УДК 004.422.8

О. Ролік, К. Ульяницька, О. Амонс

АДАПТАЦІЯ ВИЗНАЧЕННЯ ЧЕТВЕРТОЇ НОРМАЛЬНОЇ ФОРМИ ДЛЯ СУЧАСНОГО ПРОЄКТУВАННЯ БАЗ ДАНИХ

Анотація: Стаття присвячена проєктуванню реляційних баз даних в сучасних інформаційних системах. Проведений аналіз структури таблиць для створення скриптів реляційних систем керування базами даних та обґрунтоване твердження про те, що Четверта нормальна форма наразі залишається теоретичним додатком та з практикою використання вже не пов’язана.

Ключові слова: інформаційні системи, інформаційні технології, бази даних, нормалізація, четверта нормальна форма, складений первинний ключ.

Вступ

Реляційні бази даних історично є першими базами даних, що використовувались при побудові програмних застосунків. Реляційні бази даних базуються на реляційній моделі даних, принципи якої були сформовані ще в 1969-1970 рр. Е.Ф.Коддом [1]. Одним з головних етапів побудови реляційної бази даних є нормалізація даних. Предметно цей процес можна охарактеризувати як нормалізацію таблиць бази даних. Всі таблиці в таких базах даних повинні задовольняти певним правилам нормалізації – нормальним формам.

Сучасна трактовка нормальних форм базується на рукописі К. Дж. Дейта. «An Introduction to Database Systems». Останнє (8-е) видання цієї книги було опубліковане у 2003 році видавництвом Pearson/Addison Wesley [2]. Це видання залишається актуальним і широко використовується у вищих навчальних закладах, незважаючи на те, що з часу його публікації минуло понад 20 років.

Якщо відійти від теорії та торкнутися практики, то виявиться, що нині розробники баз даних користуються лише першими трьома нормальними формами, а Другу нормальну форму нівелюють завдяки використанню суцільних первинних ключів. Всі інші нормальні форми наразі є теоретичним додатком або не враховують етапу інфологічного проєктування, де всі головні залежності між атрибутами вже враховані в правилах атрибутів.

В роботах [3] та [4] започаткований аналіз нормальних форм та адаптація їхніх трактувань під сучасні умови експлуатації та розробки реляційних баз даних.

Друга нормальна форма (2НФ), незважаючи на те, що саме перші три нормальні форми вважаються постійно вживаними, насправді не використовується. В роботі [3]

 ISSN 1560-8956

проведений аналіз та запропонована нова трактовка другої нормальної форми – адаптованої під сучасні умови побудови баз даних.

В роботі [4] продовжений аналіз нормальних форм та змінений підхід у визначенні та використанні посиленої третьої нормальної форми – так званої нормальної форми Бойса-Кодда (НФБК).

Поточна робота є логічним продовженням попередніх матеріалів – аналіз вищих нормальних форм та їхня адаптація до сучасних підходів до розробки реляційних баз даних.

Четверта нормальна форма (далі, 4НФ) так само як і 2НФ та НФБК пов’язана з поняттям «складеного первинного ключа». Проблема в тому, що як атрибути, що складають цей ключ, у визначенні та демонстрації 4НФ взяті дані, які логічно пов'язані із сутністю (таблицею), а це є породженням надлишковості даних в таблицях.

В роботі [6] в якості навчального експерименту доведена корисність 4НФ. Авторка продемонструвала на прикладах, що навіть тоді, коли досягнуто 3НФ чи НФБК, в базі даних все ще може залишатися надлишковість і аномалії оновлення, якщо існують незалежні багатозначні залежності між атрибутами.

Метою статті є аналіз 4НФ задля використання при проєктуванні схеми бази даних та пропозицій щодо адаптації 4НФ до сучасних умов розробки інформаційних систем з базами даних.

Визначення та трактування 4НФ

4НФ визначається як розширення попередніх нормальних форм, спрямоване на усунення нефункціональних багатозначних залежностей. Це означає, що відношення перебуває в 4НФ, якщо для кожної нефункціональної багатозначної залежності $X \rightarrow\rightarrow Y$ виконується умова, що X є «надключем» [2].

