COMPARISON OF DIFFERENT TEMPORAL DATA WAREHOUSES APPROACHES

Georgia GARANI
Technological Educational Institute of Larisa, Department of Computer Science and Engineering, Greece
garani@teilar.gr

Canan Eren ATAY
Dokuz Eylul University, Department of Computer Engineering, Izmir, Turkey
canan@cs.deu.edu.tr

Abstract: Data warehouses are mainly used for business data analysis by querying and reporting huge collections of data. For the management of historical data, temporal data warehouses have been developed. Two current approaches for dealing with temporal data in data warehouses are compared in this paper, Object-Relational Temporal Data Warehouse (O-RTDW) model and Starnest Temporal Data Warehouse (S-TDW) model. The O-RTDW model enables data values to be associated with facts, and specifies when facts are valid, thereby providing a complete history of the data values and their changes. To accurately and completely store all data changes, the valid time should be kept at the attribute level. On the other hand, the S-TDW model uses the starnest schema for the modeling of time-varying data in dimensions. The temporal starnest schema expresses naturally hierarchy levels by the clustering of data in nested tables, with result the description of aggregation levels for a dimension in a natural way. By comparing these two temporal data warehouses models, object-oriented and nesting approaches are also compared and evaluated.

Keywords: Data Warehouse, Temporal Data Warehouse, Object-Relational Model, Starnest Schema

Introduction

Data warehouses play a critical part in the success of businesses. Decision support systems, data mining, business analysis, forecasting and product line analysis are all good examples of where data warehouses can be used.

Two IBM researchers, Barry Devlin and Paul Murphy, introduced the term 'Business Data Warehouse' in 1988 (Devlin and Murphy, 1988). It was described as a 'single logical storehouse of all the information used to report on the business'. In 1990, Ralph Kimball introduced Red Brick Warehouse, a database management system specifically for data warehousing. In the following year, 1991, a software for developing a data warehouse was built by Bill Inmon, the Prism Warehouse Manager.

Since then, a data warehouse (DW) is considered to be the main component of every business intelligence environment. It is mainly used for business data analysis by querying and reporting huge collections of data. Data in a DW must be stored in a way that is secure, reliable, easy to retrieve and to manage.

According to Bill Inmon (Inmon, 2002) a DW is "a collection of subject-oriented, integrated, non-volatile and time-variant data to support management's decisions". The non-volatile and time-variant data features of data warehousing suggest that it should allow changes to the data values without overwriting the existing values.

The main characteristics of a DW are given briefly below: A DW stores current and historical data. Stored data cannot change. Insertions, deletions and updates do not take place in a DW. Data is used only for querying and consequently, it is essential that querying performance is as high as possible.

For the management of historical data, temporal DWs have been developed. Temporal DWs use the knowledge obtained from temporal databases for the treatment of time domain. Temporal databases have built-in support for representing and managing information varying over time. They are divided in three categories, valid time databases, transaction time databases and bitemporal databases according to the type of time they support, valid time, transaction time or both respectively. Valid-time expresses the time when a fact is true in the real world and transaction-time represents the time when a fact is current in the database.

Time can be added at the tuple level in a relation and this relation is called tuple timestamping relation or at the attribute level, called attribute timestamping relation, when individual time varying attributes are timestamped. SQL has also been extended to support features introduced in temporal databases.

In this paper, two temporal DWs models are evaluated and compared, the Object-Relational Temporal Data Warehouse (O-RTDW) model and the Starnest Temporal Data Warehouse (S-TDW) model.

The O-RTDW model uses the object-relational approach for the representation of time-varying data. This model inherently groups related facts into a single row, hence allowing changes to the data value and timestamps to be kept together. Dimensions may have levels. Multivalued attributes of data type T_ATOM are used for temporal

support of a level attribute. The levels and dimensions can have many time-varying attributes stored as nested tables.
The S-TDW model uses the starnest schema for storing dimension's hierarchy. The starnest schema proposed in Garani and Helmer (2012) is based on the nested approach, where hierarchies are represented as nested tables. The starnest schema is extended in Garani, Adam and Ventzas (2016) to support time. In the temporal starnest schema every temporal dimension, i.e., dimension table dependent on time, contains time attributes for the support of time. Several queries expressed in SQL are implemented and executed using Oracle 11g in both approaches. Same data and queries are used. Execution time and result data are compared and useful results are obtained.
The remainder of the paper is structured as follows. Firstly, related research work is discussed. Afterwards, the two temporal DW models, O-RTDW model and S-TDW model are presented. The hospital's admission temporal DW case study is described in both models. Implementation issues are discussed and finally, last section concludes the paper.

