

A Logical Temporal Relational Data Model

Nadeem MAHMOOD¹, Aqil BURNEY² and Kamran AHSAN³

^{1,2} Department of Computer Science (UBIT), University of Karachi,
Karachi-75270, Pakistan

³ Faculty of Computing, Engineering and Technology & Centre for Ageing and Mental Health,
Staffordshire University, Stafford, United Kingdom

Abstract

Time is one of the most difficult aspects to handle in real world applications such as database systems. Relational database management systems proposed by Codd offer very little built-in query language support for temporal data management. The model itself incorporates neither the concept of time nor any theory of temporal semantics. Many temporal extensions of the relational model have been proposed and some of them are also implemented. This paper offers a brief introduction to temporal database research. We propose a conceptual model for handling time varying attributes in the relational database model with minimal temporal attributes.

Keywords: Temporal Data Management, Temporal Semantics, Time Varying Attributes, Relational Database

1. Introduction

The relational model [10] is based on a brand of mathematics called relational algebra. Codd used this concept to manage huge amounts of data very effectively. Codd and others have extended application of this notion to apply to database design. As a consequence they were able to take advantage of the power of mathematical abstraction and the expressiveness of mathematical notation to develop a simple but powerful structure for databases [19].

A relation is a set of tuple(s), and a well-defined structure, called a relational schema. The set of all relation schemas of a database and the database structure is called the database schema. A database schema [19] is a design, which lays the foundation on which the database and the applications are built. A relation has to be in first normal form (FNF) [12][20], which means that the domains of the attributes in its schema may only be of scalar data types. In other words, a relation can be considered as a subset of the Cartesian product of all the attribute domains contained in its schema.

The relational data model only supports functionality to access a single state (the most recent one) of the real world,

called a snapshot. The transition from one database state to another (updates) thereby giving up the old state. There exist, however, many application domains which need to have access not only to the most recent state, but also to past and even future states, and the notion of data consistency must be extended to cover all of these database states. Due to the FNF assumption in the relational model, there is a restriction in expressing the data structures. To overcome this drawback, the relational model has been extended to support, non first normal form (NFNF) [12][20] or nested relations [34][43].

Many applications in the real world require management of time varying data such as financial applications, inventory systems, insurance applications, reservation systems, health care management systems, medical databases and decision support systems. Efforts to incorporate the temporal domain into database management system have been ongoing for more than a decade and dozens of temporal models have been proposed [2], [16], [29], [41] and a few of them have been implemented [1], [7], [26],[33], [37], [38].

In the first section we will discuss the important notions of time. It includes the concept of time point or time interval. It is also important to discuss the difference between FNF relations and NFNF relations. Time is used to distinguish between past, present or future states. The recording of time allows the identification when facts are true in the modeled reality (valid time) [12], [20] or when facts are current in the database (transaction time) [12], [20]. Another important concept is time stamping, which can be done with either tuples or attributes in relations.

One area of continuing research interest is the development of a temporal data model capable of representing the temporal dimension of the mini world. The primary focus has been to extend the relational data model to support time-varying information. Second section presents a survey of temporal relational models and proposed extensions to the relational algebra [30], [32]

with a classification of the different approaches presented in the literature.

Third section gives an overview of proposed logical temporal data model. Most of the work is related to the semantics and the logical modeling of temporal data [17], [35].

2. Temporal Data and Database Types

2.1 Relational

A relational database [10], [19] is a set of tables called relations. Each relation contains one or more data categories in columns. Each row contains a unique instance of data for the categories defined by the columns. For example, a typical employee database would include a table that described an employee with columns for name, address, phone number, and so forth. Another table would describe a department: number, name, head, location, and so forth. A user of the database could obtain a view of the database that fits in the user's needs. For example, a human resource manager might need a report about employees who had availed their total leaves on a certain date. A financial services manager in the same company could, from the same tables, obtain a report on accounts that needed to be paid.

2.2 Object Relational

An object-relational database (ORD) [19], [27] is a database management system which has the basic features of the relational model with object oriented capabilities such as defining objects, classes and inheritance that are directly supported in database schemas and in the query language. In addition, it supports extension of the data model with custom data-types and methods. ORD permits the modeling of complex data (time varying data) and accommodates various data source formats and standards used by different databases. ORD technology aims to allow developers to raise the level of abstraction at which they view the problem domain.