Інакше, 4НФ говорить про те, що жодна багатозначна залежність не призводить до надлишковості даних, якщо тільки вона не базується на «надключі». Це гарантує, що всі багатозначні залежності в схемі відношення є наслідком ключових обмежень, що сприяє зменшенню аномалій вставки, оновлення та видалення.

Для досягнення 4НФ необхідно розділити відношення, які містять нефункціональні багатозначні залежності, на менші відношення, де ці залежності усуваються. Такий підхід дозволяє зберігати дані без надлишковості та забезпечує цілісність бази даних.

Поняття «надключ» або «суперключ» – це сукупність атрибутів, яка однозначно ідентифікує кожен запис у таблиці. Такий ключ складається з потенційних ключів, тобто будь-який такий потенційний ключ може бути обраний як первинний ключ.

В свою чергу, потенційний ключ може бути складеним.

Багатозначні залежності в контексті 4НФ можуть утворюватися виключно при використанні складеного первинного ключа, тобто як первинний ключ взято не сурогатний (штучний) ключ, як наразі практикується, а перелік інших атрибутів (стовпців) таблиці.

Розглянемо приклад простої реляційної таблиці, створеної в реляційній системі управління базами даних MS SQL Server (рис. 1 та рис. 2).

```
CREATE TABLE [dbo].[Documents](
    [ID] [bigint] IDENTITY(1,1) NOT NULL,
    [DocumentType] [tinyint] NOT NULL,
    [DocumentNumber] [nvarchar](100) NOT NULL,
    [DocumentDate] [datetime] NULL
 CONSTRAINT [PK_DocumentsID] PRIMARY KEY CLUSTERED
([ID] ASC) WITH (PAD_INDEX = OFF,
        STATISTICS_NORECOMPUTE = OFF,
        IGNORE_DUP_KEY = OFF,
        ALLOW_ROW_LOCKS = ON,
        ALLOW_PAGE_LOCKS = ON,
        OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Рисунок 1. Скрипт для створення таблиці «Документи»

Column Name	Data Type	Allow Nulls
ID	bigint	☐
DocumentType	tinyint	☐
DocumentNumber	nvarchar(100)	☐
DocumentDate	datetime	☑
		☐

Рисунок 2. Структура створеної таблиці «Документи»

Таблиця на рис. 2 є стандартом проєктування реляційних моделей даних зі стовпцем ID, який являє собою IDENTITY – автоматично генерований стовпчик (лічильник) для первинного ключа. Це є «надключ», тож, використовуючи такий підхід, розробник одразу дотримується 2НФ, НФБК та 4НФ завдяки тому, що:

– первинний ключ суцільний – складається лише з одного атрибуту;

– первинний ключ не є логічно пов’язаним із сутністю (таблицею) «Documents».

Це означає, що значення в цьому стовпчику не є бізнес-атрибутом. Значення цього стовпчика є окремим лічильником, що однозначно ідентифікує рядок таблиці в реляційній базі дані – тобто це штучне поле, яке використовується самою СУБД для підтримки цілісності реляційної бази даних [5].

 ISSN 1560-8956

На рис. 3 наведено скрипт таблиці (рис. 4), яка не знаходиться у 4НФ:

