I

Semantics of Temporal Data

This part presents works that offer insight into central semantic aspects particular to temporal data, thus contributing to the understanding of temporal data and thereby informing the research in the other aspects of temporal data management covered in subsequent parts, including data model and query language development, and database design, conceptual as well as logical. The part begins with a chapter that surveys the topics covered by the next two chapters as well as the fifth chapter of this part.

Facts stored in a database may be associated with time, most prominently with one or more dimensions of valid time and transaction time. One common case is that of bitemporal facts, which are associated with timestamps from exactly one valid-time and one transaction-time dimension. These two timestamps of a fact may be related in various ways, yielding *temporal specialization*. Multiple transaction times arise in application systems where facts flow from database to database, in the process accumulating associated transaction times. By retaining the transaction times, *temporal generalization* occurs where the earlier relations can be effectively queried by referencing only any later relation. Chapter 3 offers a systematic study of temporal specialization and generalization. The chapter’s framework offers a basis for more precisely characterizing, comparing, and thus understanding temporal databases and data models; and its system of additional semantics of provides a basis for improving query processing.

An ubiquitous and, as it turns out, quite intriguing notion in temporal database management is that of the ever-increasing current time. There is frequently a need to record that facts are valid or current in the database until the current time. However, SQL only permits the storage of fixed, or ground, time values in the database. Chapter 4 proposes a framework for defining the semantics of the variable databases that result when allowing the use of the variable *now* as a timestamp value in databases. This framework is extended to cover also the use of so-called *now-relative* and *now-relative indeterminate* timestamp values, and it provides a foundation for the querying of variable databases via existing query languages.

Chapter 5 proceeds to study the *semantics of modifications* involving variable *now* and of tuples timestamped with intervals containing *now*, thus addressing an aspect not covered by the framework of the previous chapter. This chapter defines semantics of modifications—including insertions, deletions, and updates—of databases without and then with *now*. The main challenges stem from modifications that extend into the future and from tuples with end times in the future. The semantics of modifications on *now*-extended databases are based on those of non-*now*-extended databases, and they necessitate the introduction of two new timestamp values. The chapter also provides an approximate semantics that may be accommodated by the existing first normal form representation formats for temporal data.

Over the past two decades, several dozen temporal relational data models have been proposed. This proliferation of models may have been caused in part by each individual model attempting to simultaneously satisfy irresolvable requirements—a single model that is ideal for data display, data modeling, and data storage is unattainable. Chapter 6 introduces a new and very simple bitemporal data model designed with the single purpose of capturing the essence of time-referenced data, the Bitemporal Conceptual Data Model (BCDM). A framework of well-behaved mappings between this and existing bitemporal models, is then provided that illustrates how it is possible to use different models for display, data modeling, and storage purposes in a temporal database system. The mappings move the distinction between the investigated data models from a semantic basis to a display-related or a physical, performance-relevant basis.

The association of timestamps with various data items, e.g., tuples or attribute values, representing facts, is fundamental to temporal data management. Most existing temporal data models employ a time interval data type for associating time values with facts. Different meanings are possible for these timestamp values, and the specific choice of meaning is important—it has a pervasive effect on most aspects of a data model, including database design, a variety of query language properties, and query processing techniques, e.g., the availability of query optimization opportunities.

Chapter 7 defines and studies two possible meanings that may be given to timestamps. Specifically, it defines the notions of *point-based and interval-based temporal data models*, thus providing a formal basis for characterizing temporal data models and obtaining new insights into the properties of their query languages. Most existing temporal data models, including the BCDM, adopt a point-based semantics, in which time intervals are merely convenient formats for recording sets of time points. While point-based models are insensitive to interval start and end points, interval-based models give significance to the actual intervals used in the timestamps and encode more semantics in the timestamps.

When the transaction-time aspect of data is captured, the usual update-in-place policy is effectively replaced by an append-only policy, where (logical) deletions are implemented as insertions at the physical level, leading to ever-growing databases. However, the capture of transaction time does not efface the need for deletions that physically remove data, termed vacuuming.

Chapter 8 delves into the consequences of this interesting aspect of the semantics of transaction time. Specifically, the chapter presents a semantic framework for the *vacuuming of transaction-time databases*. Although necessary, vacuuming facilities may be used for manipulating the database's previously perfect and reliable recollection of the past, thus rendering query results potentially "unreliable." The chapter establishes a foundation for the correct and cooperative processing of queries and updates against vacuumed databases.