Related Work

Temporal DWs have been the subject of research in recent years. In what follows, a brief description of research studies on this field is presented.
The term 'slowly changing dimensions' was introduced by Kimball (Kimball, 1996). It was used for storing slowly changing historical data. Three different techniques for dealing with attributes changing over time were proposed, either by overwriting the value, adding a dimension row, or adding a dimension column as well as a number of hybrid methods. In this research work, schema evolution and dimension updates have not been generally considered.
In Bliujute et. al. (1998) the temporal star schema proposed does not include a time dimension. Instead, every row appeared either in fact or dimension tables is timestamped which causes the increase of redundancy.
A bitemporal DW model proposed in Koncilia (2003) is an extended version of the COMET metamodel (Eder, Koncilia & Morzy, 2002). It supports valid time and transaction time both at instance and schema levels. The model allows all possible changes of schema and structure of a DW with the introduction of suitable transformation functions.
Malinowski and Zimanyi (2006) extended the conceptual multidimensional MultiDimER model for supporting valid time and transaction time. The model is suitable for representing time-varying levels, attributes and hierarchies. It distinguishes time variant elements from time invariant elements and treats them separately. The proposed model also supports DW loading time.
The bitemporal versioning of multidimensional schemas is used for defined the conceptual evolution DW model in Rechy-Ramírez and Edgard (2006). The model supports many versions with the same valid time and different transaction times. Sixteen schema evolution operators are defined for dimensions and cubes and a SQL-like language for the proposed model is also presented.
A review of issues associated to temporal data warehousing is presented in Golfarelli and Rizzi (2009). Three different topics are distinguished, handling changes in the DW, handling data changes in the data mart and handling schema changes in the data mart.
A graph based temporal semi-structured DW is presented in Combi, Oliboni & Pozzi (2009) and an appropriate query language for the modeling and querying of temporal data.
A multiversion DW management system called SysVersDW is defined in Turki, Jedidi & Bouaziz (2010). Versioning of schema and instance components is supported in the model. A number of integrity constraints are also included for data and structure consistency. The constraints presented are classified in three classes, structural constraints, temporal constraints and versioning constraints.
A new schema proposed in Garani and Helmer (2012), the starnest schema, is based on the nested approach. It is extended in (Garani, Adam, and Ventzas, 2014) for supporting time. The key component of the temporal starnest schema is the inclusion of time attributes in every dimension table depended on time.
A bitemporal DW model where both valid time and transaction time are attached to attributes is introduced by Atay and Alp in (Atay and Alp, 2016). DW objects and cubes are created with multidimensional bitemporal relational database.

Temporal Data Warehouse Models

For the management of historical data, temporal DWs have been developed for describing information changing over time. Two of the most recent proposed approaches are discussed and compared in this research work, O-RTDW and S-TDW models.

Object-Relational Temporal Data Warehouse (O-RTDW) model

The O-RTDW model uses the specifications provided by the Object-Relational model. It consists of a fact table and several dimension tables connected to it. A dimension is composed of one or more levels, whereas each level belongs to only one dimension. O-RTDW enables data values to be associated with facts and specifies when facts were valid, thereby providing a complete history of data values and their changes. To accurately and completely store all data changes, the valid time should be kept at the attribute level. Attributes can be temporal or non-temporal. Temporal attributes consist of temporal atoms (T-ATOM). A temporal atom is defined as <valid time, value> where valid time component can be applied as a time point, a time interval, or a temporal element. Therefore, a temporal atom in the form of <[VT_{lb}, VT_{ub}), V> represents valid time lower bound as VT_{lb}, valid time upper bound as VT_{ub} and data value as V, respectively. In Figure 1 the conceptual model for the O-RTDW model is shown.

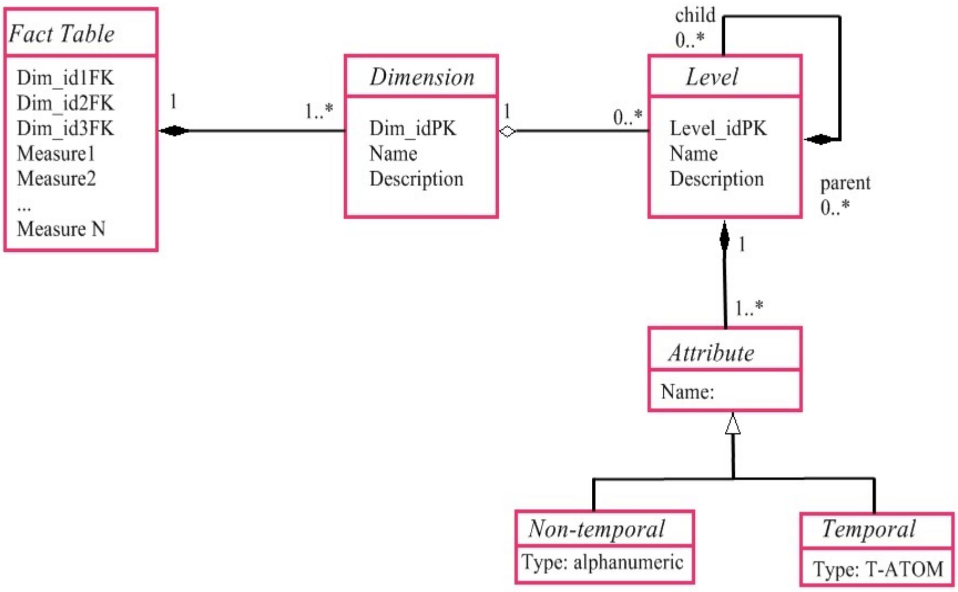

Figure 1. The conceptual model for the O-RTDW

Starnest Temporal Data Warehouse (S-TDW) model

S-TDW model uses the temporal starnest schema (Garani, Adam and Ventzas, 2014) for the modeling of time-varying data in dimensions. The starnest schema forms the integration of the star and snowflake schemas (Garani and Helmer, 2012). It expresses naturally hierarchy levels by the clustering of data in nested tables, with result the description of aggregation levels for a dimension in a natural way. It consists of a temporal fact table and a number of dimension tables. A temporal fact table can be timestamped by adding one or two time attributes representing a time point or a time interval respectively.