```
CREATE TABLE [dbo].[DocumentsNot4NF](
    [DocumentCode] [nvarchar](100) NOT NULL,
    [DocumentType] [tinyint] NOT NULL,
    [FromDepartment] [nvarchar](100) NOT NULL,
    [DocumentNumber] [nvarchar](100) NOT NULL,
    [DocumentDate] [datetime] NULL,
    CONSTRAINT PK_DocumentsNot4NF PRIMARY KEY (DocumentCode, DocumentType, FromDepartment)
)
```

Рисунок 3. Скрипт для створення таблиці, що не знаходиться у 4НФ

DocumentCode	DocumentType	FromDepartment	DocumentNumber	DocumentDate
Code1	1	Department1	Num1	2025-05-12 14:58:54.640
Code1	2	Department3	Num3	2025-05-12 14:58:54.640
Code2	1	Department2	Num2	2025-05-12 14:58:54.640
Code2	3	Department3	Num4	2025-05-12 14:58:54.640
Code3	3	Department4	Num6	2025-05-12 14:58:54.640
Code3	3	Department5	Num5	2025-05-12 14:58:54.640
Code3	4	Department4	Num7	2025-05-12 14:58:54.640
Code3	5	Department5	Num8	2025-05-12 14:58:54.640

Рисунок 4. Таблиця з тестовими даними

Нехай:

DocumentCode = **A**,

DocumentType = **B**,

FromDepartment = **C**,

DocumentsNot4NF = **R**

Тоді, $R\{A, B, C\}$ – змінна відношення, що описує дані таблиці (див. рис. 4).

Таблиця на рис. 4 не знаходиться в 4НФ через існування багатозначної залежності всередині складеного первинного ключа змінної відношення $R\{A, B, C\}$:

$A \twoheadrightarrow B$ – документи з певними кодами можуть мати лише типи документів з певної множини значень B, де $B = \{1, 2, 3, 4, 5\}$;

$A \twoheadrightarrow C$ – документи з певними кодами можуть приходити лише з департаментів з певної множини C, де $C = \{Department1, Department2, Department3, Department4, Department5\}$.

При цьому B та C не залежать одне від одного. Таким чином, утворилась така залежність:

$$A \twoheadrightarrow B \mid C \quad (1)$$

Це є порушенням 4НФ відповідно до [2].

Як бачимо, таблиця на рис. 3 та рис. 4 докорінно відрізняється від загальноприйнятих принципів побудови реляційних таблиць в сучасних реляційних СУБД –

головним чином наявністю складеного первинного ключа, елементи якого являють собою атрибути, логічно пов'язані з таблицею.

Унікальні та первинні ключі

Приведемо таблицю на рис. 4 до сучасного виду зі збереженням «надключа», застосувавши такі обмеження цілісності, як унікальний ключ та суцільний первинний ключ.

Рисунок 5 демонструє ER модель (Entity Relational model), яка відтворює зберігання даних з надлишковістю.

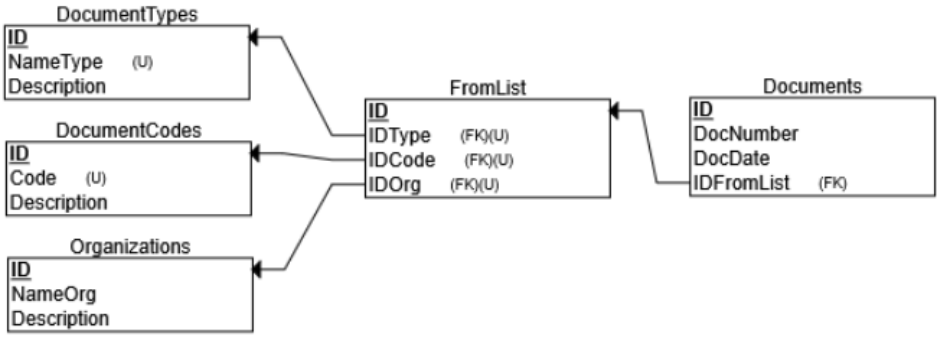

Рисунок 5. ER модель предметної області для таблиці DocumentsNot4NF

Таке представлення таблиць спрощує читання схеми бази даних, нівелює можливість недотримання 2НФ, НФБК та 4НФ та ставить під питання існування Четвертої нормальної форми в такому вигляді, як її було представлено більше 30 років тому.

При розробці бази даних для певного конкретного застосування, розробник поступово виконує етапи інфологічного, даталогічного та фізичного проєктування.

На першому етапі інфологічного проєктування використовують правила атрибутів. За умов виконання цих правил в сутностях з’являються «ідентифікатори» (майбутні первинні суцільні ключі) та вказівні атрибути (майбутні унікальні ключі таблиць).

