

В.Г. Григорович, О.Ю. Косовська, О.М. Пігур-Пастернак, А.Ю. Шілінг
Дрогобицький державний педагогічний університет імені Івана Франка

ОГЛЯД ТЕХНОЛОГІЙ ТЕМПОРАЛЬНИХ БАЗ ДАНИХ

© Григорович В.Г., Косовська О.Ю., Пігур-Пастернак О.М., Шілінг А.Ю., 2011

Розглянуто характерні особливості темпоральних баз даних, їхні основні поняття та історію, описано методи представлення темпоральних даних.

Ключові слова: база даних, темпоральні бази даних, темпоральні відношення, часовий інтервал, модель даних.

The article reviews the characteristic features of temporal databases, their basic concepts and history, describes methods of presentation of temporal data.

Key words: database, temporal database, temporal relation, time interval, data model.

1. Постановка проблеми

Урахування залежностей даних від часу є важливим у багатьох галузях діяльності.

Сформулюємо основні завдання інформаційних систем, які потребують застосування темпоральних технологій:

- організація зберігання і доступу до даних, які мають темпоральні характеристики;
- необхідність зберігання інформації, яка перестає бути актуальною з часом, але знання про неї в будь-який момент часу в минулому необхідне;
- автоматичне збереження попередніх станів об'єкта з метою обліку змін і доступу до них у разі необхідності;
- зберігання і опрацювання поточних даних.

Зазначимо, що ТБД у 1990–2000-х роках не отримали широкого практичного застосування. Фактором, що гальмував розвиток технологій ТБД, були високі вимоги до ресурсів щодо підтримки багатоверсійності об'єктів зберігання. Але в зв'язку зі стрімким розвитком техніки цей фактор став менш важливим. Вартість інформації в наш час може набагато перевищувати вартість обладнання, яке використовується.

В останнє десятиліття (2000–2010) в області побудови баз даних сформувався окремий науковий напрям досліджень в сфері темпоральних баз даних, що охоплював питання моделювання даних, організації даних у зовнішній пам'яті, мов запитів тощо.

2. Аналіз останніх досліджень

Питання розроблення темпоральних баз даних сьогодні є актуальними. У статті розглянуто як сучасні роботи дослідників [8, 11, 12, 15, 16], так і класичні роботи [1, 2, 5, 7, 9] з темпоральних баз даних. Мета цих досліджень – ознайомлення з темпоральними базами даних, сучасним станом розробок, а також актуальними завданнями та подальшими шляхами розвитку цього напрямку. В дослідженнях [11] описано історію розвитку ТБД, але недостатньо уваги приділено моделям даних. А стаття [12] оснований на реалізації темпоральної моделі даних у медичній інформаційній системі. Проте в цих роботах не розглянуто різні моделі даних та їхню реалізацію.

3. Цілі статті

Метою цієї роботи є огляд та опис характерних особливостей темпоральних баз даних, а саме:

1. Характерні особливості темпоральних баз даних.
2. Основні принципи та особливості побудови темпоральних баз даних.
3. Методи представлення даних.

4. Основний матеріал

Темпоральні дані — це довільні дані, які явно або неявно пов'язані з певними датами або проміжками часу. Під таке визначення потрапляють майже всі дані та будь-яка інформація. Навіть

якщо немає явної залежності від часу для довільного твердження, то все одно для нього існує неявна залежність від часу, оскільки колись нам (чи системі) стало відомо, що таке твердження правильне. Крім того, є вірогідність, що в майбутньому твердження буде спростоване або на умови його застосовності будуть накладені певні обмеження; тому не можна вже розглядати його як деяку абсолютну істину, справедливу в усіх ситуаціях і у будь-який момент.

Темпоральні бази даних — це бази даних, що зберігають темпоральні дані. Проте ці бази даних і дані, що містяться в них, можна розглядати як темпоральні тільки у тому випадку, якщо відоме правило інтерпретації тимчасових міток та інтервалів для конкретної системи управління базами даних (СУБД). Щоб визначити, чи є певна СУБД темпоральною в повному розумінні цього слова, необхідно зрозуміти, чи можна окремо виділити і спеціально інтерпретувати дані атрибуту "час". У категорію темпоральних СУБД не потраплятимуть звичайні реляційні СУБД, в яких підтримуються пов'язані з часом типи даних, але інтерпретацією і зв'язком даних (чи подій) з урахуванням часу доводиться займатися розробникові. У "справжній" темпоральній СУБД враховуються специфічна природа часу і мінливість даних з плином часу [12].

