

А. Н. Базаркин

Исследование и разработка темпоральной модели данных в рамках МИС Интерин PROMIS

Аннотация. В статье исследуются основные методы построения темпоральных моделей данных в реляционных СУБД. Формулируются основные понятия теории построения темпоральных баз данных, раскрывается понятие темпоральной модели данных. В работе приводится несколько критериев классификации методов построения темпоральных моделей, а также обобщается опыт реализации одной из темпоральных моделей в рамках интегрированной подсистемы МИС Интерин PROMIS.

1. Введение

Глобальный процесс информатизации коснулся практически всех сфер деятельности человека и стал неотъемлемым условием четкости и успешности функционирования отрасли в целом. Эффективное управление учреждением стало напрямую зависеть от функциональных возможностей информационных систем (ИС). В последнее время особое распространение получили идеи построения так называемого единого информационного пространства (ЕИП), связывающего отдельные информационные системы и учреждения. Ключевым понятием в этих системах является понятие информации, и, несмотря на то, что до сих пор не сформулировано универсального определения этому понятию, именно обеспечение хранения и доступа к информации, а также ее достоверность определяют основные качества и функциональные возможности ИС.

С увеличением потребностей в качестве и достоверности информации, а также с уменьшением стоимости дискового пространства особый интерес приобретают темпоральные базы данных. Актуальной областью исследований становятся формальные модели данных, ориентированные на хранение темпоральных данных.

В отличие от традиционных моделей данных, обеспечивающих хранение лишь мгновенного снимка объектов предметной области, темпоральные модели данных позволяют хранить информацию об эволюции объектов: для любого объекта, который был создан в момент времени T_1 и закончил свое существование в момент времени T_2 ,

в базе данных (БД) будут сохранены все его состояния на временном интервале $[T_1, T_2]$ [1].

Под «темпоральностью» объекта следует понимать явную или неявную связь объекта с определенными датами или промежутками времени. В самом широком смысле, темпоральные данные — это данные, которые могут изменяться с течением времени.

2. Темпоральность в ИС

Несмотря на то, что многие приложения успешно функционируют на основе традиционных систем управления базами данных (СУБД), существующих возможностей явно недостаточно для информационных систем, динамика изменения информации в которых является одним из ключевых моментов. Существует множество прикладных областей, где требуется не только восстановление более ранних состояний базы данных на определенную дату в прошлом, но и создание состояний базы данных на момент времени в будущем.

Так, например, процесс принятия решений, касающихся развития компании или медицинского учреждения, должен опираться на достоверные и актуальные данные о деятельности организации. Существенное влияние на качество принимаемых решений имеет глубина анализа данных, что в свою очередь непосредственно зависит от темпоральных возможностей конкретной информационной системы.

3. Постановка задачи

В настоящее время на рынке коммерческих баз данных отсутствуют СУБД, обладающие полноценными темпоральными возможностями. Вопрос построения полноценной темпоральной СУБД уже на протяжении нескольких десятилетий является весьма актуальным. За это время было предложено множество различных подходов и методов, сформулировано множество принципов и теорем. Теория проектирования темпоральных моделей данных является вполне самостоятельной областью исследований.

Ввиду отсутствия на сегодняшний день полноценных темпоральных СУБД [2], для реализации темпоральных возможностей в рамках ИС программистам, как правило, приходится разрабатывать специальные средства, расширяющие и дополняющие существующие реляционные модели. Весьма распространенной проблемой разработки

таких приложений является отсутствие полного понимания того, каким образом и на каком уровне должна быть осуществлена поддержка темпоральности в БД. Многими разработчиками, реализующими темпоральность в ИС, не учитывается тот факт, что за несколько десятилетий существования данной области исследований, накоплено множество различных подходов и методик, изучение которых помогло бы избежать многих традиционных ошибок и заблуждений [2].

Целью данной работы является изучение различных методов построения темпоральных моделей данных, а также реализация одного из методов в рамках подсистемы МИС Интерин PROMIS. В статье представлено краткое введение в проблематику и аналитический обзор существующих методов построения, обобщен опыт реализации темпоральной модели данных в рамках подсистемы управления персоналом МИС Интерин PROMIS, сформулированы основные требования к реализации и методологические принципы ее построения.