Якщо первинний ключ є особливим, єдиним полем, яке однозначно ідентифікує кожен рядок в таблиці, то унікальний ключ можна накласти на один або декілька полів таблиці. Тобто те, що 30 років тому підлягало підкресленню і виділенню як окремий «надключ» або «суперключ», або, навіть, їх множина, на сьогодні може бути додане в таблиці як прості атрибути з накладанням унікальності.

На рис. 5 продемонстровано два сучасних підходи до проєктування баз даних:

– використання зв'язків M:M для запобігання надлишковості у сенсі дублювання бізнес-атрибутів (типів документів, кодів документів, департаментів). Всі бізнес-атрибути зберігаються в єдиному варіанті у відповідних таблицях (DocumentTypes; DocumentCodes; Organizations), а дублювання відбувається в асоціативній таблиці FromList, де множаться лише посилання на дані (зовнішні ключі), а не самі дані;

– використання унікальних ключів для відтворення у застосунку саме тієї комбінації даних, яка потрібна адміністратору баз даних.

Таким чином, залежність (1) втрачає свою актуальність. Додавання або оновлення нового елемента у відповідні множини A, B, C призводить до зміни (створення) нового елемента в одній таблиці, а всі посилання залишаються незмінні. При спробі видалення елемента з однієї з множин реляційна СУБД, за наявності вторинного ключа в асоціативній таблиці FormList, виведе помилку, яка свідчитиме про існування дочірнього елементу.

Аномалії видалення, вставки та оновлення в такому випадку будуть неможливі, що і є на меті дотримання 4НФ.

Надлишковість даних при використанні складених первинних ключів

Повертаючись до рис. 3 і рис. 4 та методів приведення до 4НФ з використанням положень оригінальних джерел [2] та [6], бачимо, що як первинні ключі автори використовують бізнес-атрибути (ПІБ вчителя, назва курсу тощо), що вже несе приховану надлишковість в разі, якщо з’являться таблиці, що посилаються на поточну таблицю.

Розглянемо приклад таблиць після декомпозиції з відомого джерела [2]. Автор привів до 4НФ таблицю (або змінну відношення) CTX (Course Teacher teXt), утворивши дві таблиці CT та CX:

В наведених таблицях всі стовпці є частинами складених первинних ключів. Таблиці не поєднані одна з одною і є відірваними від реалій частинами деякої схеми даних.

Для наочності, створена реляційна модель даних, де додано ще таблиці, які посилаються на відношення на рис. 6:

CT

COURSE	TEACHER
Physics	Prof. Green
Physics	Prof. Brown
Math	Prof. Green

CX

COURSE	TEXT
Physics	Basic Mechanics
Physics	Principles of Optics
Math	Basic Mechanics
Math	Vector Analysis
Math	Trigonometry

Рисунок 6. Декомпозиція таблиці CTX [2]

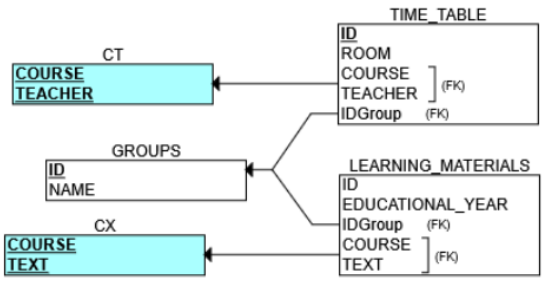

Рисунок 7. Реляційна модель даних зі складеними первинними ключами (зі значеннями атрибутів складеного ключа – бізнес-атрибутів)

Аналіз отриманої схеми даних дозволяє побачити, що бізнес-атрибути починають з’являтися в багатьох таблицях, що призводить до дублювання та очевидної надлишковості. При потребі оновити такі дані, необхідно оновити значення первинного ключа (назва курсу, ПІБ викладача тощо), що є хибною практикою і не імплементується в сучасному світі розробки застосувань, бо первинний ключ створюється лише один раз і використовується до моменту видалення рядка з таблиці.