Dimension tables can also be timestamped similarly. Timestamped dimension tables are called temporal dimension tables. Temporal dimension tables are nested, since time attributes are inserted in a dimension in a nested way, where more detailed attributes are nested inside less detailed attributes. Therefore, dimension tables are not normalized. In each temporal dimension two valid time attributes are included, the start and stop time points of the corresponding time interval.

The fact table is linked to dimension tables with one to many relationships by foreign key attributes with a reference to the most detailed hierarchical attribute of each dimension. The conceptual model for the S-TDW model is shown in Figure 2.

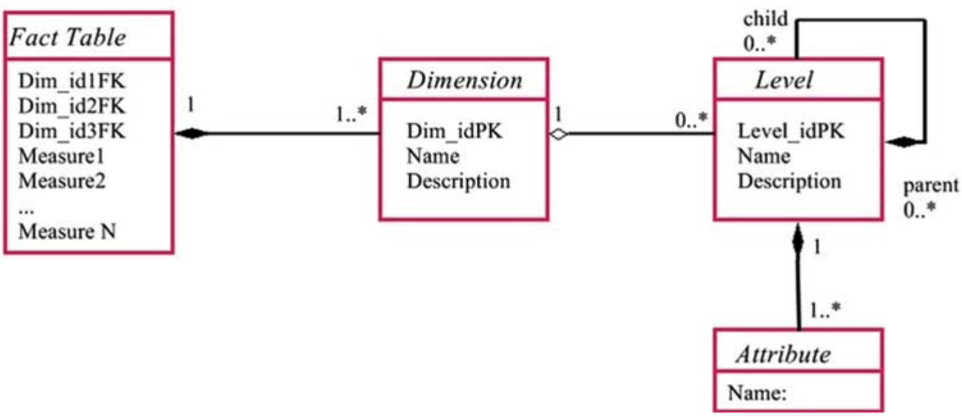

Figure 2. The conceptual model for the S-TDW

The main differences between O-RTDW and S-TDW models are presented in Table 1.

Table 1: The main differences between O-RTDW and S-TDW models

O-RTDW model	S-TDW model
Snowflake schema	Starnest schema
Object-Oriented approach	Nested approach
Temporal atom	Time attribute
Time interval	Time point (Start/Stop)

The Hospital's Admission Temporal Data Warehouse Case Study

A hospital's admission temporal DW has been used for the comparison of O-RTDW and S-TDW models. The hospital's admission temporal DW concerns the admission in the hospital of patients who suffer from different diseases and therefore, have different diagnoses and treatments.
In Figure 3 the schema of the O-RTDW model is shown. The schema is represented in snowflake format where dimension tables are split up into smaller normalized tables that express each dimension's hierarchy. Transitive functional dependencies do not exist.

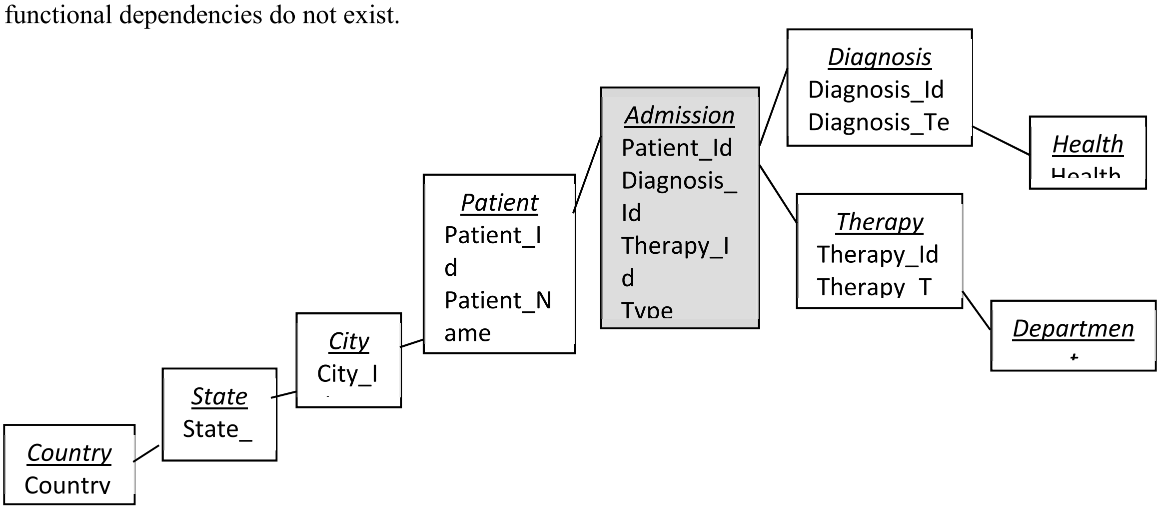

Figure 3. The schema for the O-RTDW model

Figure 4 presents the schema for the S-TDW model of the hospital's admission temporal DW. S-TDW model uses the starnest schema at the logical design. In the starnest schema a dimension's hierarchy is expressed as a nested table where hierarchy levels are expressed naturally and attributes can easily be associated within their corresponding levels. Each dimension table has a hierarchical attribute which is referred to a foreign key attribute of the fact table. The above mentioned hierarchical attribute is located in the most nested level of the dimension table.