4. Введение в проблематику

Темпоральная модель данных (ТМД) — это модель данных, ориентированная на хранение темпоральных данных, все аспекты которой также должны быть темпоральными. Традиционная модель данных $M = (DS, OP, C)$ состоит из трех компонент: структура данных DS , операции OP и ограничения целостности C . Темпоральная модель данных $MT = (DST, OPT, CT)$ должна поддерживать все понятия, входящие в каждое из трех компонент, с учетом изменений данных во времени [1]. Структура данных должна быть адаптирована таким образом, чтобы она могла хранить темпоральные данные. Алгебра и операции модификации должны быть переопределены, используя темпоральную семантику. Дополнительно, для каждого ограничения целостности в нетемпоральной модели данных M , темпоральная модель данных MT должна поддерживать темпоральный аналог нетемпорального ограничения. Семантика темпоральных ограничений целостности также должна быть переопределена.

Таким образом, разработка ТМД предполагает развитие следующих составляющих:

- Темпоральная структура данных.
- Темпоральные ограничения целостности, ключи.
- Темпоральные запросы и модификации.
- Темпоральная алгебра.

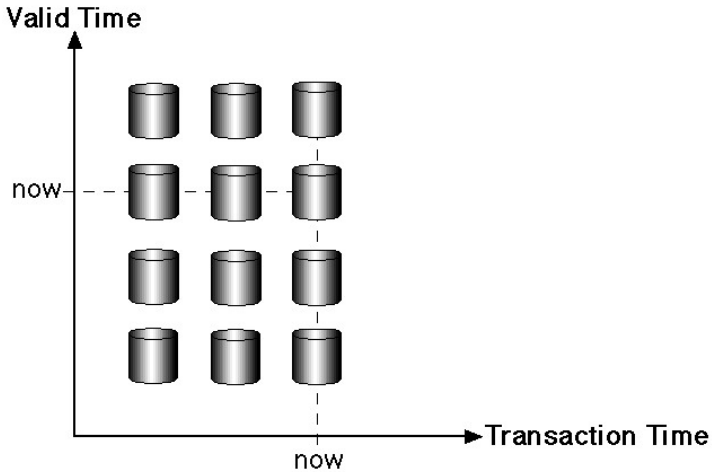


Рис. 1. Битемпоральная модель данных

Рассматривая данные, представленные в БД, в качестве некоторого отражения текущего состояния действительности в моделируемом мире, каждая запись может быть воспринята как некоторый факт, который является истинным в определенный момент. При переходе к ТБД для каждого факта можно указать промежуток времени, в течение которого этот факт являлся истинным в моделируемом мире, представленном в БД. Представление времени, когда с данными связывается промежуток времени их актуальности (с точки зрения моделируемого мира), называется модельным или действительным (valid) временем [2]. Другим типом линии времени в теории ТБД является транзакционное время. В любой СУБД каждой записи базы данных можно сопоставить некоторый промежуток времени, когда данная запись была представлена в БД, то есть промежуток времени между моментами создания и удаления записи в БД [2].

Исследователями ТМД выделяются три фундаментальных типа темпоральных данных [1]:

- Момент времени (instant) (событие, которое произошло или произойдет в определенный момент времени, например, сейчас или 1 августа 2009 года в 13.40).

- Интервал времени (interval) (длительность временного отрезка, например, 2 года).
- Период времени (period) (конкретный отрезок времени, например, с 23 апреля 2007 года по 1 августа 2009 года).

Битемпоральная модель данных оперирует как модельным, так и транзакционным временем. Именно битемпоральная модель является наиболее востребованной моделью в большинстве ИС (рис. 1).

5. Краткий обзор методов представления данных

Модифицирование реляционной модели данных с целью обеспечения поддержки работы с темпоральными данными предполагает изменения модели на уровне СУБД. Однако устройство большинства СУБД представляет собой «черный ящик», изменения в котором не представляются возможными. Поэтому основные способы обеспечения поддержки темпоральных данных заключаются в поддержке темпоральной функциональности на уровне приложения либо расширении реляционной модели данных до темпоральной.

