

<https://doi.org/10.15407/csc.2024.03.010>  
УДК 681.3

**О.І. ІВАНЕШКІН**, доктор техн. наук, провідний наук. співробітник,  
Міжнародний науково-навчальний центр інформаційних технологій і систем НАН та МОН України,  
просп. Академіка Глушкова, 40, Київ, Україна, 03187,  
ORCID: <https://orcid.org/0009-0006-6800-2944>,  
[al.ivaneshkin@gmail.com](mailto:al.ivaneshkin@gmail.com)

## **ЩОДО ОДНОГО ПІДХОДУ ДО ВИРІШЕННЯ ПИТАННЯ АВТОМАТИЗОВАНОГО БУДУВАННЯ СХЕМИ ПОСИЛАНЬ РЕЛЯЦІЙНИХ БАЗ ДАНИХ**

Запропоновано підхід, наведено результат його програмної реалізації у вигляді “пілотного” варіанта комплексу програмних засобів автоматизованого будування схем посилань реляційних баз даних. Складовими підходу є способи, методи та алгоритми, які раніше розроблено і програмно реалізовано у новій створеній універсальній *software-технології* для роботи з (поки що) неорієнтованими графами виду “змішаного лісу”. Застосування комплексу дозволяє багаторазово скоротити час будування схем посилань і дає можливість роботи з таблицями даних, загальна кількість полів яких є обмеженою лише спроможністю використовуваного комп’ютера. Забезпечуючи більше, ніж 90 % економію технічних ресурсів, використання комплексу робить “формальною” участь користувача в процесі будування схем.

**Ключові слова:** реляційні бази даних, таблиці даних, схема посилань, неорієнтовані графи-дерева, комплекс програмних засобів, інформаційна *software-технологія*.

### **Вступ**

Існуюча тенденція безперервного інтегрування науково-практичних сфер діяльності людини, сприяє постійному зростанню складності задач, успішне вирішення яких стає можливим лише завдяки використанню значних обсягів найрізноманітнішої інформації, як правило, розподіленої у множині файлів бази даних [1]. В цьому випадку, потреба знати місця розташування і особливості взаємодії її окремих складових набуває принципового значення при організації максимально зручних і оперативних шляхів доступу до неї.

В деяких випадках, задля задоволення таких потреб, можна скористатися послугами існуючих продуктів (*Microsoft Access, Microsoft SQL Server, MySQL, FileMaker Pro, Oracle Database, dBASE*, тощо). Іноді, з допомогою спеціалізованої мови *SQL*, або її діалектів, зручніше створити “*handmade*”, який відповідатиме сформульованим вимогам.

В разі бажання, або потреби користувача знати місця перебування використовуваної в роботі інформації, розуміти логіку роботи механізму доступу і оцінювати ефективність шляхів її отримання з метою повсякчасного оперативного контролю її поточної і перспек-

Cite: Іванешкін О.І. Щодо одного підходу до вирішення питання автоматизованого будування схем посилань реляційних баз даних. *Control Systems and Computers*, 2024, 3, 10-32. <https://doi.org/10.15407/csc.2024.03.010>

© Видавець ВД «Академперіодика» НАН України, 2024. Стаття опублікована на умовах відкритого доступу за ліцензією CC BY-NC-ND (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

тивної номенклатури та кількості, можуть стати в нагоді власні розробки, здатні надавати більш детальні відомості, ніж відомі засоби.

Як засвідчує досвід, отримання інтегрального і ефективного вирішення зазначеної сукупності тісно пов'язаних питань стає можливим лише за наявності суто спеціалізованого програмного засобу автоматизованого будування схеми посилань (SP), що має вигляд багатofункціонального, оперативного, простого в опануванні і зручного у використанні користувачького середовища. Наділене здатністю у будь-який час і з мінімальною участю користувача візуалізувати на екрані монітора поточну структуру створеної БД (DB) з потрібним ступенем наочності, деталізації і простоти сприйняття, таке середовище має стати засобом оперативного аналізу і оцінки шляхів можливої модифікації самої бази в напрямку створення більш ефективного у практичному значенні варіанта її структури.

