

Temporal Databases

Sumit Chhabra

*Dept of computer science and Applications
 Khalsa College for Women
 Amritsar*

Submitted: 01-08-2021

Revised: 14-08-2021

Accepted: 17-08-2021

INTRODUCTION

A Temporal Database is a database with built-in support for handling time sensitive data. Usually, databases store information only about current state, and not about past states. For example, in an employee database if the address or salary of a particular person changes, the database gets updated and the old value is no longer there. However for many applications, it is important to maintain the past or historical values and the time at which the data was updated. That is, the knowledge of evolution is required. That is where temporal databases are useful. It stores information about the past, present and future. Any data that is time dependent is called the temporal data and these are stored in temporal databases. Temporal Databases store information about states of the real world across time. Temporal Database is a database with built-in support for handling data involving time.

Timelines, points, duration and intervals

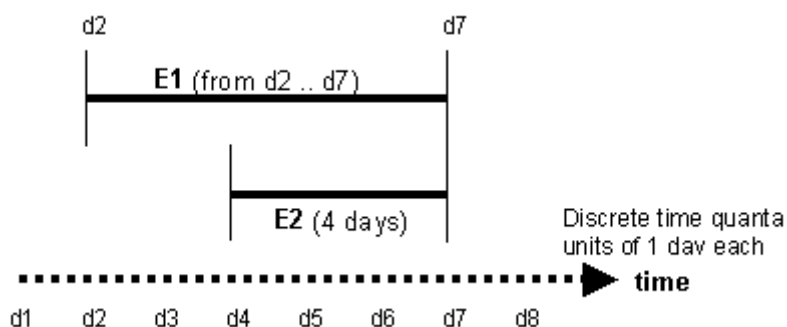
When attempting to represent and reason about time, four important concepts to consider are:

- **Points:** Formally a point in time has no duration; it simply refers to a particular position in the timeline under discussion. We

can talk of the point in time at which some event begins or ends.

- **Duration:** Duration refers to a number of time quanta; for example, a week, two months, three years and 10 seconds are all durations.
- **Interval:** An interval has a start time point and an end time point. Using more formal notation, we can refer to an interval $I(s, e)$ with start point s and end point e , and for which all points referring to time from s to e (inclusive) make up the interval.
- **Timeline:** Conceptually we can often imagine time as moving along a line in one direction. When graphically representing time, it is usual to draw a line (often with an arrow to show time direction), where events shown towards the end of the timeline have occurred later than those shown towards the beginning of the line. Often a graphical timeline is drawn like an axis on a graph (by convention, the X-axis represents time) and the granularity of the time units is marked (and perhaps labelled) along the X-axis.

The following diagram illustrates these four concepts:



In above figure, there are two events, E1 and E2. Event E1 starts at the time point d2 and ends at time point d7, therefore event E1 is an example of an interval. All that is written for event

E2 is that it lasts four days, so event E2 is a duration (although we might attempt to infer from the X-axis that E2 appears to start at d4 and end at d7, but perhaps E2 is four days measured back

from d7). The X-axis is a labelled arrow time, and represents a timeline. There are units labelled on the X-axis from d1 to d8, and a note indicates that these units are time quanta representing one day each.

Valid time and Transaction time

There are mainly two different notions of time which are relevant for temporal databases. One is called the **valid time**, the other one is the **transaction time**. Valid time denotes the time period during which a fact is true with respect to the real world. Transaction time is the time period during which a fact is stored in the database. Note that these two time periods do not have to be the same for a single fact. Imagine that we come up with a temporal database storing data about the 18th century. The valid time of these facts is somewhere between 1700 and 1799, whereas the transaction time starts when we insert the facts into the database, for example, January 21, 1998.

another example, consider Mr. X who registers for his health insurance on 4th June, 1995. However, the health insurance company could not record this information in their computer system because of technical challenges. On 8th July, 1995, the company is able to record Mr. X's information in their database. In this case, 4th June, 1995 is the valid time start event of his insurance and 8th July, 1995 becomes the transaction time.