Фактически, на практике существует два принципиальных подхода к реализации ТМД:

- Реализация темпоральной поддержки на уровне приложения.
- Расширение нетемпоральной модели данных до темпоральной.

В литературе встречаются и другие способы такие, как генерализация модели данных до темпоральной и использование абстрактных типов, однако на практике использование этих подходов имеет ряд существенных сложностей [3]. Метод реализации темпоральности на уровне приложения предполагает разработку специальных собственных средств поддержки темпоральности на уровне приложения. Однако на практике данный подход приводит к существенным проблемам, например, когда требуется изменить или заменить часть кода в приложении. Темпоральная семантика в таком случае проектируется каждым разработчиком заново. Темпоральная логика, реализованная на уровне приложения, может быть удобным ситуационным решением, но не дальновидной стратегией проектирования ИС.

Расширение нетемпоральной модели данных до темпоральной модели означает, что для спецификации темпоральных понятий используются основные концепции, поддерживаемые нетемпоральной моделью данных. Язык запросов и алгебра расширяются дополнительными операциями для того, чтобы иметь возможность описывать темпоральные операции с данными. На практике этот подход расширения схемы данных наиболее широко используется для построения ТМД. Его преимущество состоит в том, что данный метод предполагает изменение лишь отдельных частей модели, например, языка запросов или ограничения целостности. Метод доступа к информации и структура данных остаются неизменными.

В рамках данного подхода предложены различные ТМД. Принципиальные отличия этих моделей друг от друга можно разделить по следующим критериям:

- Тип темпоральных данных (дискретное или интервальное представление времени).
- Обеспечение темпоральности на уровне отдельных атрибутов или на уровне кортежа.

Темпоральные данные могут быть связаны как с дискретным представлением времени — моментом времени, так с интервальным представлением. Преимущество модели, основанной на дискретном представлении, заключается в ее простоте с точки зрения поддержки стандарта SQL-92 [4]. Однако связь темпоральных объектов с одним атрибутом времени может усложнить и без того непростые темпоральные запросы и операции. В этом плане проще в реализации оказывается модель с интервальным представлением времени. Одним из недостатков этого подхода является отсутствие поддержки понятия «интервала» в стандарте SQL-92. Это понятие может быть смоделировано посредством использования двух моментов времени.

Второй критерий построения темпоральных моделей данных приводит к появлению нескольких различных моделей данных:

- Модель представления темпоральных данных, предложенная Р. Снодграсом (Snodgrass R.) [5].

Пусть битемпоральное отношение R имеет набор атрибутов A_1, \dots, A_n, T , где T — битемпоральный атрибут, определенный на множестве битемпоральных элементов. Тогда R можно записать в следующем виде:

$$R = (A_1, \dots, A_n, T_s, T_e, V_s, V_e).$$

Набор дополнительных атрибутов T_s, T_e, V_s, V_e состоит из

атомарных темпоральных атрибутов времени, содержащих дату начала и окончания транзакционного и модельного времени. Данная модель данных является самым естественным и наиболее часто используемым способом представления битемпоральных отношений.

- Модель представления темпоральных данных, предложенная К. Дженсеном (Jensen C. S.) [6].

Особенность данного представления заключается в том, что историчные кортежи никогда не обновляются, то есть доступны только для чтения. Таким образом, это представление данных хорошо подходит для основанного на архивах хранения битемпоральных отношений. Этот подход особенно актуален в свете тенденции увеличения объемов носителей информации с одновременным снижением их стоимости. Битемпоральное отношение R с набором атрибутов A_1, \dots, A_n может быть представлено в следующем виде:

$$R = (A_1, \dots, A_n, V_s, V_e, T, Op).$$

Как и в предыдущей схеме представления данных, атрибуты V_s и V_e хранят даты начала и окончания актуальности факта в моделируемой реальности соответственно. Атрибут T хранит информацию о времени внесения кортежа в журнал изменений. Запросы на создание и удаление кортежей обозначаются в атрибуте Op соответствующими символами — I (Insert) и D (Delete). Модификации данных представляет собой пару запросов: удаления и создание записи, с одинаковым атрибутом времени T .