4.1. Характерні особливості темпоральних баз даних

4.1.1. Лінії часу

На відміну від традиційних моделей даних, що забезпечують зберігання лише миттєвого знімка об'єктів предметної області, темпоральні моделі даних дають змогу зберігати інформацію про еволюцію об'єктів: для будь-якого об'єкта, який був створений у момент часу T_1 і закінчив своє існування в момент часу T_2 , в БД будуть збережені всі його стани на часовому проміжку $[T_1, T_2]$. Отже, в ТБД при кожній зміні стану об'єкта зберігатиметься запис у базі даних. Унікальний ідентифікатор такого запису складається з ключа об'єкта і часового інтервалу, на якому цей стан об'єкта був актуальним, і має такий вигляд:

$$\{key, [T_1, T_2]\}, \quad (1)$$

де key — ключ запису, $[T_1, T_2]$. — часова мітка запису.

Важливою особливістю ТБД є те, що в них можливі запити не тільки за ключем, але й за часом. Звідси, методи збереження темпоральних баз даних повинні ефективно підтримувати як запити за ключем, так і за часом.

Під темпоральністю об'єкта слід розуміти явний або неявний його зв'язок з певними датами або проміжками часу. У найширшому сенсі темпоральні дані — це дані, які можуть змінюватися з часом.

У повсякденному житті людина найчастіше не замислюється, що використовує тільки одну лінію часу. Події могли вже відбутися в минулому або тільки плануються в майбутньому, але час завжди вимірюється згідно з одним інтервалом. З іншого боку, в базі даних може зберігатися інформація про події та проміжки часу, що відповідають різним уявленням і зв'язкам. Якщо обробкою подібних даних займається сам користувач, то використовуваний тип часу можна назвати часом, визначеним користувачем. Його відмінною ознакою є відсутність інтерпретації з боку СУБД, оскільки обробка даних, пов'язаних з часом, повністю покладається на користувача. Фактично, всі сучасні СУБД забезпечують підтримку такого різновиду часу, наприклад, за допомогою введення спеціальних типів даних DATA або TIMESTAMP.

Якщо розглядати дані, подані в БД, як якесь відображення поточного стану дійсності для моделювання картини світу, кожен запис може бути сприйнятий як факт, який є істинним у певний момент або проміжок часу. Перейшовши до темпоральної БД, для кожного факту можна вказати той проміжок часу, в який цей факт відбувся в модельованому світі, представленим в БД. Таке уявлення часу, коли з даними зв'язується проміжок часу їхньої актуальності (з погляду модельованого світу), називається модельним, або дійсним (valid), часом. Його значення можна порівняти з показниками часу модельованого світу. Оскільки досить часто відображається саме реальний світ, то можуть бути задані співвідношення між значеннями часу реального світу і представленою в базі даних моделлю. Варто зазначити, що значення цього типу часу можуть бути

моменти часу як в минулому, так і в майбутньому. Крім того, ці значення цього типу часу можуть змінюватися, тобто істинність факту у визначені моменти часу може прийматися або відхилятися.

Іншим типом лінії часу, який розглядають дослідники темпоральних БД, є транзакційний час. У будь-якій СУБД кожному запису БД можна увідповіднити той проміжок часу, коли запис був представлений в БД, тобто проміжок часу між моментами додавання запису і його видалення з БД.

Операція оновлення, яка справді вносить зміни в запис, сприймається як складова операція видалення старого запису і додавання нового. Очевидно, що значення транзакційного часу не може стосуватися до майбутнього. У більшості СУБД транзакційний час використовується для роботи з блокуванням, журналом для відновлення системи. В деяких системах адміністратори навіть можуть застосовувати спеціальне розширення мови SQL, яке дає змогу отримати доступ до транзакційного часу та історії змін записів у БД [8].

Таблиця 1

Таблиця баз темпоральних розширень

Викладач	Навантаження
Шевченко Володимир Дмитрович	498
Романенко Іван Петрович	759

Щоб відповісти на запитання, як співвідносяться модельний і транзакційний час, розглянемо приклад. Нехай є таблиця, в якій зберігається інформація про навантаження викладача (табл. 1). За наявності підтримки дійсного часу ми могли б у будь-який момент сказати, яке у викладача було навантаження в довільний період часу (табл. 2). Отже, дані про навантаження можна подати як послідовність змінних значень. За наявності підтримки транзакційного часу ми могли б сказати, в який момент в таблицю внесено зміни (табл. 3).