Віднесемо використання корисних даних до старих часів – на початкову практику, та створимо сучасний вид складеного первинного ключа. Тобто будемо використовувати не бізнес-атрибути, а зовнішні ключі як складові первинного ключа (рис. 8).

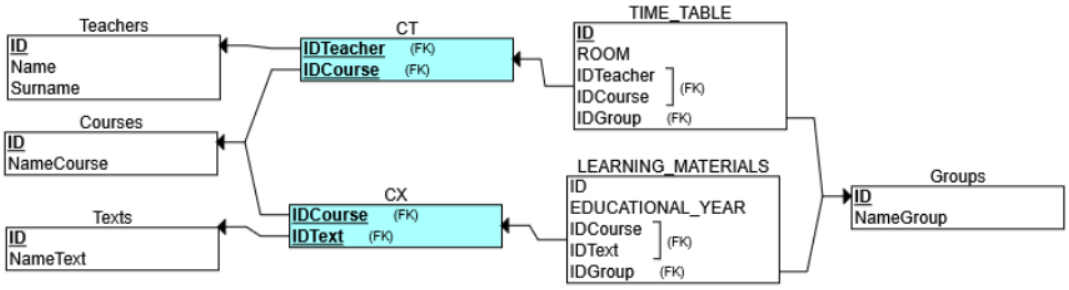

Рисунок 8. Реляційна модель даних зі складеними первинними ключами (як значення атрибутів складеного ключа використані вторинні ключі)

На рис. 8 зображений складений первинний ключ в структурі таблиць CT, CX та асоціативних таблиць TIME_TABLE та LEARNING_MATERIALS. Дана реляційна модель є такою, що всі змінні відношення (таблиці) знаходяться в усіх чотирьох нормальних формах з урахуванням нормальної форми Бойса-Кодда. Але на практиці складені ключі не застосовуються в тих таблицях, на які в подальшому плануються посилання. В нашому випадку, така надлишковість існує для двох таблиць:

CT – на таблицю посилається TIME_TABLE, утворюючи два поля (IDTeacher та IDCourse), які в подальшому в запитах повинні обов’язково фігурувати в умові поєднання;

CX – на таблицю посилається LEARNING_MATERIALS, утворюючи два поля (IDText та IDCourse) з аналогічним поясненням.

З області підвищення ефективності виконання запитів [7] в промислових таблицях відомо, що при поєднанні таблиць конструкцією JOIN, однією з обов’язкових умов є скорочення кількості поєднань між таблицями, а в даному випадку кількість поєднань зростає.

CT та CX введені для підкреслення багатозначної залежності – викладачу жорстко відповідають курси з певної множини (CT), а курсам жорстко відповідають певні підручники з множини наявних (CX). Ввівши додаткові таблиці, можна створити

 ISSN 1560-8956

певні пари $C1 \dots i \to T1 \dots k$ та $X1 \dots j \to T1 \dots m$ і в асоціативних таблицях використовувати лише посилання на пари СТ та СХ.

Таким чином, створивши штучну надлишковість даних з дотриманням нормальних форм, на рівні бази даних можна відтворити багатозначні залежності будь-якого рівня складності.

Адаптація визначення 4НФ до сучасних умов

Визначення надлишковості обумовлює те, що надлишковість даних виникає тоді, коли одне й те саме значення даних зберігається у двох або більшій кількості місць [2].

Рисунок 7 чітко демонструє наявну надлишковість даних – курси зберігаються як бізнес-атрибути в чотирьох таблицях (СТ, СХ, TIME_TABLE, LEARNING_MATERIALS), а вчителі та назви підручників – у двох таблицях. Тому визначення 4НФ необхідно адаптувати до сучасних умов, де дублювання даних ведеться через зовнішні ключі, що цілком припустимо, адже бізнес-атрибути можуть змінювати своє значення, причому посилання на них завжди буде сталим (див. рис. 8).

Наразі визначення 4НФ звучить так: Відношення R перебуває в четвертій нормальній формі, якщо для кожної нетривіальної багатозначної залежності $X \twoheadrightarrow Y$ у R, X є суперключем [8].

Адаптоване визначення 4НФ з використанням оригінальної трактовки: Відношення R перебуває в четвертій нормальній формі, якщо для кожної нетривіальної багатозначної залежності $X \twoheadrightarrow Y$ для R множина атрибутів X є «суперключем», причому кожен потенційний ключ в «суперключі» є штучним (сурогатним) утворенням ідентифікатора — унікального, непорожнього та незалежного від змісту сутності.

