Point-based Temporal Extension of SQL

David TOMAN
Department of Computer Science, University of Toronto
Toronto, Ontario M5S 1A4, Canada. Supported by a NATO/NSERC PDF fellowship 1996-98

Proposed by: Nicole BIDOIT
Supervised by: Emmanuel WALLER
Studied by: Micheline ELIAS, Adeel ANJUM
Laboratoire de Recherche en Informatique
Bat. 490 Université Paris sud 11
Orsay, France

Domain: The Domain of this article is databases which is a structured collection of related records that is stored in a computer system. The structure is achieved by organizing the data according to a database model, as relational model which groups data using common attributes found in the data set. The database model tends to determine the query languages that are available to access the database, as the structured query language SQL which represents a computer database language designed for the retrieval and management of data in relational database. The most common operation in SQL is the query, which is performed with the declarative SELECT keyword. SELECT retrieves data from a specified table or multiple related tables in a database.

Problem: A large amount of database research is directed towards to overcome the limitations of the classical relational model. Our research is about adding temporal capabilities. Temporal database has temporal aspects usually includes valid-time (the time period during which a fact is true with respect to the real world) and transaction time (the time period during which a fact is stored in the database), these two attributes go together to form bitemporal data. Timestamp is a sequence of characters, denoting the time at which a certain event occurred. Time stamping tuples (rows) leads to enormous space requirements because the tuples are repeated for each time instant in which the tuple is true and fulfills a fact. And there was a solution by using compact encodings of sets of time instants which presents periods of validity associated with a tuple which are encoded using intervals and bitemporal elements.

Sub problem: interval-based encoding of timestamp leads to tension between the syntax and the intended semantics of the language, where time references are realized by intervals, even that data model and semantics are point-based; intervals are used as compact description of large sets of instants.

Approach: In order to nullify the limitations in recent proposals relating to temporal extensions of SQL, the point-based references to time approach is taken into account, to facilitate both existing and new database management systems to enhance the support for temporal data. This paper defines a new SQL/TP which extends the syntax and semantics of SQL/92, where SQL/92 is based on the relational algebra and treats intervals as atomic values without any special temporal semantics. Therefore, SQL-92 is an interval-based data model. It also follows that SQL-92 is not pointbased.

Interest and motivation: we define a language with a new single type which presents a linearly ordered universe of individual time instants, with meaningful approach to duplicate semantics and aggregation that is independent of the encoding, so there is no need for repeating rows for presenting a period of time.

Contributions:

- Data model for temporal database:
- Time domain: discrete countable infinite linearly ordered set without end points. Individual elements represent the actual time instants while the linear order represents the progression of time. The time granularity is implementation-dependent.
- Abstract temporal database: is a set of tables where a particular tuple is related to a large or infinite set of time instants for expressing a period, instead we can use compact encoding of sets of time instants which we chose concrete temporal DB (interval based encoding).
- Concrete temporal database: the internals are represented by *tmin* and *tmax* which specifies endpoints of intervals.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Point-based Temporal Extension of SQL

David TOMAN
Department of Computer Science, University of Toronto
Toronto, Ontario M5S 1A4, Canada. Supported by a NATO/NSERC PDF fellowship 1996-98

Proposed by: Nicole BIDOIT
Supervised by: Emmanuel WALLER
Studied by: Micheline ELIAS, Adeel ANJUM
Laboratoire de Recherche en Informatique
Bat. 490 Université Paris sud 11
Orsay, France

- The language SQL/TP

- Data definition language: is like standard SQL/92, but the difference is that we defined a new data type '*time'* for declaring temporal attributes and has modifiers to determine how the time instances are stored in tables. There are two ways to specify time:
    - Using points: the time is stored as atomic value for representing a single atomic event that happened in a specific time.
    - Using [bounded|unbounded] intervals: represents continuous time instants as duration of a specific event. Bounded and unbounded keys specify ∞ and -∞ as an endpoint for the intervals.
- Query language: uses two basic syntactic constructs:
    - Select block: same to standard SQL
    - Set operations: same to standard SQL
    - We will use a special constant pseudo-relation: true (t: time) for the temporal domain; by that we can have the complementation over the temporal domain.
- Semantics: SQL/TP is SQL/92 extended with additional data type 'time'; so we can use the familiar SQL-like semantics, and not affecting the syntax and semantics of queries.

- Query Compilation Technique**:**
    - To verify that the queries built in SQL/92 are efficient
    - Definition of a translation procedure that allows compiling SQL/TP statements to standard SQL/92 statements and translation utilizes the quantifier elimination procedure for linear order to replace reference to individual time instants in the queries with references to interval endpoints.

  SQL/TP can be:
    - Implemented on top of an interval-based representation of temporal database.
    - Built to an existing RDBMS, and provide temporal capabilities without changing or modifying the underlying database system.
    - Express all representation independent SQL/Temporal queries.

Work related and distinctive points: The abstract view of point based approach to temporal extensions of SQL has advantages over the common approaches that use interval-based attributes.

Perspectives: With a new query language SQL/TP we expect several other future perspectives mentioned below:

- Use of complex temporal domains
- Optimization technique for SQL/TP
- Area of updates is always cumbersome for any query language. Management of updates in SQL/TP
- Introduction of specialized indices for SQL/TP

PDF created with pdfFactory Pro trial version www.pdffactory.com