- Модель представления темпоральных данных, предложенная С. Гадией (Gadia S. K.) [7].

Данный подход предполагает наличие битемпоральных меток у каждого из атрибутов кортежа, что обеспечивает возможность более гибкого моделирования реальности. Пусть битемпоральное отношение R имеет атрибуты A_1, \dots, A_n, T , где T — темпоральный атрибут, определенный на множестве битемпоральных элементов. Тогда битемпоральное отношение R может быть представлено в виде отношений, где каждый из атрибутов имеет свою темпоральную метку:

$$R = (([T_s, T_e][V_s, V_e]A_1), \dots, ([T_s, T_e][V_s, V_e]A_n)).$$

Кортеж состоит из n элементов. Каждый элемент представляет собой тройку значений: транзакционное время $[T_s, T_e]$, модельное время $[V_s, V_e]$ и значение атрибута A_i .

- Модель представления темпоральных данных, предложенная Е. МакКензи (McKenzie E.) [8].

В предложенной модели битемпоральное отношение — это последовательность состояний в модельном времени, проиндексированная транзакционным временем. В кортежах с модельным временем атрибуты имеют свои темпоральные метки. Битемпоральное отношение R с набором атрибутов A_1, \dots, A_n представлено в виде отношения, где каждый атрибут помечается временной меткой:

$$R = (VR, T),$$

где VR — это отношение в модельном времени, T — транзакционное время. Схема состояний модельного времени имеет вид:

$$VR = (A_1V_1, \dots, A_nV_n).$$

Здесь A_1, \dots, A_n — это набор атрибутов, V_i — это атрибут модельного времени, связанный с каждым атрибутом A_i и обозначающий время актуальности значения атрибута A_i в моделируемой реальности.

- Модель представления темпоральных данных, предложенная Дж. Бен-Зви (Bez-Zvi J.) [9].

Пусть битемпоральное отношение R состоит из набора атрибутов A_1, \dots, A_n, T , где T — темпоральный атрибут, определенный на множестве битемпоральных элементов. Тогда R может быть представлено в модели Бен-Зви следующим образом:

$$R = (A_1, \dots, A_n, T_{es}, T_{rs}, T_{ee}, T_{re}, T_d).$$

В кортеже значение атрибута T_{es} (effective start) — это время, когда значение атрибута кортежа начинает быть актуальным. Атрибут T_{rs} (record start) хранит информацию о том, когда T_{es} было сохранено в БД. Аналогично, T_{re} хранит информацию о том, когда факт перестает быть актуальным в моделируемой реальности, а T_{ee} — когда T_{re} было зафиксировано в БД. Последний атрибут T_d указывает на время, когда запись была логически удалена из БД.

Кроме этого, темпоральные модели данных могут быть различными по дополнительным критериям таким, как возможность работы с ошибочно введенными данными. Существует ряд методов, предложенных отечественными авторами, суть которых сводится к расширению традиционной модели до темпоральной посредством введения

дополнительных таблиц-связей [10]. Авторы данных работ доказывают жизнеспособность и пригодность данных методов на примере разработки прикладного программного обеспечения, однако реализованные авторами темпоральные возможности не являются полноценными с точки зрения определения ТМД.

6. Реализация темпоральной модели в МИС Интерин PROMIS

МИС Интерин PROMIS представляет собой информационную и функциональную среду, объединяющую элементы различных классов медицинских информационных систем. МИС обеспечивает комплексную автоматизацию и информационную поддержку всех служб медицинского учреждения. Важное место в МИС занимает подсистема управления кадрами, ее эффективная работа является необходимым условием нормального функционирования учреждения [11]. Подсистема управления кадрами предназначена для автоматизации работы с кадровым составом учреждения, в функции которой входит сквозное ведение штатного расписания, а также оформление и проведение в системы приказов по кадрам на дату как в прошлом, так и будущем [12]. Реализации темпоральности в данной подсистеме имеет ряд системотехнических сложностей, для решения которых потребовалось принятие научно обоснованных решений архитектурного и методологического характера.

