11 Implementation of Temporal Database Management Systems

Hai Liu\textsuperscript{1,2}, Yong Tang\textsuperscript{1,2+}, Xiaoping Ye\textsuperscript{1,2}, and Huan Guo\textsuperscript{2}

\textsuperscript{1} Computer School, South China Normal University, Guangzhou 510631, P.R. China
\textsuperscript{2} Department of Computer Science, Sun Yat-sen University, Guangzhou 512075, P.R. China

Abstract Dealing with temporal information has become a key technique in many database systems and information systems. We are in great need of a temporal database management system (TDBMS) to handle temporal data efficiently. This chapter gives a brief introduction to TDBMS and introduces two temporal data processing prototype systems—TimeDB and TempDB. The design principles, installation and user interfaces are presented. Several examples are given to demonstrate the basic functions of TDBMS.

Keywords TDBMS, TimeDB, TempDB

11.1 Introduction

There are already abundant temporal database theories including several temporal data models and query languages, which cover various characteristics of temporal data and the basic problems under consideration. Among these theories, there are several representative temporal query languages, such as TempSQL (Yau and Chat 1991), TQuel (Snodgrass 1987), TSQL2 (Snodgrass 1995), and ATSQL2 (Böhlen et al. 1995). ATSQL2 is a temporal extension for SQL and a representative language that integrates the spirits of TSQL2, Chronolog (Böhlen 1994; Böhlen and Marti 1994), and Bitemporal ChronoSQL (Pulfer 1995). However, the implementation techniques of temporal databases are relatively lagging behind the theoretical achievements. Prior to the design and implementation of a temporal data processing prototype system, we should first know what a temporal data management system (TDBMS) is and what functions it should include. In general, a TDBMS is a software system that helps users to store, access and maintain temporal data in a uniform way, that is, to manage time-varying data. TDBMS
should provide:

1. A temporal data definition language
2. A temporal data manipulation language
3. A temporal query language
4. Temporal constraints (such as temporal referential integrity)

TimeDB (Time Consult; Steiner 1998) and TempDB (TempDB) are two of the famous temporal data processing prototype systems. Both of them are based on ATSQL2. The organization of this chapter is as follows: Section 11.2 introduces the functions of TimeDB, Section 11.3 covers the operations of TempDB.

11.2 TimeDB

TimeDB, which is developed by Andreas Steiner, is considered as a successful temporal database prototype system. TimeDB is a temporal relational database system based on the query language ATSQL2 and it runs as a front-end of the commercial relational database management system (e.g. Oracle). As a relatively mature TDBMS, TimeDB realizes the basic temporal management functions: temporal query, temporal update, temporal view and temporal integrity constraints. In TimeDB, ATSQL2 statements (queries, updates and assertions) are compiled into (sequences of) SQL-92 statements that are executed by the background database. This approach guarantees high portability between different platforms and different DBMS back-ends. TimeDB provides original temporal statements. TimeDB excels due to its seamless integration of time into database, and its upward compatibility and temporal upward compatibility, whilst it supports query language, data manipulation language, data definition language and assertions/constraints. Version 1 of TimeDB was written using SICStus Prolog and was ported to run with SWI Prolog. Version 2 of TimeDB is based on Java, uses JDBC, has a set of APIs and offers more functionality. Full version of TimeDB 2.0 is a commercial version, but TimeDB 2.0 Beta 4 is available for free download for research and study purposes. In the following sections, we will introduce TimeDB based on TimeDB 2.0 Beta 4.

11.2.1 Installation

1. Software requirements

Before running TimeDB, the following software is needed:

- Java Runtime Environment 1.1 (or newer)
- A DBMS, e.g., Oracle (Version 8), Sybase (Version 11.5)
- A JDBC driver for the DBMS

You will need the login and password for the database you will use, the JDBC