SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

TMLAI TRANSACTIONS ON
MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

VOLUME 5 NO 4, 2017
ISSN 2054-7390

Migration of the Temporal RDB into Temporal ORDB including Bitemporal Data : Phases

[1]Ain El Hayat Soumiya, [2]Bahaj Mohamed
[1]LITEN Laboratory, Faculty Of Science and Tecnology,University Hassan , SETTAT, Morocco
[2]PROFESSOR, LITEN Laboratory, Faculty Of Science and Tecnology,University Hassan , SETTAT, Morocco
soumya.ainelhayat@gmail.com, mohamedbahaj@gmail.com

ABSTRACT

This paper proposes an approach for migrating existing relational database (TRDB) according to SQL: 2011 standard into temporal object relational database (TORDB) including Bitemporal data. This is done with methods that can extract various functions from a TRDB, which is based on the different kind of the relationships between tables. To do that we're going to enhance a representation of a varying time database's structure in order to make hidden semantic explicit. In contrast to other studies, our main goal here is to offer a first and better solution to mentioned limits to existing works, in order to provide the efficient method for the translation from TRDB to TORDB. We are going to take an existing RDB with bitemporal data features as input, we provide our meta-model in order to get a correct description of a data structure, enriches its metadata representation, and generates a bitemporal data Model (BTDM), which captures essential characteristics of temporal databases for migration. A prototype has been developed which proves the effectiveness of this solution, and temporal queries has been made with help of Oracle 12C.

Keywords: Bitemporal Data, SQL:2011, Temporal database, RDB,ORDB, Semantic Entichment

1 Introduction

Temporal database is one of the most important parts of the information technology. It is generally known that temporal database stores the history of the objects or the database activity. Even today, a large number of database applications are based on time in nature. It is an important attribute of each and every real world application. Every event important for operation of enterprise must be record at specific point at time. Events occur at specific points in time, objects and the relationships among objects exist overtime [1]. This importance made it necessary to make better description and clearly some tasks of database system. Especially, the need to retain trace and audit the change made to a data and the ability to plan based on past or future assumptions are important uses cases for temporal data [2]. Many database applications require management of time varying data. The most common example of such applications involves health care management, medical records of patients, reservation system, banking and accounting, and decision support system.

The literature on temporal database offers three dimensions of time for temporal data support, which are independent each to other: transaction time, valid time, user-defined time. Valid time is the period during which a row is occurred in the reality in the database, it is the fact as is valid in the modeled in the real.

DOI: 10.14738/tmlai.54.2977
Publication Date: 15th August, 2017
URL: http://dx.doi.org/10.14738/tmlai.54.2977

Transaction time automatically captures changes made to the state of time-variant data in a database [3].User-defined time is a time representation implemented to satisfy specific needs of users. There is also another dimension of temporal database called bitemporal data. Bitemporal database systems support both valid time and transaction time. On the other hand, bitemporal databases model our changing knowledge of the changing world, hence associate data values with facts and also specify when the facts were valid. There by providing a complete history of data values and their changes [4]. Several temporal data models, which support different dimension of time, are discussed in previous work.

The relational database model has come a long way since the 1970s and it has become the dominant model of database [5], most traditional database applications are based on traditional management systems. Although, the relational database is designed to serve many applications which need only store the recent state of data, they are insufficient for those which need to retrieve past as well as current and future [6] .The need to shift time manipulation from application to relational database is identified and implemented in SQL: 2011 standard. One of the advantages of the SQL: 2011 aspect, based on period time. The temporal relational database with SQL: 2011 features have been accepted as a solution for kept data changes over time with insertion of new records or update old records. However, many problems have been emerged, the weakness of such Temporal RDBMSs in supporting complex data structures, user-defined data types and data persistence required by temporal object relational database. Furthermore, the reconstruction of temporal complexes objects split across relational tables is costly because its causes many join. ORDBs with time-varying features has addressed of these problems, which showing potential, because they have a relational base and append object concept, to enhance ORDB and give the correct description of records , we associate time at attribute. Temporal ORDB overcomes the disadvantage of data redundancy introduced when using temporal RDB to store the information.