Далее описаны некоторые моменты реализации темпоральной модели данных за счет расширения существующей реляционной модели. Автором сознательно выбраны наиболее важные аспекты реализации темпоральности, которые, по его мнению, заслуживают наибольшего внимания.

6.1. Выбор метода

В качестве основной модели структуры темпоральных данных выбрана модель, предложенная Ричардом Снодграсом (Snodgrass R.) и реализующая темпоральные характеристики на уровне кортежа. Данный метод выбран по причине наибольшей естественности и простой реализации относительно других методов [3].

6.2. Модель данных

Модель данных подсистемы управления кадрами представляет собой набор более чем из тридцати таблиц, двенадцать из которых

являются темпоральными. Темпоральную модель данных мы рассмотрим на примере четырех таблиц — K_PERSONS, K_ISPOLs, K_DOLS и K_DOL_DICT. Таблица K_PERSONS содержит тридцать четыре поля, для примера мы будем рассматривать только некоторые из них: PERSON_ID, FAMILY, NAME. Таблица K_ISPOLs состоит из двадцати восьми полей, включая PERSON_ID и DOL_ID. Таблица K_DOLS содержит восемнадцать полей, включая DOL_ID и DOL_DICT_ID. В таблице K_DOL_DICT два поля.

Таким образом, нас интересуют следующие таблицы и атрибуты:

- K_PERSONS (PERSON_ID, FAMILY, NAME);
- K_ISPOLs (PERSON_ID, DOL_ID, SALARY);
- K_DOLS (DOL_ID, DOL_DICT_ID);
- K_DOL_DICT (DOL_DICT_ID, DOL_NAME).

Таблица K_PERSONS содержит информацию о сотрудниках организации. Таблица K_DOLS содержит список должностей организации, K_ISPOLs содержит информацию о том, какую должность занимает сотрудник. K_DOL_DICT является справочником должностей. PERSON_ID — первичный ключ (ПК) таблицы K_PERSONS, ПК таблицы K_ISPOLs состоит из пары атрибутов (PERSON_ID, DOL_ID), ПК таблицы K_DOLS — DOL_ID.

6.3. Добавление темпоральности

Для того, чтобы темпоральная модель данных стала битемпоральной, в таблицы были добавлены по четыре темпоральных атрибута. Первая пара атрибутов (ACTUAL_FROM, ACTUAL_TO) отражает период актуальности информации в моделируемой реальности (модельное время). Вторая пара (IN_DATE, OUT_DATE) отражает время фактической регистрации факта в БД и время его логического удаления (транзакционное время).

6.4. Темпоральные ключи

Для таблиц с темпоральной поддержкой также требуется модификация состава первичных ключей — необходимо включить в них темпоральные атрибуты. Значение первичного ключа таблицы должно быть уникальным. Для оригинальной таблицы K_ISPOLs значение пары (PERSON_ID, DOL_ID) уникально в любой момент времени, что означает, что ни один из сотрудников не может числиться больше чем на одной должности (речь идет об основных видах исполнения). С добавлением темпоральной поддержки в данной таблице

пары значений (PERSON_ID, DOL_ID) могут повторяться. Добавление темпорального атрибута ACTUAL_FROM или ACTUAL_TO в состав ключа не решает проблемы темпоральных ключей. Проблема остается в том случае, когда даты модельного времени отличаются, например, на один день, то есть периоды актуальности данных пересекаются, что приводит к неуникальности значений первичного ключа на некоторых промежутках времени. Для решения этой проблемы была разработана более сложная конструкция — последовательное (sequenced) условие уникальности в каждый момент времени (Листинг 1).