2.3 Object Oriented

An object oriented database [19], [27] is a database model in which information is represented in the form of objects. Time has a complex semantics and object databases have the ability to capture and store it as a basic entity. The typing and inheritance mechanism of object oriented approach gives a better representation of different notion of time.

Object databases also offer some kind of support for representing and managing time varying data using an

object query language. Object database and management group (ODMG) made an effort for standardization with the introduction of Object Query Language, OQL.

2.4 Temporal Query Languages

Most of the commercial database management systems provide special data types for capturing date, time and timestamps. New version of Oracle also supports user's defined types and user's defined time values. Newer versions of structured query language provide a limited support for querying and managing temporal data. None of the products listed above is said to be a complete temporal database. Essential components for temporal database systems are as follows:

- Temporal data definition language
- Temporal data manipulation language
- Temporal query language [5]
- Temporal constraints (such as temporal referential integrity)

3. Discussion

Many data models introduced so far are designed to capture the semantics of temporal data keeping the traditional entity relationship model (ERM) approach. ERM is widely acceptable model for designing conceptual schema of a relational database system. Traditional ERM is capable for capturing the whole temporal aspects. Many extensions [17], [18] have been proposed to extend the ERM in order to capture time varying information in one way or the other. Unified modeling language (UML) [27] is also used as a tool to develop the logical and conceptual schema of the mini world. UML is normally used with the object relational or object oriented databases.

Another important point is how this new conceptual model will be incorporated into a relational database. One approach is to develop a temporal layer and that will be responsible for translating the temporal queries to traditional SQL statements [31]. The other approach is to design a complete temporal query language which will not only support all SQL statements but incorporate new operators based on temporal relational algebra [25]. There are many solutions for this problem and few of them are successfully implemented.

4. Important Concepts

Following are some of the important concepts related to the design of temporal model for handling data.

4.1 Transaction Time Vs. Valid Time

The transaction time [12], [20], [42] of an object is the time when the object is stored in the database the time at which it is present in the database. For example, if an item is being purchased on 12th June, 2005 recorded on 14th June, 2005 and deleted on 20th Dec, 2005. Another example may be of a company where, an employee receives a pay rise but it comes into effect from the date when payroll clerk enters this salary rise into the database. Transaction time has duration, beginning from insertion and ending at deletion, with multiple insertions and deletions being possible for the same fact.

The valid time [12], [20], [42] of a database object is the time when the object is effective or holds in reality the time when the event took place in reality. For example salary of an employee, whose name is Waleed on 2nd of October, 2001. Valid time is independent of the recording of the fact in a database.

The consideration of valid time is a big problem because some think “only one valid time for every tuple or field” and in some cases there exist many candidate valid times. The actual valid time is always a matter of semantics. The point of view in valid time and transaction time is quite clear and applicable and most of the researchers agree about the significance of these two times in context of the temporal databases [32].

4.2 First Normal Form (FNF) Vs. Non-First Normal Form (NFNF)

There are two approaches for handling temporal relations FNF (first normal form) and NFNF (non- first normal form) relations. Relation R is in FNF if and only if all underlying domains contain atomic values only. In contrast, NFNF relations contain multi-values in the time varying attribute. In other words the time varying attribute holds values with sets of intervals. The range of the time stamped attribute is the value domain. In most of the models FNF is used where we have tuple time stamping, whereas attribute time stamping relations can form NFNF relations.

4.3 Tuple Time Stamping Vs. Attribute Time Stamping

Temporal models are divided into two broad categories: tuple time stamping and attribute time stamping. In tuple time stamping [20], [42] each tuple is augmented by one or two attributes for the recording of timestamps. One additional attribute can be used to record either the time point at which the tuple becomes valid or the time at which the data is valid. Two additional attributes are used to record the start and stop time points of the corresponding time interval of validity of the

corresponding data. Tuple time stamping is usually applied in temporal relational data models supporting only FNF relations.

The alternative is attribute time stamping [20], [42] when the time is associated with every attribute which is time-varying. Note that it is not necessary for every attribute to be time-varying in an attribute time stamping approach. Consequently, a history is formed for each time-varying attribute within each tuple. As a result, the degree of the relation is reduced by one or two compared with the tuple time stamping equivalent relation, since timestamps are part of the attribute values. Attribute time stamping overcomes the disadvantage of data redundancy introduced when applying tuple time stamping. Attribute time stamping adds timestamps to each attribute value. Values in a tuple which are not affected by a modification do not have to be repeated. So, the history of values is stored separately for each attribute.