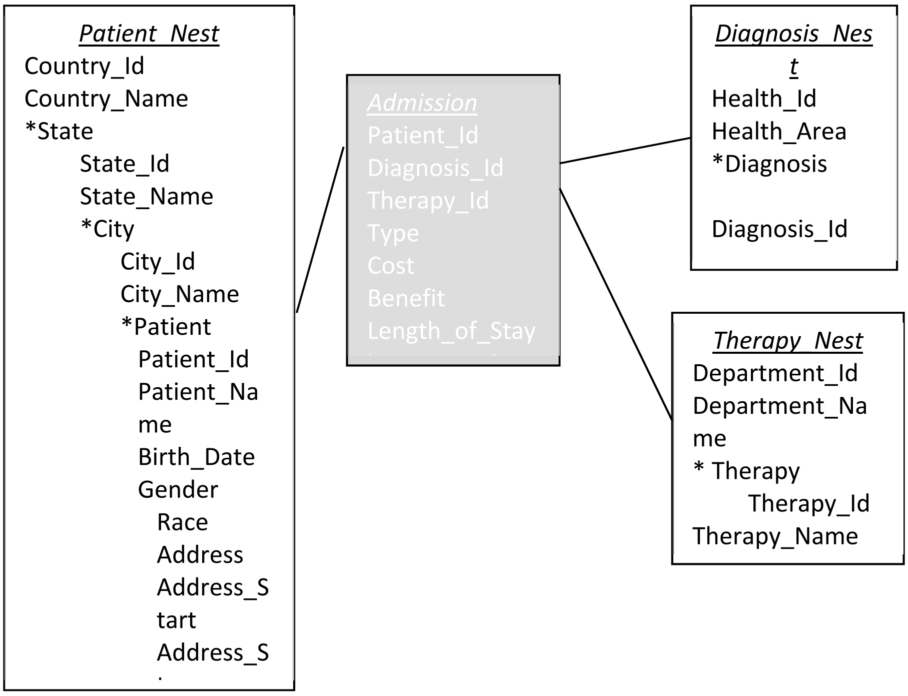

Figure 4. The schema for the S-TDW model

Figure 5 shows an instance of Patient dimension in the O-RTDW model. Address_Temporal attribute, as its name denotes, is temporal. It contains temporal atoms of the form <[VT_{lb}, VT_{ub}), V> where VT_{ub} can be either a time point representing the valid time upper bound of the corresponding time interval or ‘now’ denoting present time instant which increases as time advances.

PatientID	FirstName	LastName	BirthDate	Gender	Race	Address_Temporal	CityID
101	JACKY	WANDA	03.03.1980	F	White	< [01.30.1955, now], “625 13th Avenue”>	45
102	TOM	BROWN	01.30.1989	M	White	{< [03.03.1980, 06.06.2005), “55 Hamilton Avenue”>, < [06.06.2005, 01.01.2008), “7 Leona Street”>, < [01.01.2008, now], “96 Market Street”>}	11
103	BILL	LAWRENCE	04.25.1977	F	Asian	< [01.30.1955, now], “111 Madison Avenue” >	18
104	AMY	ANGEL	10.05.1985	F	Black	{<[04.25.1977, 11.12.2000), “5 Valley Road”>, < [11.12.2000, now], “6255 Broadway”>}	01

Figure 5. An instance of Patient dimension in the O-RTDW model

An instance of Patient dimension in the S-TDW model is presented in Figure 6. Patient dimension is a temporal dimension table since it is dependent on time. It is also nested since hierarchies in dimensions are presented as nested tables. Therefore, a temporal nested dimension table consists of hierarchical attributes, dimensional attributes and time attributes which can also be nested inside less detailed attributes. In Patient dimension *AddressDetails is a temporal nested attribute consisting of three attributes, one atomic, Address and two time attributes, AddressStart and AddressStop indicating the start and stop points of the time interval during which the corresponding address is valid.

Dimension_key	Country_Id	CountryName	*State											
			State_Id	StateName	*City									
					City_Id	CityName	*Patient							
							Patient_Id	PatientName	BirthDate	Gender	Race	*AddressDetails		
												Address	AddressStart	AddressStop
D1	C1	USA	S1	Iowa	45	Iowa City	101	Jacky Wanda	03.03.1980	F	White	625, 13th Avenue	01.30.1955	now
					11	Waterloo	102	Tom Brown	01.30.1989	M	White	55, Hamilton Avenue	03.03.1980	06.06.2005
												7, Leona Street	06.06.2005	01.01.2008
												96, Market Street	01.01.2008	now
			S2	Alabama	18	Birmingham	103	Bill Lawrence	04.25.1977	F	Asian	111, Madison Avenue	01.30.1955	now

Figure 6. An instance of Patient dimension in the S-TDW model

Implementation

All queries have been executed on an Intel(R) Core(TM) i5 processor, running at 2.3 GHz, with 2.5 GB ram memory, under Windows 7 (32bit). The DW was built in Oracle Data Warehouse builder 11.2.0.1 and Oracle SQL Developer 4.0.3 was used.