Concept of relational temporal databases

A DBMS stores the data in a well-defined format. A relational DBMS stores data in tables (also called relations). Thus, a relational database actually contains a set of tables. Each table contains rows (tuples) and columns (attributes). A row contains data about a specific entity, for example, an employee. Each column specifies a certain property of these entities, for example, the employee's name, salary etc. The following table stores data about employees:

EmpID	Name	Department	Salary
10	John	Sales	12000
12	George	Research	10500
13	Ringo	Sales	15500

In a temporal relational database, tuples are timestamped, so that the successive states of an entity (past, current and future) can be stored. So, after tuple timestamping, the employee table could result as follows:

EmpID	Name	Department	Salary	Valid TimeStart	Valid TimeEnd
10	John	Research	11000	1985	1990
10	John	Sales	11000	1990	1993
10	John	Sales	12000	1993	INF
11	Paul	Research	10000	1988	1995
12	George	Research	10500	1991	INF
13	Ringo	Sales	15500	1988	INF

The above table stores the history of the employees with respect to the real world. The attributes **ValidTimeStart** and **ValidTimeEnd** actually represent a time interval which is closed at its lower and open at its upper bound. Thus, we see that during the time period (1985 - 1990), employee John was working in the research department, having a salary of 11000. Then he changed to the sales department, still earning 11000. In 1993, he got a salary raise to 12000. The upper bound INF denotes that the tuple is valid until further notice. Note that it is now possible to

store information about past states. We see that Paul was employed from 1988 until 1995. In the non-temporal table, this information would have been deleted when Paul left the company.

Different Forms of Temporal Databases

The two different notions of time - valid time and transaction time - allow the distinction of different forms of temporal databases. A **historical database** stores data with respect to valid time, a **rollback database** stores data with respect to transaction time. A **bitemporal database** stores

data with respect to both valid time and transaction time.

Commercial DBMS are said to store only a single state of the real world, usually the most recent state. Such databases usually are called **snapshot databases**. On the other hand, a **bitemporal DBMS** such as TimeDB stores the history of data with respect to both valid time and transaction time. A table in the bitemporal relational DBMS TimeDB may either be a snapshot table (storing only current data), a valid-time table (storing when the data is valid wrt. the real world), a transaction-time table (storing when the data was recorded in the database) or a bitemporal table (storing both valid time and transaction time). An extended version of SQL allows to specify which kind of table is needed when the table is created. Existing tables may also be altered (schema versioning). Additionally, it supports **temporal queries, temporal modification statements** and **temporal constraints**.

Examples of Temporal Databases

Some examples of temporal databases are:

- **Healthcare Systems:** Doctors need the patients' health history for proper diagnosis. Information like the time a vaccination was given or the exact time when fever goes high etc.
- **Insurance Systems:** Information about claims, accident history, time when policies are in effect needs to be maintained.
- **Reservation Systems:** In Reservation systems such as hotel, airline, car rental, train, and so on, information on the dates and times when reservations are in effect are required.
- **Scientific Databases:** In Scientific databases data collected from experiments includes the time when each data is measured; and so on.
- **Company Database:** In the company database, a user may wish to keep salary, job and project histories on each employee.
- **University Database:** In the university database, time may be included in the semester and year of each section of a course, the grade history of a student, and the information on research grants.

Products Using Temporal Databases

The popular products that use temporal databases include:

- Oracle.
- Microsoft SQL Server.
- IBM DB2.

REFERENCES:

- [1]. https://www.cs.uct.ac.za/mit_notes/database/htmls/chp18.html.
- [2]. https://en.wikipedia.org/wiki/Temporal_database
- [3]. <http://www.timeconsult.com/TemporalData/TemporalDB.html>
- [4]. <https://www.slideshare.net/DabalMahara1/temporal-databases-97200226>
- [5]. http://www.brainkart.com/article/Temporal-Database-Concepts_11602
- [6]. Ekta, Ankit, Chinu, " Temporal Databases – A Review", IJRAR- International Journal of Research and Analytical Reviews, Volume-5, Issue 2, April-June 2018.
- [7]. P. Gupta, R. Rishi, R. Biswas, "A Comparative Analysis of Temporal Databases", International Journal of Advanced Computational Engineering and Networking, Volume-1, Issue- 8, Oct-2013.
- [8]. R.Mensah, V.Amankona, "SQL and Temporal Database Research: Unified Review and Future Directions", International Research Journal of Engineering and Technology, Volume-04 Issue: 09, Sep - 2017