4.4 Time Points Vs. Time Intervals

Extensive study of time representation in temporal databases is conducted by [4], [22], [30], [40]. Three common approaches of time representation are a single time point, an interval and a set of time intervals. In [6], [40], time is expressed using single time points called as events. Most of the temporal models use time intervals to represent time, [23], [28]. A detailed study of point and interval types can be found [45].

There are many reasons that lead to the time point approach. A time point denotes either the start or the end of the lifespan of an object (relation, tuple or attribute). In order to store the whole history of the object, two different attributes need to be added in the relation, i.e. the start point and the stop point, so that the lifespan of that object can be shown. In contrast, time intervals contain the complete information about the lifespan of an object in a “compact style”. In the literature, a complete study of intervals has been given, where algebra has been used for their manipulation and the operations defined have proved to be closed. It is also observed that the definition of algebra becomes very difficult when time intervals [26] are used instead of time points [44][45].

5. Temporal Data Models

Since most of the work in the research area of temporal databases has been done with respect to the relational data model, numerous proposals can be found some of which are already implemented. Table 1 mentions some of the most important ones with respect to the work presented in this paper. Temporal data models can be categorized as tuple stamped or attribute time stamped, FNF or NFNF,

valid time or transaction time and some other important attributes.

Table 1. Summary of temporal extensions of the relational data model

<i>Temporal Model</i>	<i>NF</i>	<i>Time Stamping</i>	<i>Time</i>	<i>Features / Query Language</i>
Ariav's model [2].	INF	Tuple	VT & TT	Time stamps are based on time points
Ben-Zvi's Model [3]		Tuple	VT & TT	Time Relational model Extension of snapshot algebra
Clifford & Croker's Model [6], [7]	NINF	Attribute	VT	HDBM Inhomogeneous
Gadia's model [13][14]	NINF	Attribute	VT	Homogeneous HRQUEL
Gadia & Yeung's Model [16]		Attribute	VT & TT	Heterogeneous model
Jensen & Snodgrass's Model [21][22]	INF	Tuple	VT & TT	Bi-temporal Conceptual Model (BCDM)
Lorentzos's Model [24]	INF	Attribute	VT	Interval Relational Model (IRM) Extended Relational Model (XRM)
McKenzie's Model[28],[29]	NINF	Attribute	VT & TT	Extension of Snapshot algebra Historical algebra
Snodgrass's Model [37]	NINF	Tuple	VT & TT	TQUEL
Tansel's Model [20]	NINF	Attribute	VT	Non homogeneous
TSQL2 [38]	INF	Tuple	VT, TT & User Defined Time	Homogeneous model

6. Methodology

We adopt comparison methodology to propose new model of temporal database. This proposed model is based on comparison of same level but different behavior of different available databases such as RDBMS and OODBMS.

7. Introduction to the Proposed Model

By now the relational model is the most effective method for organizing huge amounts of data and still the widely

accepted technology amongst vendors and enterprises all around the world. Temporal data has its own semantics and organization [15] of such data requires some modifications in the relational model [4]. A lot of work has been done in the area of temporal models (fig. 1). Attributes in a relation can be time varying or non-time varying attributes. For e.g. salary attribute in an employee relation is a time varying relation, because salary changes with time and contrary gender attribute in the employee relation is a non time varying attribute.

7.1 Structure of Time:

The structure of the set time to be $\langle T, < \rangle$, where T is some countable set and less than, " $<$ " is a linear order on T, i.e. for any two time points t_1 and t_2 , either $t_1 = t_2$, $t_1 \leq t_2$, or $t_2 \leq t_1$. In our model we treat time as discrete and isomorphic to the natural numbers because any practical domain that we might define for time attributes in our proposed model would have at most an infinite countable set of names for time moments or time intervals.

Tuple time-stamping approach has been adopted to define a temporal model. The reason for this is the simplicity and to keep the INF assumption and the essence of the relational model.

The proposed logical temporal data model (fig. 1) comprises of three main constructs namely, the entity (temporal or no temporal), attributes (time-varying, non-time-varying and partial time varying attributes) and thirdly the relationship type amongst the entities are n-ary and categorized as temporal, non-temporal or semi temporal. Fully temporal attribute will be handled by a separate entity called temporal entity.