The hospital's admission temporal DW consists of 9 tables in the O-RTDW model and 4 tables in the S-TDW model. Consequently, the number of joins in the O-RTDW model is much higher than in the S-TDW model. Relationship between a dimension and the number of tables it contains is one-to-one in the S-TDW model compared to one-to-many in the O-RTDW model. Tables in both models do not contain any data redundancy. Implementation of O-RTDW approach is platform independent in comparison to S-TDW approach which is platform dependent. The disk space required is more than three times higher in the O-RTDW approach than the S-TDW approach as it is shown in Table 2. In particular, the total space for the O-RTDW model is 14.3125 Mb in comparison to the S-TDW model where it is 4.1875 Mb.

Similarly, the number of rows is much higher in the O-RTDW approach than in the S-TDW model. Specifically, O-RTDW contains about 200,000 rows while S-TDW contains about 50,000 rows.

Table 2: O-RTDW model

Table name	Schema name	Size (Mb)	Number of rows
ADMISSION_O_CUBE_TAB	Admission	4.0	48,000
ADM_DEPARTMENTS	Department	0.0625	502
THERAPY_DIMENSION_O_TABLE	Therapy	3.0	50,200
ADM_HEALTH	Health	0.0625	501
DIAGNOSIS_DIMENSION_O_TABLE	Diagnosis	3.0	50,100
ADM_COUNTRIES	Country	0.0625	10
ADM_STATES	State	0.0625	100
ADM_CITIES	City	0.0625	1,000
PATIENT_DIMENSION_O_TABLE	Patient	4.0	50,000

Table 3: S-TDW model

Table name	Schema name	Size (Mb)	Number of rows
ADMISSION_N_CUBE_TAB	Admission_Nest	4.0	48,000
THERAPY_DIMENSION_TABLE	Therapy _Nest	0.0625	502
DIAGNOSIS_DIMENSION_TABLE	Diagnosis_Nest	0.0625	501
P_PATIENT_DIMENSION_TABLE	Patient_Nest	0.0625	10

Five different temporal and non-temporal queries are presented below in SQL. The same queries are expressed in both approaches and compared.

Query 1:
Which diagnoses have the same treatment? (non temporal)

O-RTDW model

```
SELECT Diagnosis1.Value AS Diagnosis1, Diagnosis2.Value AS Diagnosis2
FROM ADMISSION_O_CUBE_TAB A1, ADMISSION_O_CUBE_TAB A2,
DIAGNOSIS_DIMENSION_O_TABLE D1, DIAGNOSIS_DIMENSION_O_TABLE D2,
Table(D1.Diagnosis_Temporal) Diagnosis1,
Table(D2.Diagnosis_Temporal) Diagnosis2
WHERE D1.Diagnosis_Id < D2.Diagnosis_Id
```


AND A1.Therapy_Id = A2.Therapy_Id
AND D1.Diagnosis_Id = A1.Diagnosis_Id
AND D2.Diagnosis_Id = A2.Diagnosis_Id

S-TDW model
SELECT V1.DIAGNOSIS_NAME, V2.DIAGNOSIS_NAME
FROM ADMISSION_N_CUBE_TAB A1, ADMISSION_N_CUBE_TAB A2,
DIAGNOSIS_DIMENSION_TABLE D1, DIAGNOSIS_DIMENSION_TABLE D2,
Table(D1.Diagnosis) V1, Table(D2.Diagnosis) V2
WHERE V1.Diagnosis_Id < V2.Diagnosis_Id
AND A1.Therapy_Id = A2.Therapy_Id
AND V1.Diagnosis_Id = A1.Diagnosis_Id
AND V2.Diagnosis_Id = A2.Diagnosis_Id

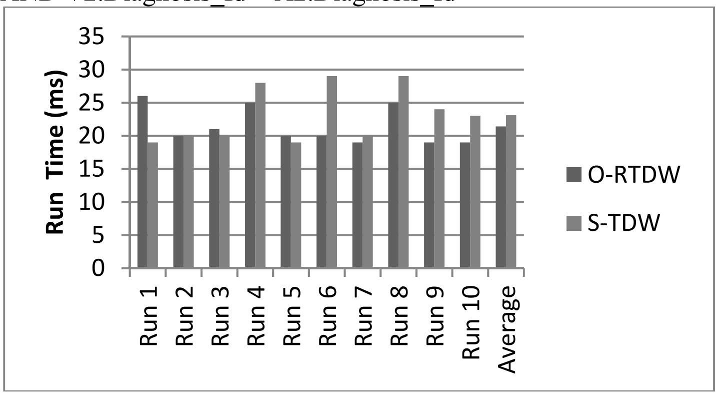

Figure 7. Query 1 run comparison chart

Query 2:
What is the average number of days for therapy in each department? (temporal)

