II

The TSQL2 Query Language

Part I presented contributions that aimed to enhance the understanding of temporal data and data models and query languages. Those contributions were in large part generic rather than being specific to a particular data model or query language. This part offers a concrete case study—applying the insights from Part I, in addition to the insights accumulated in earlier models and languages and related contributions by other researchers—in the development of primarily the core aspects of a concrete temporal data model and query language.

A temporal extension to the SQL-92 standard, TSQL2 results from an ambitious initiative that aimed to consolidate more than fifteen years of active research in temporal data models and query languages. At the initiative's start in 1993, a substantial number of rather diverse temporally enhanced data models and query languages had been proposed over the preceding fifteen years by the database research community. It was argued that this state of affairs had an adverse affect on further progress in the research community and in the industrial arena. Based on observations like this, the initiative's objective was to propose a single temporal data model and query language that built maximally on and consolidated previous models and query languages, had wide backing among temporal database researchers, and served as a recommendation to the industry for a temporal data model and language on which to base a temporal database product.

Following a general invitation to the database research community, the TSQL2 committee was formed. The committee consisted of Ilsoo Ahn, Gad Ariav, Don Batory, James Clifford, Curtis E. Dyreson, Ramez Elmasri, Fabio Grandi, Wolfgang Käfer, Nick Kline, Krishna Kulkarni, T. Y. Cliff Leung, Nikos Lorentzos, John F. Roddick, Arie Segev, Richard T. Snodgrass, Michael D. Soo, Surynarayana M. Sripada, and the author. The originater of the initiative, Richard T. Snodgrass, chaired the committee.

A preliminary design of TSQL2 was completed in early 1994, and its Language Specification was publicized in the March, 1994 version of the ACM SIGMOD Record. This version was subsequently revised, and the definitive version of the TSQL2 Language Specification was released in September, 1994, and a tutorial of the language appeared in the ACM SIGMOD Record that month.

The language specification proper was accompanied by a collection of commentaries that offered insight into the design of TSQL2, including the background of and objectives behind the constituent constructs and their relationships to the many existing proposals, the design alternatives considered and the rationales underlying important design decisions, and the properties of the data model and language. These commentaries and the language specification appeared as a book in September, 1995. The chapters in this part derive primarily from that book. TSQL2 is the temporal query language with the richest set of features to date, and is also the most comprehensively documented temporal query language.

The first two chapters serve to introduce TSQL2. Chapter 9 uses an application from advertising, namely media plan management, for introducing TSQL2. The chapter covers temporal database schema definition, including temporal granularities, and covers a variety of fundamental and advanced query language constructs. Chapter 10 offers a quite different example of TSQL2 in action. This chapter illustrates how different classes of natural-language queries on temporal data may be expressed in TSQL2. The different classes of queries originally served as a means of evaluating the design of TSQL2 and of comparing TSQL2 to other temporal query languages. The next nine chapters cover the design of the soul of the query language and its underlying data model.

A typical temporal database records multiple versions of the properties of objects as these evolve over time. Therefore, it is attractive to have available means of cleanly identifying the versions that pertain to an object. The surrogate data type introduced in Chapter 11 provides such a means. Surrogates are ideal for encoding (non-value based) identity: Unlike for any other data type, surrogate values cannot be seen, but can only be compared for equality.

Chapter 12 proceeds to introduce the TSQL2 data model that the surrogate data type is embedded into. The data model includes tables with (or without) support for the temporal aspects of transaction time and valid time; tuples are timestamped with temporal elements, which are arbitrary subsets of the cross product space of the temporal aspects supported. Note that this is also the BCDM data model, proposed in Chapter 6. Chapter 13 follows up by giving the TSQL2 syntax necessary for specifying temporal tables.

With schema definition in place, attention shifts to querying. TSQL2 extends the `SELECT`, `FROM`, and `WHERE` clauses of SQL's declarative `SELECT` statement, in Chapters 14 and 15. The `FROM` clause specifies the tables to be used in a query statement. Exploiting the closure properties of temporal elements, TSQL2 provides convenient syntax for the grouping of tables on any subset of their non-temporal columns, thus automatically coalescing timestamp values. "Opposite" facilities are also provided that make it possible to partition a table, so that value-equivalent tuples are generated for each maximal time period in a tuple's timestamp. The first facility is well suited for queries regarding durations, while the latter targets queries embodying consecutiveness.

Valid-time selection and projection concerns the `WHERE` and `SELECT` clauses. Following a survey of these aspects in existing query languages and a statement of the design goals, Chapter 15 in turn presents the constructs available for valid time selection and projection. Specifically, constructs are given for referencing the timestamps of tuples, for the extraction of components of timestamps, for the construction of timestamp values, and for the comparison of timestamp values. A new clause, `VALID INTERSECT`, is included for the specification of the timestamps of result tuples, and a new keyword, `SNAPSHOT`, is offered for discarding the timestamps of result tuples.

Chapter 16 proceeds to provide facilities for database modification, covering insertion, deletion, and update with a focus on the specification of the timestamp values involved in these operations.

Cursors, covered in Chapter 17, provide the means for TSQL2-based database access from applications. The novel aspect is that the timestamps of tuples are temporal elements that consist of an arbitrary number of periods. This leads to a nested cursor model.

Chapter 18 introduces the use of the variable *now*, denoting the current time, as an period-end value in timestamps in tuples. In addition to *now*, also *now* plus or minus a duration of time, e.g., *now* $+$ 3 *days*, is allowed. Chapter 4 provides the background for this aspect of TSQL2.

When a table supports transaction time, deletion statements no longer actually delete data at the physical level, prompting a need for new physical-deletion facilities, termed vacuuming. Based on the foundation laid out in Chapter 8, Chapter 19 enhances TSQL2 with restricted vacuuming facilities, in the form of cut-off points.

Aspects related to the data types for time (e.g., the model for time and calendar support) are covered by other authors, as is tables for recording events ("event" tables), transaction-time support, and support for temporal indeterminacy and granularity. Finally, aggregates and schema versioning are also covered by others.

The penultimate two chapters concern implementation aspects and were originally included with the purpose of arguing for the feasibility of TSQL2. Chapter 20 discusses the extensions necessary to a conventional database management system architecture in order to support TSQL2. In a conventional architecture, an SQL-level statement is translated to an internal algebraic form, which is then optimized and evaluated. Chapter 21 defines a temporally extended algebra to go with TSQL2.

The last chapter contains the language specification for TSQL2. This chapter follows the format of the SQL-92 language standard, which it extends.