Таке визначення дає чітке уявлення про те, що потенційні ключі повинні бути штучними, логічно не пов’язаними із сутністю і, по факту, бути зовнішніми або вторинними ключами таблиці. Таким чином, надлишковість даних буде знівельована, а багатозначні залежності різного вид складності – імплементовані із застосуванням декомпозиції.

Висновки

В статті наведене сучасне трактування четвертої нормальної форми з повним прикладом створення реляційної схеми бази даних. Проведений аналіз запропонованих до використання «суперключів», потенційних ключів та складених первинних ключів.

Проведений аналіз на надлишковість та зроблені відповідні висновки.

4НФ в сучасних умовах практично не зустрічається завдяки процесу інфологічного моделювання, де, дотримуючись правил атрибутів (особливо, першого – наявність ідентифікатора в кожній сутності), 4НФ завжди буде дотримана.

Незважаючи на це, все ж існують випадки порушення 4НФ, тому для сучасних умов проєктування реляційних баз даних в статті запропоноване адаптоване визначення 4НФ, коли за складові «надключі» взяті штучні (сурогатні) атрибути, які не призведуть до виникнення надлишковості даних. Проблема використання складених первинних ключів описана в попередніх статтях [3], [4], а в [4] запропоноване їх використання лише задля підвищення ефективності виконання запитів.

Список використаних джерел

1. Codd E.F. A Relational Model of Data for Large Shared Data Banks / E.F. Codd // Communications of the ACM. – 1970. – Vol. 13, No. 6. – P. 377–387.

2. Date C.J. An Introduction to Database Systems / C.J. Date. – 8th ed. – Boston : Pearson/Addison Wesley, 2003. – 1024 p.

3. Rolik O., Amons O., Ulianytska K., Kolesnik V. (2021) Modernization of the Second Normal Form and Boyce-Codd Normal Form for Relational Theory. In book: Hu Z., Petoukhov S., Dychka I., He M. (eds) Advances in Computer Science for Engineering and Education III. ICCSEEA 2020. Advances in Intelligent Systems and Computing, vol 1247. Springer, 2021. – pp. 296-305. DOI: 10.1007/978-3-030-55506-1_27

4. Rolik, O., Ulianytska, K., Khmeliuk, M., Khmeliuk, V., Kolomiiets, U. Increase efficiency of relational databases using instruments of second normal form / 2021 IEEE 3rd International Conference on Advanced Trends in Information Theory, ATIT 2021 – Proceedings, 2021, pp. 221–225.

5. Elmasri R., Navathe S.B. Fundamentals of Database Systems / R. Elmasri, S.B. Navathe. – 7th ed. – Boston: Pearson, 2016. – 1152 p.

6. Wu, Margaret S. (March 1992). "The Practical Need for Fourth Normal Form". ACM SIGCSE Bulletin. 24 (1): 19–23. doi:10.1145/135250.134515

7. Garcia-Molina H., Ullman J.D., Widom J. Database Systems: The Complete Book / H. Garcia-Molina, J.D. Ullman, J. Widom. – 2nd ed. – Pearson, 2008. – 1344 p.

8. Fagin R. Multivalued Dependencies and a New Normal Form / R. Fagin // ACM Transactions on Database Systems. – 1977. – Vol. 2, No. 3. – P. 262–278. – doi:10.1145/320557.320571