Following are some constraints to ensure the consistency of the conceptual schema:

- Entity is categorized as a temporal and non-temporal entity
- Temporal entity type must have a combinational primary key composed of time-varying and non-time-varying attributes. The activation start time is the part of the key.
- The n-ary relationship determines whether it is non-temporal, temporal or semi temporal relationship.

A LTRM Schema Modeling: Employee Database

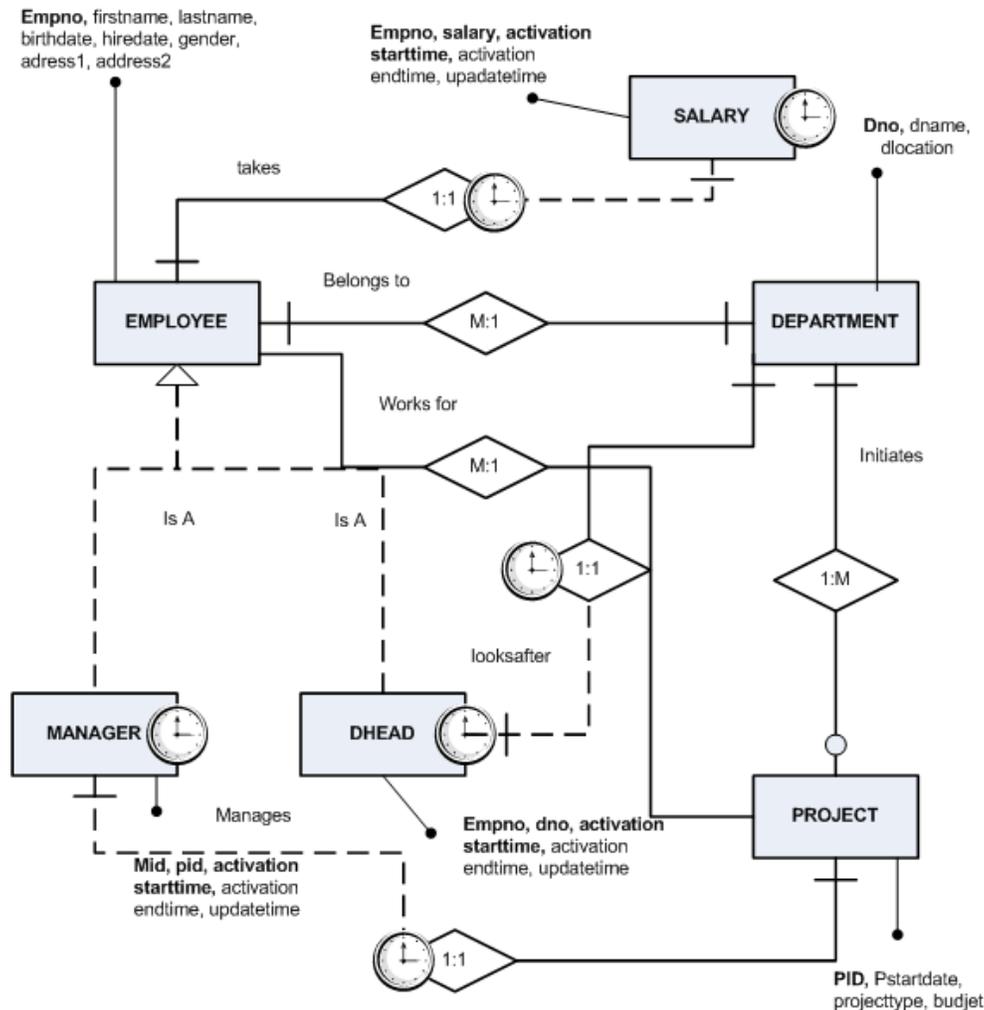


Figure 1: Temporal Conceptual Model of Employee Database

7.2 Representing Temporal Characteristics

- We have introduced a new term as activation_start and activation_end rather than valid time because validity itself is of many types and it creates confusion while defining an entity.
- It represents temporal facts both at time points as well as on intervals.
- Update time is introduced instead of transaction time. An update refers to change in data (tuple) of any sort (insert, delete or change).
- Activation time can be represented with different

time granularities [11] such as year, month, week, day, hour, minute and second and even beyond that. Conversion from one time granularity [46] to another is accomplished by the conversion functions.