The purpose of this paper is to propose a method for migrating bitemporal data from RDB based on SQL: 2011 standard, into Temporal ORDB integrating time-varying features. The method comprises three basic phases. In the first, we will examine the useful features of temporal data in order to create an RDB meta-Model, which holds the necessary elements for the correct description of temporal database. In the second step, the method takes the entire bitemporal relational database and stores it in a temporal structured table that contains several parameters, attributes, class, relationships, cardinalities, integrity constraint and time periods, in order to enrich and realize the schema translation (BTDM). The BTDM so obtained is converted into a temporal ORDB model, which treats complexes object and data semantic that can expressed in its metadata in the third step. The prototype has been developed to demonstrate the migration process, and TORDB queries example will be generated to valid our approach.

2 Related Work

Significant researches address the problem of numerous temporal models and query language. Atay compared interval-based attribute and tuple time stamped bitemporal data models, accesses and evaluated their usability using the same data and same queries for both approaches. According to this comparison, Petkoviç examined the performance implication for tuple and attribute timestamping, her test stored data using two different forms, and perform the 36 query on both.

A research work in [7] proposed a temporal object relational SQL language handling valid time at the attribute level in a temporal transparency environment his paper. The approach in [8] presented a database application development environment for both temporal and non-temporal using SQL: 2003 following the attribute timestamping. Comparison of three different storage models (OODB, ORDB, and XML) for the parametric temporal data model and estimate storage costs are discussed in [9].

Oracle added temporal processing to its database technologies. There are a few work deals with integrating temporal dimension to an existing DBMS. Sandro Radovanović evaluated performance of traditional relational DBMS (RDBMS) and temporal databases using Oracle 12c DBMS [10]. Verification of temporal data using valid time dimension support in Oracle is proposed in [11]. Ptrovic uses the most important temporal concepts to investigate their implementations in enterprise database systems such as Oracle, DB2 and teradata[12].

The ISO (international organization for standard) and IEC (International Electrotechnical Commission) committee, initiated o project to create a language extension to support temporal database, is given in [13]. The most important features in SQL: 2011 to create and manipulate temporal database implemented by IBMDB2, is discussed in [14].

From the all-overhead results, we conclude that most proposed solutions contain limited and simple rules compared to our work, notably regarding the scope and methodology. We believe that studies are based on time-variant data schema to make records of data, not even completed to provide a comprehensive data model, especially at the representation level. Furthermore, Some semantics aspects are not considered in the previous works .Our study is using the concept of semantic enrichment that give the possibility to understand the structure , meaning of temporal databases and construct schema translation which is enhanced by additional data semantic.

During our criticized analysis, we feel that some aspects of time variant data was overlooked in many works, which reflect that the challenge for those authors was only to cover the technical part of the storage, retrieval rather than on migration and gain from the offered advantages in temporal object relational, in order to reduce the redundancy of data by supporting the attribute timestamping.

The goal of our work is not mainly to create better temporal objet relational database from BTDM, but to afford a well arranged and a complete as possible transformation from temporal RDB into temporal ORDB, that can be a concrete reference for further investigations and works in this common area. we summarize and illustrates clearly the completeness of our mapping strategy, that include entities, objects, annotation, relationships between classes, attributes in their various forms, data types, value types, class constructs, constraint types, valid time period, Transaction period and much more.

We create our solution by combine several results from the existing methods and apply our enhancement using some semantics concepts. More precisely, we propose the rules that facilitate the transformation from Temporal RDB into temporal database based on ORDB using bitemporal data dimension. Therefore, this study presents a meta-model to define set of stereotype for the specification of new characterization of the bitemporal data associated with UML class diagrams, which simplify the creation of BTDM.

3 Process of Migration from TRDB into TORDB:

In this section, we outline the important phase for translation. In the fist, we provide the TRDB design, and then we will define the BTDM and TORDB model.