## Постановка проблеми

З точки зору сприйняття кінцевого результату, найбільш зручним засобом будування і відображення на екрані таких схем є апарат теорії графів. Іти відомими з теорії графів шляхами, використовуючи різного роду матриці або списки, щоразу вишукуючи серед них найбільш ефективний варіант формалізованого подання самих графів є малоприємним заняттям зі значними часовими і ресурсними витратами. Навіть за наявності широко вживаних стандартизованих мов опису структури, спеціалізованих програм для будування і візуалізації графів тощо, лівова частка загальних зусиль при будуванні схем перепадатиме на “ручну” роботу, обсяг якої істотно зростатиме при збільшенні кількості таблиць даних (ТД), які мають бути обов'язково враховані — це потребує значної уваги, акуратності і наполегливості.

Аналіз підходу, способів, методів, алгоритмів тощо, розроблених і покладених у основу викладеної у роботах [2—5] і програмно реалізованої універсальної інформаційної *software*-технології для роботи з (поки що) неорієнтованими

ними графами виду “змішаного лісу” [4], виявив можливість їхнього застосування у процесі будування схем посилань реляційних баз даних (БД). При цьому, участь користувача стає “формальною”, бо обмежуватиметься лише конкретизацією імені “базової” таблиці даних з якої починається будування і вибором (шляхом маркування) у загальному списку прописаних в її хедері імен тих полів, що стануть “базовими” ключами, потрібними комплексу програмних засобів (КПЗ) для активізації процесу розгалуження, який завершиться виводом на екран монітора створеної ним схеми.

Наведений у статті матеріал, значною мірою є “пілотним” і експериментальним, призначеним для виявлення можливостей і кордонів практичного втілення результатів зазначених робіт щодо створення базових елементів інтегрованого користувачького середовища, яке має вигляд КПЗ автоматизованого будування схем посилань реляційних баз даних. Робиться спроба застосування результатів цих робіт, для створення ефективного засобу низьковитратного розв'язання сукупності задач, які вирішують питання будування, всілякої модифікації, оперативного візуального контролю, перегляду і оцінки структур, створених користувачем БД реляційного типу задля створення цілісної, максимально деталізованої картини взаємозв'язку включених на поточний момент часу до їхнього складу ТД і окремих типів інформації, які в них містяться.

## Мета досліджень

Мету, що визначила напрям проведення робіт, можна сформулювати наступним чином. Дослідження і практична апробація наявних можливостей підходів, методів, алгоритмів і програмних модулів, розроблених при створенні інформаційної *software*-технології для роботи з неорієнтованими графами типу “змішаний ліс” для оцінки ефективності і доцільності їхнього використання у комплексі інтегрованих програмних засобів автоматизованого будування схем посилань реляційних БД з мінімальною участю користувача у комплексі.

Загальний вигляд інтерфейсу користувача, опис розроблених і включених до його складу операцій, а також деякі з отриманих в разі використання КПЗ результати, наведено по ходу викладання матеріалу.

### Програмні засоби формування схем посилок реляційного типу баз даних

Зазначені у статті програмні засоби розроблені на алгоритмічній мові високого рівня VISUAL C++ 6.0 і забезпечують:

- автоматизоване “з нуля” будівництво схем посилок, призначених для ефективного і оперативного контролю структурних і функціональних особливостей створених БД реляційного типу з метою їхнього максимально ефективного практичного використання;
- найбільш зручне для сприйняття, графічне (у вигляді графа-дерева) подання особливостей взаємного зв’язку окремих, включених до складу БД на поточний момент часу ТД, а також конкретних даних, які містяться у цих таблицях і є ідентифікованими полями, зазначеними у хедерах їхніх *dbf*-файлів;
- оперативні модифікації раніше сформованих варіантів схем посилок (дані яких, зберігаються у бінарних, вигляду *FileName.ref* файлах), в разі додавання до складу БД нових ТД, чи їхніх полів, або вибіркового тимчасового чи остаточного вилучення зі складу БД не використовуваних ТД;
- одночасний перегляд на екрані згрупованих за певною ознакою сукупностей схем посилок

$$SP^i \left( SP^i = \bigcup_{j=1}^J SP_j^i; SP_j^i \in SP; \bigcup_{i=1}^I \bigcup_{j=1}^J SP_j^i \subseteq SP; \right)$$

для наявної у робочому каталозі сукупності баз даних

$$BD_j^i \left( \bigcup_{i=1}^I \bigcup_{j=1}^J BD_j^i \subseteq BD; BD^i = \bigcup_{j=1}^J BD_j^i; BD \neq \emptyset; \right);$$

- мінімальну участь користувача в усіх, розроблених і включених на поточний час до

складу КПЗ операціях модифікації структур БД і відповідних схем посилок.