Листинг № 1 «Первичный темпоральный ключ»

```
WHERE NOT EXISTS (SELECT *
                   FROM K_ISPOLS I1 ,
                        K_ISPOL I2
                   WHERE I1.PERSON_ID = I2.PERSON_ID
                        AND I1.DOL_ID = I2.DOL_ID
                        AND I1.ACTUAL_FROM < I2.ACTUAL_TO
                        AND I2.ACTUAL_FROM < I1.ACTUAL_TO
                        AND I1.rowid <> I2.rowid )
                   AND NOT EXISTS (
                       SELECT *
                       FROM K_ISPOLS I1
                       WHERE I1.K_PERSON_ID IS NULL OR
                             I1.DOL_ID IS NULL)
```

6.5. Темпоральная уникальность

Для оригинальной таблицы K_ISPOLS условие уникальности может быть записано в виде UNIQUE (PERSON_ID, DOL_ID). Однако с добавлением темпоральной поддержки в таблицы этого условия оказывается недостаточно. Недостаточным является и добавление одного темпорального атрибута или пары атрибутов UNIQUE (PERSON_ID, DOL_ID, ACTUAL_FROM, ACTUAL_TO), поскольку пара тех же значений (PERSON_ID, DOL_ID) может быть добавлена с датами, отличающимися, например, на один день. В этом случае условие уникальности выполняться не будет. Для решения этих проблем были написаны более сложные условия уникальности следующего вида (Листинг 2).

6.7. Темпоральные запросы

В теории темпоральных баз данных выделяют три фундаментальных типа запросов и модификаций [1]:

- Текущие (current).
- Последовательные (sequenced).
- Произвольные (non sequenced).

Запросы к оригинальным таблицам до добавления темпоральной поддержки соответствуют текущему состоянию моделируемой реальности. Запросы к битемпоральным таблицам приобретают специфику. Рассмотрим более подробно различные типы запросов.

6.7.1. Текущие запросы

Текущий запрос представляет собой запрос значений на некоторый момент времени в прошлом, то есть создание среза истинности фактов на произвольную дату. Например, для обычного реляционного запроса «какую зарплату сейчас получает каждый из сотрудников?» можно легко сформулировать его темпоральный аналог «какую зарплату получал каждый из сотрудников в указанную дату?» В этом случае результат запроса останется в рамках реляционного представления [2].

6.7.2. Последовательные запросы

Более сложным случаем являются последовательные запросы. Вполне естественным оказывается запрос «когда и какую зарплату получал каждый из сотрудников?» Здесь уже в результатах запроса появляется линия времени. Алгоритм формирования результатов подобных запросов можно упрощенно представить следующим образом: для каждого момента времени вычисляется реляционный подзапрос «какую зарплату получает каждый из сотрудников», после чего к общему результату добавляются результаты этих подзапросов с учетом интервалов истинности. Подобная семантика «последовательной» интерпретации реляционных запросов называется последовательной [2].

Для случая последовательных запросов рассмотрим более подробно особенности операций выборки и связывания.

Последовательная выборка данных не требует особых средств поддержки темпоральности и достаточно просто реализуется.

Более сложной задачей является связывание двух темпоральных таблиц. Например, для вычисления зарплаты и должности для каждого сотрудника мы должны вычислить значение заработной платы для каждого момента времени.

История заработной платы хранится в таблице K_ISPOLS. Прямое связывание двух таблиц K_ISPOLS и K_PERSONS для каждого момента времени неэффективно, потому что значения заработных плат и информация о сотрудниках остаются неизменными в течение длительных периодов. Поэтому эти таблицы следует связывать с использованием их темпоральных периодов. Возможны два различных случая:

- Период в первой из двух связываемых таблиц является более продолжительным, чем период во второй таблице.
- Период во второй из двух связываемых таблиц является более продолжительным, чем период в первой таблице.

Таким образом, выборки темпоральных данных в случае, когда все таблицы имеют темпоральную поддержку, становятся более сложными и состоят, как правило, из нескольких подзапросов, в зависимости от пересечений и наложений периодов темпоральности. Так, например, запрос на выборку истории зарплат всех сотрудников потребует рассмотрения двух общих случаев наложения периодов таблиц в одном запросе.

6.8. Темпоральные модификации

6.8.1. Текущие модификации

Текущие модификации в общем случае предполагают обновление записи и изменение периода актуальности с некоторого момента в прошлом и по настоящее время [1].

Операции создания записи в этом случае требуют дополнительных явных условий проверки в блоке WHERE на ограничения ссылочной целостности и уникальности.

В общем случае текущее удаление записи представляет собой обновление даты окончания периодов модельного и транзакционного времени.