3.1 Creation of TRDB design :

The main purpose behind constructing a TRDB design is to simplify the comprehension of essential metadata stored in temporal databases. An efficient TRDB modeling overcomes the complications that occur during matching period time and keys in order to classify relations between classes, attribute, relationships and dimensions of time. An important advantage of TRDB design is that it identifies the migration of periods, therefore adding more semantics to a TRDBs metadata.

Consider the database shown in Fig1, which modeling the purchase orders administration. This model will be used in the examples presented along the paper. Primary keys are underlined in bold ex, and Foreign Keys are marked by "*".

-Customer (**IDcust**,name,city,street,phone,*OrderNo ,**vt-start,vt-end**,TT-start,TT-end)
- Company (***IDcust**,Type, taxes)
-Customer-Ass (**IDcust**,descpt,percent, **vt-start,vt-end**,TT-start,TT-end)
- Purchase-Order (**OrderNo**,tocity,tostreet,tozip,shippingdate, *lineNo, **vt-start,vt-end** TT-start,TT-end)
- OrderLine-Item (**lineNo**,*ProdNo , Quantity, **vt-start,vt-end**,TT-start,TT-end)
- Products (**ProdNo**,descpt,price, **vt-start,vt-end**,TT-start,TT-end)
- Store (**IDstore**, location,capacity,street,city)
- Stock (***ProdNo** , ***IDstore**, Quantity, **vt-start,vt-end**,TT-start,TT-end)

Figure1.Sample input temporal Relational database

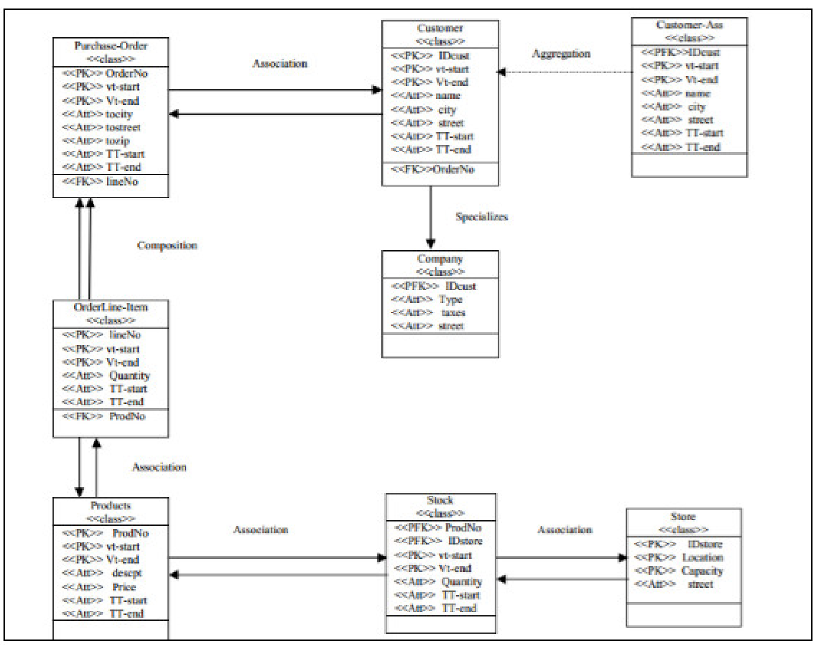

Figure 2: TRDB Meta-Model

The creation of TRDB design is the first step of the migration into TORDB. To do this, we define a new set of UML model elements that provides mechanisms that enable new kinds of modeling elements to be defined, and relate the information to new modeling elements. This is accomplished by integrating stereotype, constraints and tagged values.Fig2, illustrates the meta-model that is proposed for relational databases, which support SQL: 2011 standard.

3.2 Semantic enrichment of temporal relational Database: BTDM

Semantic Enrichment is a process of analyzing databases to understand their structure, and to make hidden semantics explicit [15].To enrich the semantic of temporal RDB schema, we have to extract its data semantic to be enriched and converted into a much enhanced BTDM. To do this task, the process starts by extracting the basic information about an existing temporal RDB schema, integrating relations names, Periods time, and Attributes properties. We assume that data dependencies are represented by primary keys and foreign keys, as for each foreign key value, there is a reference to an existing PK value. The next step is to identify the BTDM constructs based on classification of data and the relationships, which may be performed through data access. In the last step, the BTDM structure is generated.