O-RTDW model
SELECT Department_Id, AVG(Therapy.VALID_TIME_UB- Therapy.VALID_TIME_LB)
FROM THERAPY_DIMENSION_O_TABLE T, Table(T.Therapy_Temporal) THERAPY
GROUP BY Department_Id

S-TDW model
SELECT Department_Id, AVG(H.Therapy_Stop - H.Therapy_Start)
FROM Therapy_Dimension_Table T, Table(Therapy) H
GROUP BY Department_Id

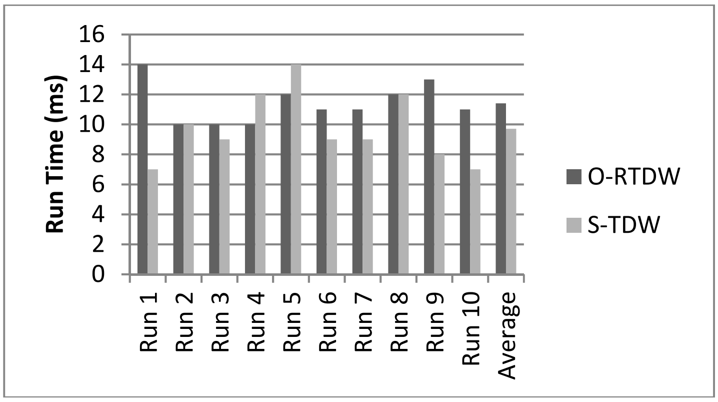

Figure 8. Query 2 run comparison chart

Query 3:
For each patient find the amount he/she paid for each admission and where he/she lived at that time. (temporal)

O-RTDW model
```
SELECT A.Patient_Id, A.Cost, ADDRESS1.Value
FROM ADMISSION_O_CUBE_TAB A, PATIENT_DIMENSION_O_TABLE P,
TABLE(P.Address_Temporal) ADDRESS1
WHERE ADDRESS1.VALID_TIME_LB <= A.Admission_Time
AND ADDRESS1.VALID_TIME_UB >= A.Admission_Time
AND A.Patient_Id = P.Patient_Id
ORDER BY A.Patient_Id
```

S-TDW model
```
SELECT A.Patient_Id, A.Cost, F.Address
FROM P_PATIENT_DIMENSION_TABLE P, ADMISSION_N_CUBE_TAB A,
Table(P.State) S, Table(S.City) C, Table(C.Patient) F
WHERE F.Address_Start <= A.Admission_Time
AND F.Address_Stop >= A.Admission_Time
AND F.Patient_Id = A.Patient_Id
ORDER BY A.Patient_Id
```

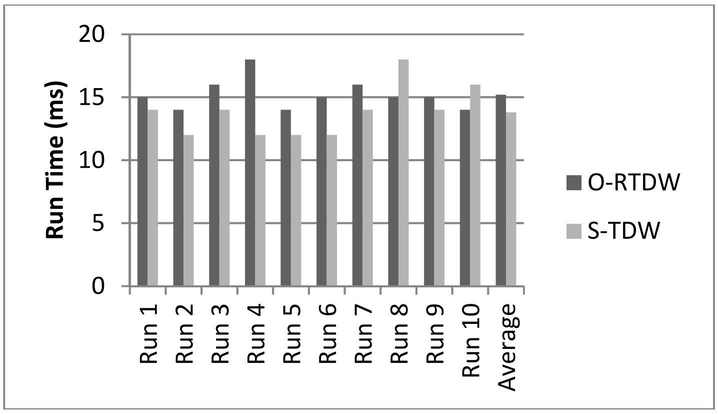

Figure 9. Query 3 run comparison chart

Query 4:
Which patients had cancer at New York in 2013? (temporal)

O-RTDW model
```
SELECT P.Patient_Name
FROM DIAGNOSIS_DIMENSION_O_TABLE D, ADMISSION_O_CUBE_TAB A,
Patient_Dimension_o_table P, ADM_CITIES C, TABLE(D.Diagnosis_Temporal) DIAGNOSIS1
WHERE D.description_d LIKE '%Cancer%'
AND A.Diagnosis_Id= D.Diagnosis_Id
AND A.Patient_Id=P.Patient_Id
AND C.City_Id=P.City_Id
AND   C.City_Name='New York'
AND ( EXTRACT(YEAR FROM DIAGNOSIS1.VALID_TIME_UB) = 2013
      OR EXTRACT(YEAR FROM DIAGNOSIS1.VALID_TIME_LB) = 2013 )
```

S-TDW model
```
SELECT F.Patient_Name
FROM   DIAGNOSIS_DIMENSION_TABLE D, ADMISSION_N_CUBE_TAB A,
P_PATIENT_DIMENSION_TABLE P, table(Diagnosis)   V, table(State) S, table(S.City) C,   table(C.Patient) F
WHERE V.description_d LIKE '%Cancer%'
AND V.Diagnosis_Id= A.Diagnosis_Id
AND A.Patient_Id=F.Patient_Id
AND C.City_Name='New York'
AND ( EXTRACT(YEAR FROM V.Diagnosis_Start) = 2013
```