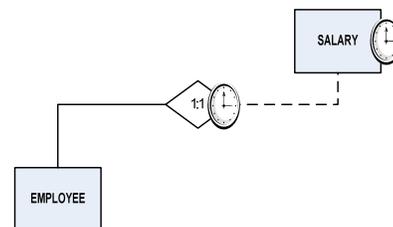


Figure 2(a): Employee and Salary Relationship

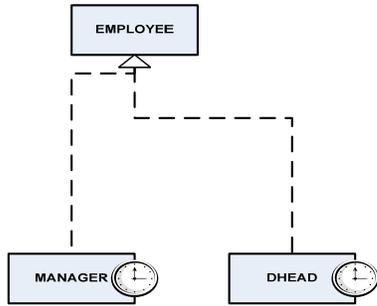


Figure 2(b): Generalized-Specialized Employee Relationship with Manager and Dhead

Salary is a temporal entity represented in fig. 2(a) that records the history of an employee salary. Activation_start shows that the value associated to it is the starting point. Employee represents the snapshot of employee’s entity and it is non-temporal, but it has a relationship with the entity salary which is temporal in nature. A dotted line from a salary is representing partial temporal relationship (see fig. 2(a). Manager and dhead are special type of employee represented in fig. 2(b) where manager and dhead are temporal entities.

7.3 Temporal Relation Schema

A temporal relation schema $TR = \langle A, K \rangle$ is an ordered pair consisting of finite set of $A = \{A1, A2, \dots, An, activation_start, activation_end, updatetime\}$. Each temporal relation has a current state, and the changed states (future and past). Relations and key attributes of the EMPLOYEE database are as follows.

Employee = (Empno, firstname, lastname, birthdate, hiredate, address1, address2, gender)

- Empno is a primary key and it is not a time varying attribute
- Address is a partial time varying attribute: handled by an application program

Manager = (empno, activation_start, activation_end, updatetime)

- Manager is a temporal relation
- Empno, startdate is a time-variant key
- Pid is a foreign key referencing project schema

Dhead = (empno, dno, activation_start, activation_end, updatetime)

- Dhead is a temporal relation
- Empno and startdate is a combinational primary key
- Dno is foreign key referencing department schema

Department = (Dno, dname, dlocation, dcontact)

- Dno is a primary key and it is not a time varying attribute

- No temporal attributes
- Project = (Pid, pstartdate, ptype, budget)
- Pid is a primary key and it is not a time varying attribute
- No temporal attributes
- Dno is a foreign key referencing department schema

Salary: (empno, salary, activation_start, activation_end, updatetime)

- Salary is a temporal schema
- Primary key is (Empno, salary and activation_start)
- FK (Employee_Empno)

Project: (Pid, pstartdate, projecttype, budget)

- Project is a non-temporal schema
- Primary key is Pid

Table 2a: EMPLOYEE

Empno	First name	Last name	Hiredate	Birthdate	Address1	Address2	Gender
101	ahmed	mumtaz	01-03-2000	02-05-1980	A60	Street 27	M
102	majid	bhati	14-03-2002	10-02-1976	A78	Street 22	M
103	aliya	rasheed	13-05-1995	11-08-1968	C12	Street 21	F
104	salim	khan	10-04-2006	12-11-1981	B21	Street 11	M
105	danish	khan	11-07-1997	01-02-1978	C76	Street 9	M
106	sajid	mahmood	02-04-1999	10-09-1977	D1	Street 9	M
....

Table 2b: DHEAD (Temporal relation)

Employee Empno	Department Deptno	Activation_start	Activation_end	Update time
103	10	13-05-1995	13-05-1998	14-05-1995
103	10	14-05-1998	21-07-2002	14-05-1998
103	20	22-07-2002	Current time	22-07-2002
....

Table 2c: SALARY (Temporal relation)

Empno	Salary	Activation_start	Activation_end	Update time
103	Rs. 900	13-05-1995	01-03-1998	12-05-1995
103	Rs. 1100	02-03-1998	11-12-2002	27-02-1998
103	Rs. 1300	12-12-2002	10-07-2005	11-12-2002
103	Rs. 1500	11-07-2005	Current time	09-07-2005
....
....

The non temporal facts of employees are stored in a relation employee (table 2a). Although the address attribute is also a time varying attribute, employee do not change address so often so it can be managed in the same relation, introducing a separate relation over here will increase overhead. Dhead relation (fig. 2b) stores the current and historic information of the head of the department.