3.3 Definition of BTMD

The BTDM is a representation of RDB using SQL: 2011 Standard, which is enriched with semantic data in order to provide a new type of tables describing the different classes, extracted from temporal RDB with the data necessary for the creation of temporal ORDB. This phase produces a data reference model that is designed to allow the exchange the schema and the sharing of information to reuse.

The BTDM is defined in our approach as a set of element:

BTDM :={ C| C := {Cn,Att,Rel,clas,bitDim }}

Where:

- Each class has a name Cn
- Att= denotes a set of Attributes of class C
- Att={a|a:=(N,T,L,NL,D,tag)} , where N is an attribute name , T is an attribute type, L: data length, NL: if the attribute accepts the parameter null or not(N/ NoN),Tag: primary key (PK)| foreign key(FK)| primary foreign key (PFK).
- REL: Relations BTDM each class C has a set of relationships with other classes ,where rel is defined in C with another class C’ REL={rel | rel:= RelType,DirC,Car } ,where RelType denotes a relationship, DirC is the name of C’ that interacts with C, Car means cardinalities describing the relationship.

  RelType supports five types of relationships, and accepts the following values:

  1. “Ass” for Association
  2. “Agg” for Aggregation
  3. “comp” for Composition
  4. “inher” and “inherBy”for Inheritance

- Clas: classification

  Classification divides classes into two different kinds of categories:

  1. Temporal class (TCls): class contains a varying time period. The time period during which a row is regarded as correctly reflecting reality by the user of database [16].

2. Simple class or No Temporal class (SCls): class without varying time attributes.

- BitDim: Bitemporal data dimension

  Each temporal class has a valid time period, transaction period or the both. BitDim is used to specify time variant attributes:

  BitDim={B | B=: {VT, TT}}

where:

VT: is valid time period, it can have the following values:

VT={VT_LB,VT_UB}, which specify the lower and upper bound of the valid time data value.

TT: is transaction time period, TT accepts two values:

TT ={TT_LB, TT_UB}, which specify also the lower and upper bound of the transaction time.

3.4 Generation of BTMD

In this paper, the BTMD is considered the interesting phase for the migration process which in the end generates the target scheme.

		Attribute					Rel			BitDim			
Cn	**Clas**	**N**	**T**	**L**	**NL**	**Tag**	**RelType**	**DirC**	**Car**	**VTL**	**VTU**	**TTL**	**TTU**
Customer	TCLS	IDcust	Number		NoN	PK	Ass	Purchase-Order	1				
		name	Varchar	25	NoN		Ihne by	Company	1				
		city	Varchar	25	NoN		Agg	Customer-ass	1...N				
		street	Varchar	25	NoN					Y	Y	Y	Y
		phone	Number		N								
		OrderNo	Number		NoN	FK							
		vt-start	Date		NoN	PK							
		Vt-end	Date		NoN	PK							
		TT-start	sysdate		NoN								
		TT-end	date		NoN								
Company	TCLS	IDcust	Number		NoN	PFK	Ihne	Customer	1				
		Type	Varchar	25	NoN					Y	Y	Y	Y
		taxes	Varchar	25	NoN								
Customer-Ass	TCLS	IDcust	Number		NoN	PFK	Agg	Customer	1				
		Descpt	Varchar	25	N								
		percent	Varchar	25	NoN								
		vt-start	Date		NoN	PK				Y	Y	Y	Y
		vt-end	Date		N	PK							
		TT-start	sysdate		NoN								