OR EXTRACT(YEAR FROM V.Diagnosis_Stop) = 2013)

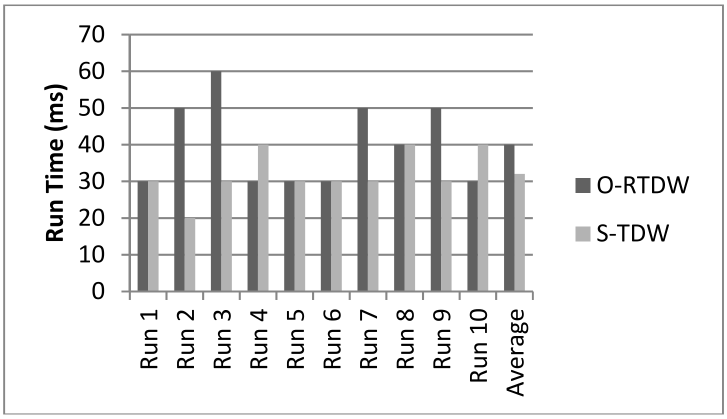

Figure 10. Query 4 run comparison chart

Query 5:
Which patients lived at the same city at the same time and had the same diagnosis? (temporal)

O-RTDW model

```
SELECT P1.Patient_Name, P2.Patient_Name, P1.City_Id, A1.Diagnosis_Id
FROM ADMISSION_O_CUBE_TAB A1, ADMISSION_O_CUBE_TAB A2, Patient_Dimension_o_table P1,
TABLE(P1.Address_Temporal) ADDRESS1, Patient_Dimension_o_table P2, TABLE(P2.Address_Temporal)
ADDRESS2
WHERE P1.Patient_ID < P2.Patient_ID
AND P1.City_Id = P2.City_Id
AND ((ADDRESS1.VALID_TIME_LB <= ADDRESS2.VALID_TIME_LB
          AND ADDRESS2.VALID_TIME_LB <= ADDRESS1.VALID_TIME_UB)
     OR (ADDRESS2.VALID_TIME_LB <= ADDRESS1.VALID_TIME_LB
          AND ADDRESS1.VALID_TIME_LB <= ADDRESS2.VALID_TIME_UB))
AND A1.Patient_Id = P1.Patient_Id
AND A2.Patient_Id = P2.Patient_Id
AND A1.Diagnosis_Id = A2.Diagnosis_Id
```

S-TDW model

```
SELECT F1.Patient_Name, F2.Patient_Name, C1.City_Id, A1.Diagnosis_Id
FROM P_PATIENT_DIMENSION_TABLE P1, Table(P1.State) S1, Table(S1.City) C1, Table(C1.Patient) F1,
P_PATIENT_DIMENSION_TABLE P2, Table(P2.State) S2, Table(S2.City) C2, Table(C2.Patient) F2,
ADMISSION_N_CUBE_TAB A1, ADMISSION_N_CUBE_TAB A2
WHERE   F1.Patient_Id < F2.Patient_Id
AND C1.City_Id = C2.City_Id
AND ((F1.Address_Start <= F2.Address_Start AND F2.Address_Start <= F1.Address_Stop)
   OR (F2.Address_Start <= F1.Address_Start AND F1.Address_Start <= F2.Address_Stop))
AND A1.Patient_Id=F1.Patient_Id
AND A2.Patient_Id = F2.Patient_Id
AND A1.Diagnosis_Id = A2.Diagnosis_Id
```

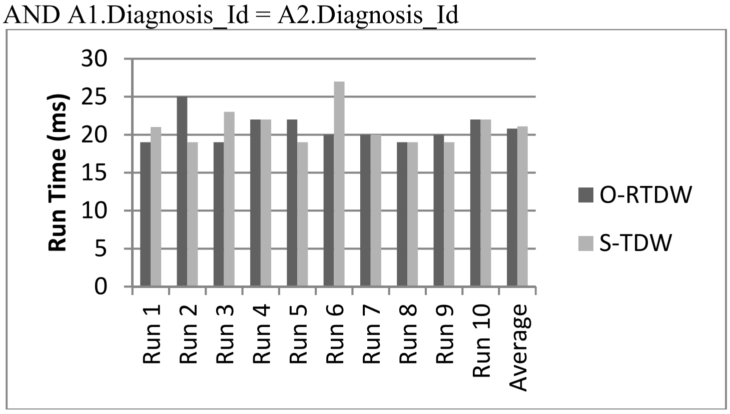

Figure 11. Query 5 run comparison chart

Overall, run times of the two approaches are similar, though S-TDW model is a little bit faster than O-RTDW as shown in Figure 12.