Current time is a function which returns the current time. For e.g. employee 103 is getting Rs. 1500 since 11th of July, 2005. Minute is the default time granularity [39][46] for defining these relations but it can be changed by invoking conversion functions if required. The complete salary history of the employee 104 can easily be recovered from the salary relation (table 2c).

Update operations (insert, delete and change) result in insertion of a new tuple that contains the new values associated with a new timestamp. The new attribute value does not replace the previous (old) value of the attribute in case of an update such as deletion or insertion. Similarly, the deletion of tuples does not cause an actual deletion from database tables.

8. Conclusions

Temporal data model provides the mechanism to capture the time varying nature of entities and to design temporal query language to retrieve, manipulate and process the temporal data. The semantics of time is a big issue to deal with and it varies with the application and domain. Temporal relational database also holds time series data about entities and the events occurring in real time.

The proposed conceptual temporal relational model can be easily extended to incorporate time varying attributes. In our model each time varying attribute, identified in the mini world has a separate relation called as temporal relation. The concept of temporal relationship is introduced. We have presented some of the most important temporal models and discussed the semantics of our proposed temporal schema of the employee database.

The benefits of the proposed model are not restricted to reducing temporal attributes in the relation but it keeps the essence of the relational model and causes less overhead [36]. Due to the simplicity of the conceptual model, it will be easier to implement. Inclusion of too many temporal attributes in relations causes inconsistencies and redundancy in the database [36].

9. Future work

There are still some points which have not been addressed in this paper. The overhead reduced by this model in the case of insert, update and change operations should

formally be estimated and verified through experimental results. The fact that some events might occur in the modeled reality, but the reflection of these events to the database is delayed. We defer the discussion of these points in a future work.

Temporal aspect is significantly important in various fields and it can be very helpful in analyzing and understanding domain performance. We will evaluate this model in healthcare mobile system for the analysis of patients' movements to improve healthcare service. One of our co-author is working on healthcare care application domain in which context based knowledge management system model will be designed to improve service for hospital patients. In this model knowledge factor in changing patients' movement scenario will be captured through mobile technology. Our temporal model will be of vital use for their database implementation and integration.

References

- [1] Ahn, (1993) SQL+T: A Temporal Query Language. Proceedings of the Infrastructure for Temporal Databases, Arlington.
- [2] Ariav, G. (1986) A Temporally Oriented Data Model. ACM Transactions on Database Systems, 11(4):499-527.
- [3] Ben-Zvi, (1982) The Time Relational Model. PhD thesis, Computer Science Department, UCLA.
- [4] Böhlen, M. H. and Jensen, C.S. (1996) Seamless Integration of Time into SQL. TR R-96-2049, Aalborg University, Department of Computer, Fredrik Bajers Vej 7, DK-9220 Aalborg.
- [5] Chomicki, (1994) Temporal Query Languages: A Survey. In Gabbay, D. and Ohlbach, H., editors, Temporal Logic, First International Conference, Springer-Verlag, LNAI 827, pages 506-534.
- [6] Clifford, J. and Croker, A. (1987) The Historical Relational Data Model (HRDM) and Algebra Based on Lifespans. In Proceedings of the International Conference on Data Engineering. IEEE Computer Society Press, pages 528-537.
- [7] Clifford, J. and Croker, A. (1993) The Historical Relational Data Model (HRDM) Revisited. In A. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass, editors, Temporal Databases: Theory, Design, and Implementation, Benjamin/Cummings Publishing Company, pages 6-27.
- [8] Clifford, J. Croker, A. and Tuzhilin, A. (1993) On the Completeness of Query Languages for Grouped and Ungrouped Historical Data Models. In [Tansel et al., 1993], pages 496-533.
- [9] Clifford, J. Croker, A. and Tuzhilin, A. (1994) On Completeness of Historical Relational Query Languages. ACM Transactions on Database Systems, 19(1):64-116.
- [10] Codd, E. F. (1970) A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, 13(6), pages 377-387.
- [11] Combi, C. Franceschet, M. and Peron, A. (2004) Representing and Reasoning about Temporal Granularities. Journal of Logic and Computation, 14(1):51-77.