Таблиця 2

Таблиця з підтримкою дійсного часу

Викладач	Навантаження	Дійсний час
Шевченко Володимир Дмитрович	498	3 1 січня 2010
Романенко Іван Петрович	620	3 1 вересня 2010 по 30 листопада 2010
Романенко Іван Петрович	759	3 1 грудня 2010

Таблиця 3

Таблиця з підтримкою транзакційного часу

Викладач	Навантаження	Транзакційний час
Шевченко Володимир Дмитрович	498	3 20 грудня 2009
Романенко Іван Петрович	620	3 3 вересня 2010 по 27 листопада 2010
Романенко Іван Петрович	800	3 28 листопада 2010 по 5 грудня 2010
Романенко Іван Петрович	759	3 6 грудня 2010

Припустимо, що для таблиці підтримується як дійсний, так і транзакційний час (табл. 4). Тоді у випадку, якщо неправильні дані як наслідок були виправлені, можна буде точно сказати, коли це було зроблено. Крім того, інформація про такі зміни необхідна, оскільки некоректні дані могли б бути вже використані у звітах. Тому в цьому випадку потрібна підтримка транзакційного часу. У разі оновлення значень в системі (навіть у разі виправлення помилки в даних) інтервал транзакційного часу також оновлюється, тому що можна розглядати список змін у базі даних.

Таблиця з підтримкою обох ліній часу

Викладач	Навантаження	Дійсний час	Транзакційний час
Шевченко Володимир Дмитрович	498	3 1 січня 2010	3 20 грудня 2009
Романенко Іван Петрович	620	3 1 вересня 2010 до 30 листопада 2010	3 3 вересня 2010 до 27 листопада 2010
Романенко Іван Петрович	800	3 1 грудня 2010	3 28 листопада 2010 до 5 грудня 2010
Романенко Іван Петрович	759	3 1 грудня 2010	3 6 грудня 2010

Звідси, часові мітки транзакційного часу надають інформацію про час зміни даних або виправлення помилок, а часові мітки дійсного часу зберігають інформацію про зміну деяких параметрів світу, що моделюється. Отже, модельний і транзакційний час є ортогональними один одному (рис.1) [10].

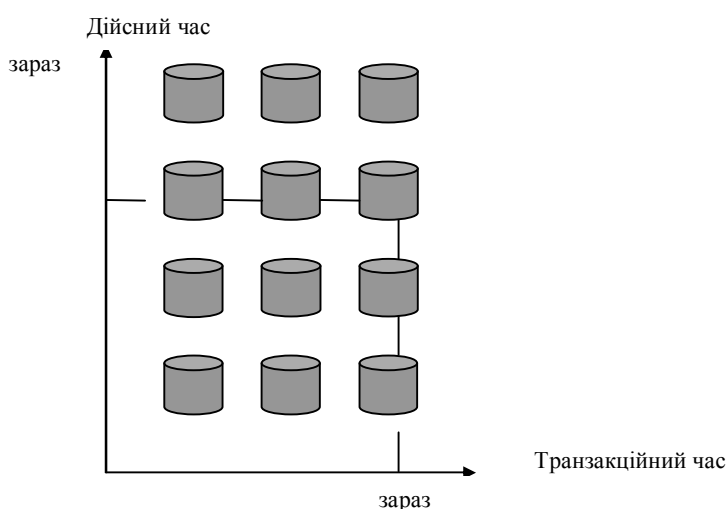


Рис. 1. Взаємовідношення між дійсним та транзакційним часом

Дослідники ТБД зазвичай використовують один із цих типів часу або обидва одночасно.

Говорячи про лінії часу, необхідно ввести ще один термін – *гранулярність*, яка показує, наскільки близькі моменти на осі часу все ще відрізнятимуться один від одного. Наприклад, можливо, що для даних про річне навантаження працівника достатньо розбиття по днях, але для транзакційного часу може бути недостатнім навіть поділ по секундах, якщо в СУБД можлива частіша фіксація транзакцій.

В загальному випадку з лінією часу може бути ще пов'язаний деякий *календар*, який визначає діапазони значень, гранулярність, відповідність і перетворення між моментами часу для різних його осей.

4.1.2. Інтервальне і точкове представлення

Як вже зазначалась, існує деякий інтервал, в якому конкретний факт є істинним. Це так зване *інтервальне* представлення. Але можна розглядати окремий момент часу і всі факти, які були істинними в цей конкретний момент. Тут ідеться про представлення часу з погляду користувача, тобто тих умовних моделей, у межах яких можуть формулюватися запити. У разі використання будь-якого з цих явищ істинність фактів не змінюється, але у випадку *точкового* представлення ми отримуємо зріз всіх фактів на конкретний момент часу, а для інтервального представлення нас цікавить конкретний факт і період його істинності. Якщо говорити про звичайну реляційну модель, то вона спирається на точкове представлення для актуального стану даних [9].