Засобом математичної формалізації окремих схем посилок, їхнього візуального зображення на екрані монітора і різного роду модифікацій, обрано апарат теорії графів. Зокрема, його частина, що безпосередньо пов’язана з графами-деревами  $G(A, B)$ , де  $A$  — множина вершин ( $\dim(A) = N (2 < N)$ ),  $B$  — множина ребер.

Графічним зображенням (інтерпретацією) схеми є граф-дерево, чия топологічна структура відображує особливості (механізм, специфіку) взаємозв’язку даних, що містяться у *dbf*-форматах файлах (ТД) робочого каталогу на ім’я “*Data\_Cross*” і тимчасово або постійно задіяні у складі частково, або повністю сформованої реляційного типу БД.

Для візуального подання структур схем, використовуємо добре відомий з *OS Windows*, *Norton* тощо, підхід. Його суть полягає у зображенні вершин дерева прямокутниками, які розміщені в цілочисельних точках перетину горизонтальної ( $X$ ) і вертикальної ( $Y$ ) віртуальних осей, що виходять з лівого верхнього кута екрана монітора. Це дає змогу легко зрозуміти всі особливості співпорядкування вершин дерева одна з одною.

Перед описом програмних модулів, що реалізують окремі операції у створеному КПЗ, введемо ряд визначень, які використовуватимемо при викладанні матеріалу статті.

**Головне репрезентативне вікно.** Інформаційне вікно інтерфейсу користувача, призначене для подання на екрані вперше створених “з нуля” схем посилок, або раніше створених і нині обраних для перегляду, аналізу, модифікації, чи використання.

**Робоча БД.** Сформована користувачем БД, структура якої присутня на екрані і містить повну (на цей час) сукупність функціонально пов’язаних *dbf*-формату ТД, для безпосереднього перегляду, модифікації, або практичного використання.

**ROOT-вершина схеми посилок.** Єдина “штучно” введена вершина схеми, яка не несе інформаційного навантаження. Слугуючи коренем графа-дерева, вона надає схемам ціліс-

ності «зовнішнього» вигляду як єдиних інтегрованих сукупностей взаємно пов'язаних елементів, спрощує зрозуміння логіки їхньої будови і функціонування, дає змогу використання розробленого типу структуризації даних схем в разі їхнього збереження в бінарному *FileName.ref* файлі (далі, *ref*-файл) після створення, чи модифікації.

**Однорівневі вершини.** Сукупність  $\{A_i\}_i \in A$  вершин конкретного  $j$ -го  $X$ -рівня дерева, загальним суміжним елементом яких є єдина вершина попереднього вищого  $(j - 1)$ -го  $X$ -рівня, що є коренем гілки, всі  $\{A_i\}_i$  якої є одночасно її прямими нащадками.

**Моносхема посилань.** Одиночна (єдина) схема посилань, що подана  $N$ -вершинним ( $3 < N$ ) графом-деревом, дані якого записано в окремому бінарному *ref*-формату файлі.

**Полісхема посилань** Схема, що інтегрує в собі  $D$  ( $1 < D$ ) моносхем, дані яких записано у загальному *ref*-файлі. Такі схеми забезпечують зберігання і одночасне графічне подання на екрані кількох функціонально (не)пов'язаних моносхем. Вони відображують особливості організації даних в кожній з згрупованих за будь-якою ознакою БД задля їхнього одночасного оперативного перегляду. У головному репрезентативному вікні моносхеми розташовані одна під одною, розділені проміжком і дозволяють перегляд в *scroll*-режимі.

**$Y$ -рівень ключа моносхеми посилань.** Порядковий номер  $n$   $Y$ -рівня  $N$ -вершинного графа-дерева схеми посилань ( $n = 0, 1, 2, \dots, N$ ), на якому розташовано відповідний ключ. Відлік (зверху до низу)  $Y$ -рівнів моносхеми починається з значенням "0", яке припишемо особливій *ROOT*-вершині, що не є ключем. При кількості ключів (вершин графа)  $N$ , що входять в структуру графа, останній  $Y$ -рівень дерева матиме значення  $N$ , а загальна кількість  $Y$ -рівнів моносхеми дорівнюватиме  $(N + 1)$ .

**$X$ -рівень ключа моносхеми посилань.** Порядковий номер  $m$  займаного ключем горизонтального  $X$ -рівня ( $m = 0, 1, 2, \dots, M$ ) відносно "0"-го  $X$ -рівня кореня моносхеми.