- [12] Date, C. J. Darwen, H. and Lorentzos, N. A. (2003) *Temporal Data and the Relational Model*. Morgan Kaufmann Publishers.
- [13] Gadia, S. K. (1988) *A Homogeneous Relational Model and Query Languages for Temporal Databases*. ACM Transactions on Database Systems, 13(4):418-448.
- [14] Gadia, S. K. (1992) *A Seamless Generic Extension of SQL for Querying Temporal Data*. Technical Report TR-92-02, Computer Science Department, Iowa State University.
- [15] Gadia, S. K. and Ben-Zvi's (1993) *Pioneering Work in Relational Temporal Databases*, chapter 8, pages 202-207.
- [16] Gadia, S. K. and Yeung, C. S. (1988) *A Generalized Model for a Relational Temporal Database*. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Chicago, IL, pages 251-259.
- [17] Gregersen, H. and Jensen, C. S. (2004) *Conceptual Modeling of Time-varying Information*. In Proceedings of International Conference on Computing, Communications and Control Technologies, pages 248–255.
- [18] Gregersen, H. (2005) *TimeERplus: A Temporal EER Model Supporting Schema Changes*. In BNCOD, volume 3567 of Lecture Notes in Computer Science, pages 41–59. Springer.
- [19] Hoffer, J. A. McFadden, F. R. and Prescott, M. B. (2005) *Modern Database Management*, 7th edition Upper Saddle River, NJ: Pearson/Prentice Hall.
- [20] Jensen, C. S. and Dyreson, C. E. [editors] (1998). *The Consensus Glossary of Temporal Database Concepts - February 1998 Version*. In *Temporal Databases: Research and Practice*, volume 1399 of Lecture Notes in Computer Science, pages 367–405. Springer-Verlag.
- [21] Jensen, C. S. and Snodgrass, R. T. (1992) *The TEMPIS Project. Proposal for a Data Model for the Temporal Structured Query Language*. TEMPIS Technical Report No. 37, Department of Computer Science, University of Arizona, Tucson.
- [22] Jensen, C. S. and Snodgrass, R. T. (1999) *Temporal Data Management*. IEEE Transactions on Knowledge and Data Engineering, 11(1), 36-44.
- [23] Lorentzos, N. A. (1988) *A Formal Extension of the Relational Model for the Representation and Manipulation of Generic Intervals*. Ph.D. Thesis, Department of Computer Science, Birkbeck College, University of London.
- [24] Lorentzos, N. A. (1993) *The Interval-extended Relational Model and its Application to Validtime Databases*. In A. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass, editors, *Temporal Databases: Theory, Design, and Implementation*, Benjamin/Cummings Publishing Company, chapter 3, pages 67-91.
- [25] Lorentzos, N. A. and R. G. Johnson, R. G. (1988) *TRA: A Model for a Temporal Relational Algebra*. In Proceedings of the Conference on Temporal Aspects in Information Systems, pages 99-112.
- [26] Lorentzos, N. A. and Mitsopoulos, Y. G. (1997) *SQL Extension for Interval Data*. IEEE TKDE, 9(3):480–499.
- [27] McBrien, P. J. and McCann, J. (2001) *Advanced Databases*, Imperial Course Notes.
- [28] McKenzie, J. E. (1988) *An Algebraic Language for Query and Update of Temporal Databases*. PhD thesis, Computer Science Department, Univ. of North Carolina at Chael Hill.
- [29] McKenzie, J. E. and R. T. Snodgrass, R. T. (1991) *Supporting Valid Time in an Historical Relational Algebra: Proofs and Extensions*. Technical Report TR_91_15, Department of Computer Science, University of Arizona, Tucson, AZ.
- [30] McKenzie, J. E. and Snodgrass, R. T. (1991) *Evaluation of Relational Algebras Incorporating the Time Dimension in Databases*. ACM Computing Surveys, 23(4), pages 501-543.
- [31] Melton, J. and Simon, A. R. (1993) *Understanding the new SQL: A Complete Guide*. Morgan Kaufmann.
- [32] Ozsoyoglu, G. and Snodgrass, R. T. (1995) *Temporal and real-time databases: a survey*. IEEE Trans Knowl Data Eng, 7(4): pages 513-532.
- [33] Sarda, L. (1990) *Algebra and Query Language for a Historical Data Model*. The Computer Journal, 33(1), pages 11-18.
- [34] Schek, H. J. and M. H. Scholl, M. H. (1991) *From Relations and Nested Relations to Object Models*. In M. S. Jackson and A. E. Robinson, editors, *Aspects of Database Systems: Proceedings of the 9th British National Conference on Databases*. Butterworth-Heinemann, pages 202-225.
- [35] Segev and A. Shoshani, A. (1987) *Logical Modelling of Temporal Data*. Proceedings of ACM SIGMOD Conference on Management of Data, San Francisco, pages 454-466.
- [36] Skyt, J. Jensen, C. S., and Mark, L. (2003) *A foundation for vacuuming Temporal Databases*. Data and knowledge engineering, 44(1): 1-29.
- [37] Snodgrass, R. T. (1987) *The Temporal Query Language TQuel*. ACM Transactions on Database Systems, 12(2):247-298.
- [38] Snodgrass, R. T. editor (1995). *The TSQL2 Temporal Query Language*. Kluwer Academic Publishers, 101 Philip Drive, Assinippi Park, Norwell, Massachusetts 02061, USA.
- [39] Spranger, S. (2006) *Calendars as Types – Data Modeling, Constraint Reasoning, and Type Checking with Calendars*. PhD Thesis. Herbert Utz Verlag, M`unchen.
- [40] Tansel, (1986) *Adding Time Dimension to Relational Model and Extending Relational Algebra*. Information Systems, 11(4), 343- 355.
- [41] Tansel, (1997) *Temporal Relational Data Model*. IEEE Transactions on Knowledge and Data Engineering, 9(3), 464-479.
- [42] Tansel, Clifford, J. Gadia, S. Jajodia, S. Segev, A. and Snodgrass, R. (1993) *Temporal Databases: Theory, Design, and Implementation*. Benjamin/Cummings Publishing Company.
- [43] Tansel and Garnett, L. (1989) *Nested Historical Relations*. Proceedings of the ACM SIGMOD International Conference on Management of Data, Portland, Oregon, pages 284-293.
- [44] Terenziani, P. and Snodgrass, R. (2004) *Reconciling point-based and interval based semantics in temporal databases: A proper treatment of the telic/atelic distinction*. IEEE Transactions on knowledge and Data Engineering, 16(4).
- [45] Toman, D. (1996) *Point vs. Interval-based Query Languages for Temporal Databases*. Proceedings of the 15th ACM SIGACTSIGMOD- SIGART Symposium on Principles of Database Systems (PODS), Montreal, Canada, pages 58-67.