Текущее обновление записи в общем случае состоит из следующей последовательности действий:

- Создание новой записи с соответствующими битемпоральными атрибутами.

- Обновление всех записей, дата актуальности которых больше даты начала действия созданной записи.

6.8.2. Последовательные модификации

Последовательные модификации предполагают обновление записи в некоторый период времени в прошлом, то есть границы актуальности этого периода на оси модельного времени располагаются до настоящего момента [1].

Создание новой записи в случае последовательных модификаций осуществляется при выполнении следующих условий:

- в этом периоде актуальности нет дубликатов записи;
- для этого периода есть актуальное значение в таблице, на которую ссылается новая запись;
- нет разрывов во временной оси модельного времени.

При удалении записи возможны четыре различных варианта пересечения периодов актуальности оригинального периода (ОП) и удаляемого периода (УП): УП целиком входит в ОП, УП начинается в рамках ОП, УП заканчивается в рамках ОП и ОП целиком входит в УП. В общем случае удаление записи в случае последовательных модификаций состоит из следующих действий:

- Копируется запись ОП, атрибут ACTUAL_FROM задается равным дате окончания УП.
- Для этой записи атрибут ACTUAL_TO задается равным дате начала УП.
- Для этой записи атрибут ACTUAL_FROM задается равным дате окончания УП.
- Удаляются записи ОП, которые целиком входят в период УП.

В случае обновления записи также возможны четыре различных варианта взаимного расположения двух периодов — оригинального (ОП) и модифицируемого (МП). Обновление записи при последовательных модификациях представляет собой следующий набор операторов:

- Копируется запись ОП, атрибут ACTUAL_TO задается равным дате начала МП.
- Копируется запись ОП, атрибут ACTUAL_FROM задается равным дате окончания МП.

- Обновляются необходимые атрибуты у тех записей, период актуальности которых пересекаются с МП.
- Атрибут ACTUAL_FROM задается равным дате начала МП для тех записей, период актуальности которых пересекает МП.
- Атрибут ACTUAL_TO устанавливается равным дате окончания МП для тех записей, период актуальности которых пересекает МП.

6.9. Произвольные запросы и модификации

Запросы и модификации произвольного типа оперируют темпоральными данными некоторым произвольным образом, являются достаточно редкими и должны рассматриваться отдельно в каждом конкретном случае. Например, к таким запросам можно отнести запросы, требующие «сравнения» нескольких последовательных моментов времени, обычно включающие агрегационные функции «во времени», например, «вывести среднюю заработную плату сотрудника за все периоды времени» [2]. Запросы данного типа при реализации ТМД в рамках системы управления кадрами МИС Интерин PROMIS не рассматривались.

7. Выводы

Исследования в области темпоральных баз данных ведутся уже более трех десятилетий и до сих пор остаются актуальными и востребованными в наше время. За это время было сформулировано множество методик и подходов построения темпоральных баз данных, предложено множество различных способов построения темпоральных моделей данных. Функциональные возможности ИС, разработанной на базе темпоральной БД, поднимаются на качественно новый уровень. Практически все данные, которыми оперируют ИС, являются темпоральными, то есть в той или иной мере связаны с динамикой изменения во времени. Особое значение корректная и оперативная работа с темпоральными данными приобретает в сфере медицинских информационных технологиях, где качество информации, в конечном счете, может оказывать влияние на здоровье человека.

В настоящее время на рынке коммерческих баз данных практически отсутствуют СУБД, обладающие полноценными темпоральными возможностями [2]. Единственным наиболее перспективным решением в сложившейся ситуации может быть построение ТМД в рамках

расширения реляционной модели. Понятие ТМД включает в себя темпоральные структуры данных, темпоральные ключи и ограничения целостности, а также темпоральные запросы [1].

В данной работе выполнен обзор основных темпоральных структур данных. На основе наиболее естественного и простого из них реализована ТМД в подсистеме управления персоналом МИС Интерин PROMIS. Подсистема управления персоналом является интегрированной частью МИС Интерин PROMIS, где особенно востребованы темпоральные возможности. Изложенный подход построения темпоральной модели успешно реализован и апробирован на практике.