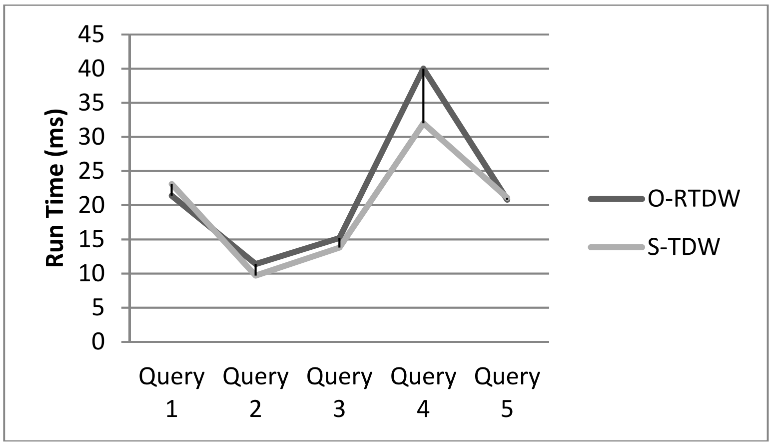

Figure 12. Queries' average run time comparison chart

Conclusion

Two current approaches for dealing with temporal data in DWs have been evaluated and compared in this research work, Object-Relational Temporal Data Warehouse (O-RTDW) model and Starnest Temporal Data Warehouse (S-TDW) model. Results showed that the S-TDW model requires significantly less space and it is a little faster on average than the O-RTDW model.

Future work includes implementation of optimization techniques for efficient evaluation of complex temporal nested queries, and the addition of transaction time to the implemented model.

Acknowledgements

The authors would like to thank George Tsaprazlis, undergraduate student from the Department of Computer Science and Engineering of the Technological Educational Institute of Thessaly, who helped with the implementation of this research work during his diploma thesis.

References

Atay, C. E. & Alp, G. (2016). Modeling and Querying Multidimensional Bitemporal Data Warehouses. *International Journal of Computer and Communication Engineering*, 5(2) (pp.110-119). San Bernardino, California : International Academy Publishing.

Bliujute, R., Saltenis, S., Slivinskas, G. & Jensen, C.S. (1998). Systematic Change Management in Dimensional Data Warehousing. In *Proceedings of the 3rd International Baltic Workshop on Databases and Information Systems*, Riga, Latvia (pp.27–41). Riga, Latvia: Latvian Academic Library.

Combi, C., Oliboni, B. & Pozzi, G. (2009). Modeling and Querying Temporal Semistructured Data Warehouses. In Kozielski, S. and Wrembel, R. (Eds.), *New Trends in Data Warehousing and Data Analysis, Annals of Information Systems*, 3 (pp.299–324). New York, USA: Springer.

Devlin, B. & Murphy, P.T. (1988). An Architecture for a Business and Information System. *IBM Systems Journal* 27(1) (pp.60-80). Yorktown Heights, New York, USA: IBM.

Eder, J., Koncilia, C. & Morzy, T. (2002). The COMET Metamodel for Temporal Data Warehouses. In *Proceedings of the 14th International Conference on Advanced Information Systems Engineering (CAiSE)*, Toronto, Canada (pp.83-99). Lecture Notes in Computer Science, Berlin-Heidelberg, Germany: Springer-Verlag.

Garani, G., Adam, G.K. & Ventzas, D. (2016). Temporal Data Warehouse Logical Modeling, *International Journal of Data Mining, Modelling and Management*, 8(2) (pp.144-159). Olney, Bucks, UK: Inderscience Publishers.

Garani, G. & Helmer, S. (2012). Integrating Star and Snowflake Schemas in Data Warehouses. *International Journal of Data Warehousing and Mining* 8(4) (pp.22-40). Hershey, Pennsylvania, USA : IGI Global.

Golfarelli, M. & Rizzi, S. (2009). A Survey on Temporal Data Warehousing. *International Journal of Data Warehousing & Mining* 5(1) (pp.1-17). Hershey, Pennsylvania, USA : IGI Global.

Inmon, W. (2002). *Building the Data Warehouse*. Indianapolis, Indiana, USA: John Wiley & Sons Publishers.

Kimball, R. (1996). *The Data Warehouse ToolKit*. Indianapolis, Indiana, USA: John Wiley & Sons Publishers.

Koncilia, C. (2003). A Bi-Temporal Data Warehouse Model. In *Proceedings of the 15th International Conference* *on Advanced Information Systems Engineering (CAiSE)*, Klagenfurt, Austria (pp.77-80). Lecture Notes in Computer Science, Berlin-Heidelberg, Germany: Springer-Verlag.

Malinowski, E. & Zimányi, E. (2006). A Conceptual Solution for Representing Time in Data Warehouse Dimensions. *In Proceedings of the 3rd Asia-Pacific Conference on Conceptual Modeling (APCCM)*, Hobbart, Australia (pp.45-54). Darlinghurst, Australia: Australian Computer Society.

Rechy-Ramírez, E.-J. & Edgard, B.-G. (2006). A Model and Language for Bitemporal Schema Versioning in Data Warehouses. In *Proceedings of the 15th International Conference on Computing (CIC)*, Mexico City, Mexico (pp.309-314). Los Alamitos, California, USA:IEEE Computer Society.

Turki, I.Z., Jedidi, F.G. & Bouaziz, R. (2010). Multiversion Data Warehouse Constraints. In *Proceedings of the 13th ACM International Workshop on Data Warehousing and OLAP (DOLAP)* (pp.11-18). New York, USA: ACM.