		TT-end	date		N								
Purchase -Order	TCLS	OrderNo	Number		NoN	PK	Ass	Customer	1…N				
		tocity	Varchar	25	NoN		Com	OrderLine-Item	1				
		tostreet	Varchar	25	NoN								
		tozip	Varchar	25	NoN					N	N	Y	Y
		lineNo	Number		NoN	FK							
		vt-start	Date		NoN	PK							
		Vt-end	date		NoN	PK							
		TT-start	sysdate		NoN								
		TT-end	date		NoN								
OrderLin e-Item	TCLS	lineNo	Number		NoN	PK	Com	Purchase-Order	1..N				
		ProdNo	Number		NoN	FK	Ass	products	1..N				
		Quantity	Number		NoN					Y	Y	Y	Y
		vt-start	Date		NoN	PK							
		vt-end	Date		N	PK							
		TT-start	sysdate		NoN								
		TT-end	date		N								
Products		ProdNo	Number		NoN	PK	Ass	OrderLine-Item	1				
		descpt	varchar		NoN		Ass	Stock	1..N				
		price	Number		NoN								
		vtstart	Date		NoN	PK				Y	Y	Y	Y
		vt-end	Date		N	PK							
		TT-start	sysdate		NoN								
		TT-end	date		N								
Store	SCLS	IDstore	Number		NoN	PK	Ass	Stock	1..N				
		Location	Varchar	25	NoN								
		Capacity	Number		NoN					N	N	N	N
		street	Varchar	25	NoN								
		city	Varchar	25	NoN								
Stock	TCLS	ProdNo	Number		NoN	PFK	Ass	Products	1				
		IDstore	Number		NoN	PFK	Ass	Store	1				
		Quantity	Number		NoN					Y	Y	Y	Y
		vt-start	Date		NoN	PK							
		Vt-end	Date		NoN	PK							
		TT-start	sysdate										
		TT-end	date										

Figure 3: Result of BTDM Generation

3.5 Definition and Identification of TORDB Model:

This step, presents the different elements composed a TORDB Model, which provide a complete description of temporal OR database. The TORDB model is defined as a set of temporal typed table based on temporal structured type TST for storing data.

Each ST consists of a set of non varying-time attributes and varying-time attributes defined as primitive collection data type or reference pointing to a specific ST or TST. In this paper, the varying time attribute can be a bitemporal attribute, which actually stored in nested table collection type. Each value assigned to the temporal attribute must respect the temporal data representation of a temporal column.

Definition of TORDB model: is denotes as three-tuples:

TORDB Model = {TTs, STs ,Tm}

Where: TTs is set of temporal and non temporal typed table, STs is a set of temporal structured type or simple structured type , and Tm is a time-varying Period. The sets TTs, STs and Tm are defined as follows:

- STs ={Sn , S, AT},where Sn is the name of a structured type , S is the super type of ST, and AT is a set of structured type's attributes:
  AT={A | A:={N,T,D,NL,BitT,M}},where N: is the name of attribute, T: means data type which can be primitive , UDT or reference. N: if the attribute accepts Null or not. D: default value. M: denotes if the AT is a single valued or collection valued. BitT: denotes if the attribute contains a bitemporal attribute is defined:
  BitT={(AT1,AT2,….,VTL,VTU,TTL,TTU)}.
- TTs ={typedtable|Ttable={TTn,STn,PK,Tp}}where TTn is the name of typed table, STn is the name of the structured type based upon which TT is defined, PK : primary key, TP: means if the TT is temporal or not.

3.6 Creating tables:

It can be able to produce temporal ORDB queries for relationships with oracle 12C which is formed by the temporal and no temporal queries. We use Oracle's concept of nested table to create the varying-time attributes. Example1 show the creation of the customer table and of all necessary auxiliary (object) types.

Example1:

```
Create type Bitemporal_T as object{
TT_LB date,
TT-UB date,
VT_LB date,
VT_UB date}/
Create type Bit_period is table of  Bitemporal_T;
Create type CustAss-T as object(
Type varchar(25),
Taxe varchar(25))
Create type PurchOrd_T as object {
```

```
OrderNo Number,
Tocity varchar(25),
Tostreet varchar(25),
Tozip varchar(25),
history  Bit_period}/
Create table PurchaseOrd_T of PurchOrd_T CONSTRAINT OrderNo-PK PRIMARY KEY(OrderNo), NESTED TABLE  Purch-history
STORE AS Bitemporal_tab;
Create type NT_PurchOrd as Object{
OrderNo Number,
VT_Start date,
VT_End date}/
Create type PurchaseOrder_Hist  is table of  NT_PurchOrd;
Create type Customer_T as Object{
IDcust Number,
Name varchar(25),
city varchar(25),
street varchar(25),
phone Number,
Customer_Association   CustAss-T,
Cust_history  Bit_period,
Purchase_Order  PurchaseOrder_Hist
}/
Create table Customer of Customer_T CONSTRAINT Cust-PK PRIMARY KEY(IDcust), NESTED TABLE Cust_history  STORE AS
Cust_tab, NESTED TABLE  Purchase_Order  STORE AS  PurchOrd_tab;
```