Дослідники виділяють три фундаментальні типи темпоральних даних [5]:

- момент часу (instant) (подія, яка сталася або станеться в певний момент, наприклад, «зараз» або «1 серпня 2009 року в 13.40»);
- інтервал часу (interval) (тривалість тимчасового відрізка, наприклад «2 роки»);
- період часу (period) (конкретний відрізок часу, припустимо, «з 23 квітня 2007 до 1 серпня 2010 року»).

4.2. Основні принципи і особливості побудови ТБД

Принципи побудови ТБД передбачають, що в БД повинні зберігатись всі версії об'єктів і користувач повинен мати можливість доступу до всіх попередніх версій. Часові розширення системи управління базами даних реалізують ці принципи, автоматично зберігаючи множину залежних від часу версій об'єктів даних.

Основним поняттям у темпоральній моделі є період *життя об'єкта* (lifespan). Він характеризує відрізок часу, який асоціюється з існуванням об'єкта в конкретному стані.

На відміну від БД взагалі, ТБД мають певні специфічні особливості, а їхня організація потребує застосування спеціальної технології – технологій ТБД. Звичайні БД зберігають миттєвий список моделі предметної області. Будь-яка зміна об'єкта в момент часу t приводить до недоступності стану цього об'єкта в попередній момент часу. Відбувається так зване «затирання» даних. Основними принципами технології ТБД є: зберігання всіх станів об'єкта з часу його створення до моменту знищення і надання доступу до будь-якого зі станів у довільний момент часу. Інакше кажучи, основна теза темпоральних систем полягає в тому, що для довільного об'єкта даних, створеного в момент часу t_1 і знищеного в момент часу t_2 , в БД зберігаються і доступні користувачам всі його стани в часовому інтервалі $[t_1, t_2)$.

Незважаючи на переваги та актуальність темпоральної технології й існування окремого наукового напрямку досліджень і розробок в області темпоральних баз даних, досі існують проблеми промислової реалізації темпоральних БД. До певної міри ці проблеми пов'язані із вибором платформи для темпоральних БД. На вибір платформ впливають не тільки вимоги до показників ефективності й функціональності БД, але і багато інших факторів.

Дослідження і побудова прототипів темпоральних СУБД зазвичай ґрунтуються на деякій реляційній СУБД. Темпоральна СУБД є надбудовою реляційної системи. Звісно, це не кращий спосіб реалізації з погляду ефективності, але є простим і доступним як для розробників систем, так і для їх користувачів. В результаті, сьогодні більша частина ТБД реалізована в середовищі реляційних СУБД, хоча існують і спеціалізовані темпоральні СУБД. Це пояснюється низкою фактів, які утруднюють використання останніх на великих підприємствах. Основними недоліками темпоральних СУБД сьогодні є:

- відсутність єдиного стандарту;
- нездійсненність розроблених у межах темпоральних СУБД задач;
- відчутне відставання за потужністю і надійністю промислових реляційних систем;
- слабка підтримка виробника.

Разом з тим, на користь реляційних СУБД свідчать такі факти:

- вони є випробуваними і такими, що добре себе зарекомендували упродовж декількох десятиліть, системами;
- виробники реляційних СУБД постійно розвивають свої продукти;
- підтримка загальноприйнятих стандартів;
- реляційне середовище поки що є кращим для зберігання і забезпечення цілісності даних;
- більшість підприємств має багаторічний успішний досвід роботи з реляційними системами;
- багато діючих прикладних систем розроблено за реляційною технологією.

Розглядаючи проблеми використання реляційних СУБД для роботи з часовими даними, можна передовсім відзначити таке:

- реляційні СУБД не володіють спеціалізованими засобами для розроблення ТБД, тоді як темпоральні СУБД створюються саме під цю задачу;

- реляційна СУБД не володіє знаннями про семантику часового поля відношень і не може контролювати коректність його значень;

- мови запитів реляційних СУБД не пристосовані для роботи з часом.

Зважаючи всі «за» і «проти», підприємства, як правило, жертвують вбудованою темпоральною функціональністю на користь надійності системи, вкладених інвестицій, напрацьованого досвіду і перспективи розвитку. Тому багато нових систем продовжують реалізовуватись в реляційних середовищах.

Незважаючи на практичну відсутність стандартних засобів створення темпоральних систем у реляційних СУБД, їх реалізація на базі більшості реляційних систем можлива, але потребує глибокої переробки моделі даних, знань темпоральної технології, а також особливостей вибраної для реалізації СУБД [4].