- [46] X. S. Wang, G. Mason, Logical Design for Temporal Databases with Multiple Granularities University, Fairfax, ACM 0362-5915/97/0600-0115 ACM Transactions on Database Systems, Vol. 22, No. 2, (1997).

Mahmood, Nadeem

Nadeem Mahmood is a PhD research scholar and an assistant professor in Department of Computer Science, (UBIT), University of Karachi. His area of research is the application of temporal logic in relational database systems, temporal databases and temporal data modeling. He has masters in Computer Science from University of Karachi. He has several research publications in journals and presented papers in national and international conferences. Currently a member of ACM and member of Healthcare Care Information System group at University of Karachi.

Burney, Aqil

Dr. S. M. Ail Burney received his B.Sc, first class first M.Sc. M.Phil. from Karachi University in 1970, 1972 and 1983 respectively. He received Ph.D. degree in Mathematics from Strathclyde University, Glasgow with specialization in estimation, modeling and simulation of multivariate Time series models using algorithmic approach with software development. He is currently professor at (UBIT) University of Karachi and approved supervisor in Computer Science and Statistics by the High Education Commission, Government of Pakistan. His research interest includes AI, soft computing, neural networks, fuzzy logic, data mining, statistics, simulation and stochastic modeling of mobile communication system and networks and network security. He is author of three books, various technical reports and supervised more than 100 software/Information technology projects of Masters level degree programs and project director of various projects funded by Government of Pakistan. He is member of IEEE (USA), ACM (USA) and fellow of Royal Statistical Society United Kingdom and also a member of Islamic society of Statistical Sciences. Dr. Burney has received appreciations and awards for his research and as educationist including NCR-2002 award for Best National Educationist.

Ahsan, Kamran

Kamran Ahsan is a PhD researcher and lecturer in FCET (Faculty of Computing, Engineering and Technology) and, web researcher in Centre for Ageing and Mental Health, Staffordshire University, UK since 2005. He is researcher with Health Strategic Authority, UK. He has an MSc in Mobile Computer Systems from Staffordshire University and MCS in Computer Science from University of Karachi. He has several publications and member of various Journals/Conferences reviewing committee. His research interests are mobile technology applications in healthcare including knowledge management.