Figure4: TORDB Queries: Example

4 Conclusion

This work outlines the basics phases of migrating from RDB according to SQL: 2011 standard into ORDB including bitemporal data, with a simple and practical method to capture the relationships between different kinds of classes. Currently, no approach has proposed such a solution to extract data model from RDB implemented by SQL: 2011 aspect. This approach is superior to existing work as it generates the ORDB with bitemporal data, including the schema and data semantics, and it exploits the range of powerful features provided by SQL standard.

Our method exceeds the existing works as it generates the TORDB tables by creating a BTDM from a TRDB, and we use it as an input enriched with semantic data, and this last provides a TORDB Model to capture the characteristics of temporal and non temporal SQL query.

A forthcoming paper will propose an algorithm for converting method from TRDB into TORDB, with Bitemporal data that not requires any human interference.

REFERENCES

[1] Jaypalsinh A. Gohil, Dr. Prashant M. Dolia ."Checking and Verifying Temporal Data Validity using Valid Time Temporal Dimension and Queries In Oracle 12c", International Journal of Database Management Systems (IJDMS), Vol.7, No.4, August 2015.

[2] M. Kaufmann, P. M. Fischer, N. May, D. Kossmann: "Benchmarking Bitemporal Database Systems: Ready for the Future or Stuck in the Past?" . EDBT, 738-749, 2014

[3] D. Petković , "Performance Issues Concerning Storage of Time-Variant Data, Egyptian Computer Science Journal", Vol. 38 No, 2014.

[4] Atay, C. , " A Comparison of Attribute and Tuple Time Stamped Bitemporal Relational Data Models", Proc. of the Int. Conf. on Applied Computer Science, p: 479-489, 2010.

[5] D. Brdjanin , S Maric , Z S Pavkovic «On Suitability of Standard UML Notation for Relational Database Schema Representation » . Enterprise Business-Process and Information Systems Modeling , Vol 248, pp 399-413, 2016

[6] Dušan Petkovic. "Temporal Data in Relational Database Systems: A Comparison". New Advances in Information Systems and Technologies,Vol 444,pp 13-23. 2016

[7] Chau, V.T.N. and Chittayasothorn, S. "A Temporal Object Relational SQL Language with Attribute Timestamping in a Temporal Transparency Environment,"Data & Knowledge Engineering, (67), p. 331-361, 2008.

[8] T. Chau and S. Chittayasothorn, "A Temporal Compatible Object Relational Database System", The IEEE Southeastcon, 2007.

[9] Noh, SY., Gadia, S.K. & Jang, H. J. Cent, "Comparisons of three data storage models in parametric temporal databases". Journal of central south university , vol 20: 1919_1927, 2013.

[10] Slavimir Vesić, Slađan Babarogić Nenad Aničić, "Use of the Temporal Concepts in Transaction Database", PROC. SYMORG, p 850-857 , 2014.

[11] ISO/IEC 9075-2:2011, Information technology-Database languages-SQL - Part 2: Foundation (SQL/Foundation), 2011.

[12] Kulkarni, K., Michels, J. "Temporal Features in SQL: 2011", SIGMOD Records, Vol. 41, No. 3, 2012.

[13] Maatuk, A., Ali, M. A. and Rossiter, N.: "Semantic Enrichment: The First Phase of Relational Database Migration". In CIS2E '08, 6pp, Bridgeport, USA, 2008.