4.2.1. Традиційна модель даних

На ринку комерційних БД відсутні СУБД, що повною мірою забезпечують всі темпоральні можливості. Для їх реалізації в інформаційній системі (ІС) програмістам, як правило, доводиться розробляти спеціальні засоби, що розширюють і доповнюють реляційні моделі. Вельми поширеною проблемою розроблення таких програм є відсутність повного розуміння того, як і на якому рівні повинна підтримуватись темпоральність у БД.

Відповідно до усталених понять темпоральна модель даних (ТМД)— це модель даних, орієнтована на зберігання темпоральних даних, всі аспекти якої також мають бути темпоральними.

Традиційна модель даних:

$$M = (DS, OP, C)$$

складається з трьох компонентів: структура даних DS, операції OP і обмеження цілісності C, а темпоральна:

$$MT = (DST, OPT, CT)$$

повинна підтримувати всі поняття, що входять до кожного з трьох компонентів, з урахуванням змін даних у часі [5].

Структура даних повинна бути адаптована так, щоб можна було зберігати темпоральні дані. Алгебру та операції модифікації слід перевизначити, використовуючи темпоральну семантику. Додатково для кожного обмеження цілісності в нетемпоральній моделі даних M темпоральна, — MT, — повинна підтримувати темпоральний аналог нетемпорального обмеження. Семантика темпоральних обмежень цілісності також повинна бути перевизначена.

Отже, розроблення ТМД припускає розвиток таких темпоральних складових: структури даних, обмеження цілісності й ключі, запити і модифікації, алгебра.

4.2.2. Темпоральні моделі даних

Сьогодні в літературі найчастіше описують п'ять моделей ТБД. Моделі відрізняються за типом відношень і за способом представлення даних. Аналіз моделей, які описані в літературних джерелах, а також деякі практичні реалізації дають змогу виділити ряд типів часових відношень. Найпоширеніші із них можна розділити за категоріями.

- За типом діапазонів:

- відношення з часовими діапазонами, які не перетинаються;
- відношення з часовими діапазонами, які перетинаються;
- відношення, які не підтримують версійність об'єктів.

- За набором часових атрибутів:

- відношення з особливими часовими атрибутами (дні тижня, парність дня тощо);
- відношення з одним часовим атрибутом (дата початку, дата кінця або дата фіксації). Це може бути звичайне реляційне відношення, яке містить часову ознаку, яка властива об'єктові в житті (дата встановлення навантаження, дата звільнення тощо).

- Здатність зберігати помилково введені дані:

- підтримується;
- не підтримується.

4.2.3. Математичний опис моделі

Для початку опишемо темпоральну модель даних, яка призначена для зберігання об'єктів темпоральної системи.

Як оптимальну модель даних пропонується використовувати відкриту модель з абстрактним ідентифікатором об'єкта (Abstract Identify, AOID). В цій моделі життєвий цикл об'єкта O описується через життєві цикли всіх його властивостей, які визначені в різних відношеннях і мають часові атрибути $T = \{T_{start}, T_{end}\}$, що визначають відповідно час початку і час закінчення життєвого циклу.

Вважається, що об'єкт темпоральної системи O характеризується унікальним абстрактним ідентифікатором K^s і набором атрибутів A і може бути поданий у вигляді: $O = \{K^s, A\}$.

Набір атрибутів A , своєю чергою, ділиться на множину статичних атрибутів A^s , які не змінювалися з часом, і множину динамічних атрибутів A^d , які змінюються з часом, тобто: $A = A^s \cup A^d$.

В моделі не будемо розглядати множинні атрибути, які є в множинах A^s і A^d , оскільки в контексті, в якому ми розглядаємо, це не принципово і тільки призведе до зайвого ускладнення моделі.

Оскільки значення атрибутів у реляційних системах повинні бути атомарними, то для представлення динамічних атрибутів одного реляційного відношення недостатньо. Подамо об'єкт O у вигляді сукупності взаємопов'язаних реляційних відношень:

$$O = \{R^s, R_1^d, R_2^d, \dots, R_n^d\}$$

де $R^s = \{K^s, A^s\}$ — батьківське відношення, яке описує абсолютний ідентифікатор об'єкта K^s і яке містить атрибути об'єкта A^s .

$R_1^d, R_2^d, \dots, R_n^d$, — відношення, яке описує дискретно змінні динамічні атрибути $A^d = \{A_1^d, A_2^d, \dots, A_m^d\}$ і має схеми

$$R_i^d = \{K^s, K_i^d, T, A_i^d\}, i=1,2,\dots,n,$$

в яких $T = \{T_{start}, T_{end}, T_{tr}\}$ — вектор атрибутів часу, трійка (K^s, K_i^d, T_k) — первинний ключ, де T_k — довільний атрибут із T , A_i^d — дискретний динамічний атрибут, який входить до складу A^d , K_i^d — частина первинного ключа R_i^d , яка вирішує нетемпоральні відношення «один до більшості» до R^s . Якщо $K_i^d \neq 0$, нетемпоральний зв'язок буде «один до одного» (рис.2) [3].