Однако, несмотря на положительный опыт внедрения описанных разработок, использование темпоральной модели порождает ряд проблем, многие из которых достаточно существенны. Так, например, скорость работы запросов в темпоральной модели в некоторых ситуациях значительно замедляется. Нераскрытыми остались проблемы быстрогодействия темпоральных моделей, возможности редактирования ошибочно введенных данных, а также вопросы сравнения различных моделей в условиях практического использования, что может послужить направлением для дальнейших исследований данной проблематики.

Список литературы

- [1] Snodgrass R. Developing Time-Oriented Database Applications in SQL: Morgan Kaufmann Publishers, 1999. ↑1, 4, 4, 6.7, 6.8.1, 6.8.2, 7
- [2] Костенко Б. Б. История и актуальные проблемы темпоральных баз данных. — МГУ, 2007, Эл. ресурс: <http://www.citforum.ru/database/articles/temporal>. ↑3, 4, 4, 6.7.1, 6.7.2, 6.9, 7
- [3] Jensen C. S., Soo M. D., Snodgrass R. T. Unifying Temporal Data Models Via a Conceptual Model: Information Systems Vol 19, No. 7, 1994. — 513-547 с. ↑5, 6.1
- [4] JCC's SQL Standards Page: JCC Consulting, 2007, Эл. ресурс: <http://jcc.com/SQL.htm>. ↑5
- [5] Snodgrass R., Ahn I. A Taxonomy of Time in Databases: In Proceedings of ACM SIGMOD, 1985. ↑5
- [6] Jensen C. S., Snodgrass R., Soo M. D. Extending Normal Forms to Temporal Relations. — Tucson, AZ: Technical Report TR-92-17, Department of Computer Science, University of Arizona, 1992. ↑5
- [7] Gadia S. K., Yeung C. S. A Generalized Model for a Relational Temporal Database: In Proceedings of ACM SIGMOD, 1998. ↑5
- [8] McKenzie E., Snodgrass R. Supporting Valid Time in an Historical Relational Algebra: Proofs and Extensions. — Tucson, AZ: Technical Report TR-91-15, Department of Computer Science, University of Arizona, 1991. ↑5

- [9] Ben-Zvi J. The Time Relational Model. — UCLA: PhD thesis, Computer Science Department, 1982. ↑5
- [10] Порай Д. С., Соловьев А. В., Корольков Г. В. Реализация концепции темпоральной базы данных средствами реляционной СУБД // Тр. Института системного анализа Российской академии наук «Документооборот. Концепции и инструментарий» / Под редакцией Арлазарова В.Л., Емельянова Н.Е.: Едиториал УРСС, 2004. — 92-109 с. ↑5
- [11] Назаренко Г. И., Гулиев Я. И., Ермаков Д. Е. Медицинские информационные системы: теория и практика / Под редакцией Г. И. Назаренко, Г. С. Осипова. — Москва: Наука. Физматлит, 2005. ↑6
- [12] Базаркин А. Н., Хаткевич М. И., Хаткевич Ю. И. Подсистема управления кадрами в интегрированных медицинских информационных системах // Тр. междунар. конф. «Программные системы: теория и приложения», ИПС РАН, Переславль-Залесский, 2006: В 2 т. / Под ред. С.М. Абрамова. — М.: Наука. Физматлит, 2006. — Т. 2. — 113-124 с. ↑6

Учреждение Российской академии наук Институт программных систем им А.К. Айламазяна РАН Исследовательский центр медицинской информатики, аспирантура

A. N. Bazarkin. *Research and developing temporal data model in MIS Interin PROMIS subsystem* // Proceedings of Program Systems institute scientific conference “Program systems: Theory and applications”. — Pereslavl-Zaleskij, v. 2, 2009. — p. 37-54. — ISBN 978-5-901795-18-7 (*in Russian*).

АБСТРАКТ. В статье описаны основные методы построения темпоральной реляционной модели данных. Сформулированы основные концепции теории построения темпоральных баз данных, выявлены концепции темпоральных моделей данных. В статье рассмотрены некоторые критерии методов классификации, а также опыт реализации темпоральной модели как части интегрированной подсистемы MIS Interin PROMIS.