З кожним ключем пов'яжемо рядок структурованих параметрів, який розташуємо на

тому ж  $Y$ -рівні після прямокутника, що зображує вершину. Компонентами рядка є: порядковий номер поля пов'язаної з ключем ("власної", або "материнської") ТД, прописаний у прямокутнику вершини; імена ключа, поля і його материнської ТД (позначені "*KEY\*\**" і "ключ з *DATA\*\**");  $Y$  і  $X$ -рівні, займані ключем у схемі посилань ("*Y\_Lev = \*\**" і "*X\_Lev = \*\**");  $Y$ -відстань ("*Y\_Dist\_ToROOTBr = \*\**") між  $Y$ -рівнем ключа і  $Y$ -рівнем найближчого до нього зверху кореня гілки дерева, якій належить ключ; додаткова інформація;

Задля зручності подання схем на екрані монітора і використання мов програмування високого рівня при створенні КПЗ, *ROOT*-вершина графа-дерева будь-якої схеми посилань завжди матиме нульові значення координат  $Y$  і  $X$  (0; 0) і статус "віртуальної" вершини. Візуальною ознакою цього буде присутність символу "#" на полі її прямокутника.

**Ключ схеми посилань.** Вершина дерева схеми з координатами  $(n; m)$ , яка має вигляд прямокутника з наявним всередині ім'ям "*NAME*", що слугує ідентифікатором імені "*NAME\_0*" одного з полів, прописаних в хедері (заголовку) материнської ТД цього ключа. Цю обставину зручно інтерпретувати як породження ключа з ім'ям "*NAME*" полем, що має ім'я "*NAME\_0*" у його материнській ТД.

Покладемо ідентичними імена ключа і поля, що його породило. Це спростить і рішення задачі візуального встановлення їхнього взаємного зв'язку "один з одним", і розуміння суті запропонованого підходу. Всі наявні в репрезентативному вікні ключі (за винятком тих, що породжено нульовими номерами полів відповідних ТД) і імена полів, які породили такі ключі, зручно визначати як "активні". В цьому разі імена полів ТД, що входять в робочу БД, але не породили ключів, які присутні в робочій схемі посилань і отже не представлені на екрані ключами, має сенс назвати "пасивними". Кількість і склад ключів схеми посилань може бути змінено користувачем згідно ситуації, що виникла. Для цього у КПЗ передбачено відповідні операції. Зокрема, причиною додавання в схему активних ключів є створення і долучен-

ня до складу робочої БД нових ТД, чи полів вже наявних у ТД. Причиною вилучення частини ключів є вилучення зі складу робочої БД таблиць (або їхніх окремих полів), подальше використання даних яких тимчасово, або остаточно визнане недоцільним. Виняток становлять ключі з координатами  $(m, n)$ , які породжено “0”-ми номерами полів їхніх власних ТД (на рисунках, це листя графа-дерева). Особливістю таких ключів є брак у них нащадків — суміжних вершин чергових  $(n + 1)$ -х  $X$ -рівнів. В загальному випадку, такі ключі можуть бути породжені не тільки “0”-ми, а будь-якими номерами полів.

Якщо імена зазначених вище ключів і полів зробити різними, все наведене нижче залишиться в силі, в разі зберігання імені кожного поля, що породило відповідний йому ключ, у частині пов’язаної з цим ключем строки, що відповідає додатковому параметру.

**Номер поля ключа схеми посилань.** Порядковий номер  $k$  ( $k = 0, 1, 2, \dots, K$ ) імені поля, яке “породило” цей ключ і прописане в хедері файлу його материнської ТД. Цей номер є ідентичним порядковому номеру поля в структурі запису даних МТД ключа і виведений у прямокутнику, який означає конкретну (окрім *ROOT*) вершину дерева.

**Материнська (власна) таблиця даних (МТД, ВТД) ключа схеми посилань.** *dbf*-формат файлу ТД, в хедері якого зазначено ім’я поля будь-якого типу даних, що породило активний ключ схеми з ідентичним ім’ям, забезпечуючи їхню взаємно однозначну відповідність. Оскільки корінь дерева не є ключем, то візуальна наявність ключа в схемі посилань вказує на наявність серед сукупності ТД робочої БД такої ТД, ім’я нульового номера поля якої ідентично імені цього ключа. Вище зазначалось, що число всередині кожного прямокутника схеми є порядковим номером поля, яке породило пов’язаний з цим номером ключ і прописано в хедері його власної ТД. Розмістивши стрілку маніпулятора всередині прямокутника і натиснувши його праву кнопку, можна отримати інформацію про всю сукупність імен полів власної ТД конкретного ключа.