4.3. Методи представлення темпоральних даних

Модифікування реляційної моделі даних для забезпечення підтримки роботи з темпоральними даними передбачає зміни моделі на рівні СУБД. Однак будова більшості СУБД — це «чорна скринька», зміни в якій не видаються можливими. Тому основні способи забезпечення підтримки темпоральних даних полягають у підтримці темпоральної функціональності на рівні програми або в розширенні реляційної моделі даних до темпоральної.

На практиці існують два принципові підходи до реалізації темпоральної моделі даних (ТМД):

- реалізація темпоральної підтримки на рівні програми;
- розширення нетемпоральної моделі даних до темпоральної.

У літературі описано й інші способи, такі як:

- генералізація моделі даних до темпоральної;
- використання абстрактних типів.

Але на практиці використання цих підходів пов'язане з певними складнощами. Метод реалізації темпоральності на рівні програми передбачає розроблення програмних засобів. Однак на практиці цей підхід призводить до суттєвих проблем, наприклад, коли потрібно змінити або замінити частину коду в програмному продукті. Темпоральну семантику в такому випадку проектує кожен розробник заново. Темпоральна логіка, реалізована на рівні програми, може бути зручним миттєвим рішенням, але недалекоглядною стратегією проектування інформаційної системи.

Розширення нетемпоральної моделі даних до ТМД означає, що для специфікації темпоральних понять використовуються основні концепції, підтримувані нетемпоральною моделлю даних. Мова запитів і алгебра розширюються додатковими операціями для того, щоб можна було описувати темпоральні операції з даними.

На практиці цей підхід розширення схеми даних найширше використовується для побудови ТМД. Його перевага полягає в тому, що передбачена зміна лише окремих частин моделі, наприклад, мови запитів або обмеження цілісності. Метод доступу до інформації і структура даних залишаються незмінними.

У межах цього підходу запропоновано різні ТМД. Критерії принципових відмінностей цих моделей такі:

- тип темпоральних даних (дискретне або інтервальне подання часу);
- забезпечення темпоральності на рівні окремих атрибутів або на рівні кортежу.

Темпоральні дані можуть бути пов'язані як з дискретним поданням часу — моментом, так і з інтервальним. Перевага моделі, основаної на дискретному поданні, полягає в її простоті з погляду підтримки стандарту SQL-92. Проте зв'язок темпоральних об'єктів з одним атрибутом часу може ускладнити і без того непрості темпоральні запити та операції. У цьому плані простішою в реалізації є модель з інтервальним поданням часу.

Одним з недоліків цього підходу є відсутність підтримки поняття інтервалу в стандарті SQL-92, воно може бути змодельоване використанням двох моментів часу.

Другий критерій побудови ТМД визначає п'ять підходів до подання темпоральних даних [7].

1. Модель подання темпоральних даних, яку запропонував Річард Снодграс (Richard Snodgrass).

Нехай бітемпоральне відношення R має набір атрибутів (A_1, \dots, A_n, T) , де T – бітемпоральний атрибут, визначений на множині бітемпоральних елементів. Тоді R можна записати у вигляді:

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

Додаткові атрибути T_s, T_e, V_s, V_e – це атомарні темпоральні атрибути часу, що містять дату початку і закінчення транзакційного (T_s, T_e) та модельного часу (V_s, V_e). Це подання відносин є найприроднішим і найчастіше використовуваним способом представлення бітемпоральних відносин.

2. Модель подання темпоральних даних, запропонована К. Дженсеном (Jensen C.S).

Особливість цього подання в тому, що історичні кортежі ніколи не оновлюються, тобто доступні лише для читання. Отже, це подання даних добре підходить для основаного на архівах зберігання бітемпоральних відносин. Бітемпоральне відношення R з набором атрибутів A_1, \dots, A_n можна записати у такому вигляді:

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

Як і в попередній схемі представлення даних, атрибути V_s і V_e зберігають дати початку та закінчення актуальності факту модельованої реальності відповідно. Атрибут T зберігає інформацію про час внесення кортежу в журнал змін. Запити на створення і видалення кортежів позначаються в атрибуті Op відповідними символами — I (Insert) і D (Delete). Модифікація даних являє собою пару запитів — видалення і створення запису — з однаковим атрибутом часу T .

3. Модель подання темпоральних даних, запропонована С. Гадіей (Gadia S.).

