entity carries its own only one of its kind surrogate anyway, so recognition is preserved for internal reasons or to make redundant identifiers like pin codes. So to conclude three opinion are there at the very first that not everybody mandatory calls to something by the similar name, as a result this is not optimal to need entities to carry single, particular name.

4. The UDM model uses scalar values and strings in its own right by means of independent existence and identity, as a replacement for isolated values which signify objects. This Method considerably decreases the total amount of work required to run queries, and it deals with some other benefits if today's databases will integrate this capability, the several bug would be solved have with a little resources will consumed . Though, the method relies at a basic level on the make use of references or pointers to values in place of values themselves, and it is controversial.

VII.CONCLUSION

This paper includes the various building blocks of Unique data model and states the relationship among them and it shows that how this proposed UDM model will be useful for handling the analytical environment of an enterprise, and by their unique features the data warehouse can be implemented in a better way for Business enterprise.

REFERENCES

- [1] Fitchman P, Preventing credit card fraud and identity theft: A primer for online merchants, Information Systems Security, p 5259, 2001
- [2] Heimann J, Introduction to Oracle Access Manager, Oracle Corporation, 2006
- [3] TDWI Data Warehousing Concepts and Principles: An Introduction to the Field of Data Warehousing. Chris Adamson, June 2012.
- [4] Data Virtualization in Business Intelligence Architectures. Revolutionizing Data Integration for Data Warehouses. Rick F. van der Lans, 2012.
- [5] T. Kessel, O. Stern and F. Rousselot. "From frames to concepts: building a concept language on a frame-based system." Int. Workshop on Description Logic (DL-95) (Rome, 1995) pp. 140-142.
- [6] . P. Bresciani. "Querying Databases from Description Logics." KI'95 Workshop in KRDB'95 (Bielefeld, Germany, 1995) pp.1-4.
- [7] S. Bergamaschi and C. Sartori. "On Taxonomic Reasoning in Conceptual Design." ACM Transactions on Database Systems Vol.17, n°3 (1992) pp.385-422.
- [8] D. Benevenuto, S. Bergamaschi, S. Lodi and C. Sartori. "Using Subsumption in Semantic Query Optimization." (CIOCCNR, Bologna, Italia, 1993
- [9] E.F. Codd, S.B. Codd and C.T. Salley. Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate. White Paper funded by Arbor Software. 1993.
- [10] Victor González Castro " The Use of Alternative Data Models in Data Warehousing Environments", Heriot-Watt University, May 2009
- [11] G. I. Popek and D. Downs, "Data Base Management Systems and Ingres," Universit of California at Los Angeles, 1979.