Цей підхід передбачає наявність бітемпоральних міток у кожного з атрибутів кортежу, що забезпечує можливість гнучкішого моделювання реальності. Нехай бітемпоральне відношення R має атрибути (A_1, \dots, A_n, T) , де T — темпоральний атрибут, визначений на множині бітемпоральних елементів. Тоді бітемпоральне відношення R можна представити у вигляді відносин, де кожен з атрибутів має свою темпоральну мітку:

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

Кортеж складається з n елементів. Кожен елемент є трійкою значень: транзакційний час $[T_s, T_e]$, модельний час $[V_s, V_e]$ і значення атрибуту A_i .

4. Модель подання темпоральних даних, запропонована Є. Мак-Кензі (McKenzie E.).

У цій моделі бітемпоральне відношення — це послідовність станів у модельному часі, проіндексованих транзакційним часом. У кортежі з модельним часом атрибути мають свої темпоральні позначки. Бітемпоральне відношення R з набором атрибутів A_1, \dots, A_n можна записати у вигляді відношення, в якому кожен атрибут позначається тимчасовою міткою:

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

де VR — відношення у модельному часі; T — транзакційний час.

Схема станів відношень модельного часу має вигляд:

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

де A_1, \dots, A_n — набір атрибутів; V_i — атрибут модельного часу, пов'язаний з кожним атрибутом A_i і позначає час актуальності значення атрибуту A_i у модельованій реальності.

5. Модель подання темпоральних даних, запропонована Дж. Бен-Зві (Jacob Ben – Zvi).

Нехай бітемпоральне відношення R складається з набору атрибутів (A_1, \dots, A_n, T) , де T — темпоральний атрибут, визначений на множині бітемпоральних елементів. Тоді R можна подати в моделі Бен-Зві так:

$$R = (A_1, \dots, A_n, Tes, Trs, Tee, Tre, Td).$$

У кортежі значення атрибуту Tes (effective start) — це час, коли значення атрибуту кортежу стає актуальним. Атрибут Trs зберігає інформацію про те, коли Tes було збережено в БД. Аналогічно Tre зберігає інформацію про те, коли факт перестає бути актуальним у модельованій реальності, а Tee — коли Tre було зафіксовано в БД. Атрибут Td вказує на час, коли запис був логічно видалений з БД.

Крім цього, темпоральні моделі даних можуть відрізнятися додатковими критеріями, такими як можливість роботи з помилково введеними даними. Існують методи, запропоновані різними авторами, суть яких зводиться до розширення традиційної моделі до темпоральної за допомогою введення додаткових таблиць-зв'язків [13].

5. Висновки

У роботі наведено огляд та опис характерних особливостей та принципи побудови БД, а також методи представлення даних.

Дослідження в області темпоральних баз даних сьогодні залишаються актуальними і необхідними, хоча і ведуться вже більше ніж тридцять років (1980 – 2010). За цей час сформульовано багато методів та підходів до побудови ТБД, запропоновано різні способи побудови темпоральної моделі даних. Функціональні можливості інформаційної системи, яка розроблена на базі ТБД, досягли якісно нового рівня. Будь-які дані в інформаційній системі є темпоральними, тобто певною мірою пов'язані з динамікою зміни в часі. Сьогодні на ринку комерційних БД практично немає СУБД з повноцінними темпоральними можливостями. Найефективнішим та перспективним вирішенням цієї ситуації, на наш погляд, є побудова ТМД у межах розширення реляційної моделі.

1. Ben-Zvi J. *The Time Relational Model. Ph.D. thesis/ Ben-Zvi J. – Computer Science Department – UCLA, 1982.* 2. E. F. Codd. *A Relational Model of Data for Large Shared Data Banks/ E. F. Codd. – Communications of the ACM, 13(6), 1970. – 377–387 p.* 3. Date C.J., Hugh Darwen, Nikos A. Lorentzos. *Temporal Data and the Relational Model: A Detailed Investigation into the Application of Interval and, Relation Theory to the Problem of Temporal Database Management/ Date C.J., Hugh Darwen, Nikos A. Lorentzos. – Morgan Kaufmann, December, 2002. – 480 p.* 4. Kopitovs J., Demidovs V., Petoukhova N. *Method of Temporal Databases design Using Relational Environment. In: Scientific proceedings of Riga Technical University – Computer Science: Applied Computer Systems, Series #5, Issue #13. Riga: RTU, 2002. – pp. 236-246.* 5. Snodgrass R. *Developing Time-Oriented Database Applications in SQL/ Snodgrass R. – Morgan Kaufmann Publishers, 1999.* 6. Richard T. Snodgrass, editor, Ilsoo Ahn, Gad Ariav, Don Batory, James Clifford, Curtis E. Dyreson, Ramez Elmasri, Fabio Grandi, Christian S. Jensen, Wolfgang Kaefer, Nick Kline, Krishna Kulkarni, T. Y. Cliff Leung, Nikos Lorentzos, John F. Roddick, Arie Segev, Michael D. Soo and Suryanarayana M. Sripada, *The TSQL2 Temporal Query Language Kluwer Academic Publishers, 1995.* Специфікація TSQL2 доступна за адресою <ftp://ftp.cs.arizona.edu/tsql/tsql2/bookspec.pdf>, а коментарі – на <ftp://ftp.cs.arizona.edu/tsql/tsql2/eval.pdf>. 7. Jensen C.S., Soo M.D., Snodgrass R.T. *Unifying Temporal Data Models Via a Conceptual Model/ Jensen C.S., Soo M.D.,*

Snodgrass R.T. – *Information Systems Vol. 19, No. 7, 1994.* – P. 513–547. 8. Paolo Terenziani, Richard T. Snodgrass. *Reconciling Point-based and Interval-based Semantics in Temporal Relational Databases: A Proper Treatment of the Telic/Atelic Distinction/ Paolo Terenziani, Richard T. Snodgrass – IEEE Transactions on Knowledge and Data Engineering, 16(5) – May, 2004.* 9. D. Toman. *Point-Based Temporal Extensions of SQL and Their Efficient Implementation. Temporal Databases: Research and Practice/ D. Toman.* – Springer, 1st edition. – July 1, 1998 10. What are temporal databases? [електронний ресурс] – <http://www.timeconsult.com/TemporalData/TemporalDB.html>. 11. Базаркин А.Н. Разработка темпоральной модели данных в медицинской информационной системе // Программные продукты и системы, 2009, № 2 – С. 34–40. 12. Костенко Б.Б., Кузнецов С.Д. История и актуальные проблемы темпоральных баз данных [електронний ресурс] – <http://citforum.ru/database/articles/temporal/>. 13. Порай Д.С., Соловьев А.В., Корольков Г.В. Реализация концепции темпоральной базы данных средствами реляционной СУБД // Документооборот. Концепции и инструментарий / ред. В.Л. Арлазаров, Н.Е. Емельянов: Эдиториал УРСС, 2004. – С. 92–109, Тр. Института системного анализа Российской академии наук. 14. Темпоральные базы данных [електронний ресурс] – <http://www.chair36.msiu.ru/articles/3/html/node56.html>. 15. Григорович В.Г., Шілінг А.Ю. Темпоральні бази даних: історія та основні характеристики // Матеріали сьомої Всеукраїнської науково-практичної інтернет-конференції «Сучасність, наука, час». – м. Київ, 18–20 листопада 2010 р. Режим доступу: <http://intkonf.org/kf-mn-grigorovich-vg-shiling-ayu-temporalni-bazi-danih-istoriya-ta-osnovni-harakteristiki/>. 16. Григорович В.Г., Косовська О.Ю. Темпоральні бази даних та методи представлення даних // Комп'ютерні науки та інженерія. Матеріали четвертої Міжнародної конференції молодих вчених CSE – 2010. – Львів: Видавництво Львівської політехніки. – С 68–69.

УДК 004.056, 004.75

В.Б. Дудикевич, Ю.Р. Гарасим

Національний університет “Львівська політехніка”,
Кафедра захисту інформації

ДОСЛІДЖЕННЯ МОДЕЛІ ОЦІНЮВАННЯ ЖИВУЧОСТІ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ КОРПОРАТИВНИХ МЕРЕЖ ЗВ'ЯЗКУ ЗА ДОПОМОГОЮ МЕРЕЖ ПЕТРИ

© Дудикевич В.Б., Гарасим Ю.Р., 2011

Розглянуто актуальну науково-практичну проблему розроблення моделей і методів оцінювання живучості систем захисту інформації корпоративних мереж зв'язку. Розроблено та проаналізовано модель оцінки живучості систем захисту інформації корпоративних мереж зв'язку.

Ключові слова: властивість живучості, оцінка живучості, системи захисту інформації.

The work is devoted to enterprise communication networks information security systems survivability assessment models and methods development as an actual scientific and practical problem solution. Within the paper enterprise communication networks information security systems survivability assessment model was developed and investigated.

Key words: survivability, survivability assessment, information security system.

Вступ

Для підвищення ефективності процесу забезпечення живучості систем захисту інформації (СЗІ) корпоративних мереж зв'язку (КМЗ) на етапі проектування та експлуатації виникла необхідність розробити адекватні моделі та методи її оцінювання. Ці моделі та методи повинні