Серед цих імен буде і ім’я поля ключа, маркування якого сприяло отриманню цієї інформації. Ознакою вибору ключа є зміна кольору поля його прямокутника на зелений. Список імен полів власної ТД вибраного ключа буде виведено в крайньому правому інформаційному вікні з заголовком “Імена полів ТД \*\*\*\*” де (\*\*\*\*) — конкретизоване ім’я цієї ТД. Отримання такої інформації для ключів, породжених нульовими номерами полів їхніх власних ТД, в програмі не передбачено. Її можна отримати, маркуючи будь-який з ключів, що має ту саму власну ТД, ім’я якої прописано в пов’язаному з ключем інформаційному рядку і має вигляд (“ключ з ТД = \*\*\*\*”).

**Однорівневі, ( $j$ -го  $X$ -рівня) ключі схеми посилань.** Сукупність  $Q$  ( $1 \leq Q$ ) присутніх у головному репрезентативному вікні ключів з іменами  $\{KEY_q^j\}_{q=1}^Q$ , які в схемі посилань є  $Q$  вершинами дерева, розміщеними винятково на  $j$ -му  $X$ -рівні. Всі вони мають за суміжний один і той же ключ вищого  $(j - 1)$ -го  $X$ -рівня і є коренями породжених ними гілок, або висячими вершинами. Важливим фактом є те, що всі їхні імена  $\{KEY_q^j\}_{q=1}^Q$  породжені іменами полів єдиної ТД, яка одночасно для всіх цих ключів є материнською.

**Базові ключі моносхеми посилань.** Всі присутні в моносхемі посилань ключі 1-го  $X$ -рівня, які породжені вибраними користувачем із загального списку іменами полів їхньої материнської (надалі, “базової”) ТД і слугують базою (основою) для організації процесу автоматизованого будівництва КПЗ нової повноцінної схеми посилань. Використовуючи ці ключі та принцип розгалуження, КПЗ без будь-якої участі користувача, конкретизуючи і вибудовуючи всі зв’язки між даними, які на поточний момент мають входити до БД, генерує функціонально повну (на поточний момент часу) схему у вигляді графа-дерева.

Надалі, для зручності викладання матеріалу, всі ключі схеми посилань, які не є базовими, називатимемо “вторинними” ключами.

**Базовий варіант схеми посилань (структури БД).** Початково сформована КПЗ схема

посилань, структура якої містить лише віртуальну *ROOT*-вершину і базові ключі. Слугує основою для будування за допомогою відповідної операції повноцінної схеми посилань, що відображує зв'язок усіх компонентів робочої БД і даних, які в них містяться.

**Шлях на схемі посилань.** За аналогією з теорією графів — це впорядкована (в КПЗ виділена зеленим кольором) сукупність вершин  $\{A_r\}_{r=1}^R \in A$  і ребер  $\{B_d\}_{d=1}^D \in B$  виведеної на екран схеми посилань, яка визначає і візуалізує доступ користувача до тих даних створеної БД, які його цікавлять. Шлях автоматично будує КПЗ, використовуючи включену до свого складу операцію і отримавши на це команду користувача. Кінцевими вершинами шляху є будь-який вказаний користувачем вторинний ключ і базовий ключ 1-го *X*-рівня схеми посилань.

Створення БД і пов'язаної з нею схеми посилань КПЗ виконуватиме одночасно, тому під словами “будування схеми”, будемо розуміти і будування відповідної БД. І навпаки.

Детальна інформація по кожній розробленій і включеній до складу КПЗ операції, автоматично виводиться у вікні, що має заголовок “Вікно виведення інформації загального типу, а також по кожній, передбаченій програмою операції” після натиснення користувачем відповідної кнопки меню, розміщеної у верхній частині екрана. У цьому ж вікні, незалежно від виконуваної в конкретний момент операції, користувач може отримати інформацію з будь-якої з них, натискаючи кнопку, яка міститься в нижній лівій частині екрану і має напис “Отримати інформацію по активній операції”.

Ознакою вибору користувачем операції (її активність) є “крапка” в центрі радіо-кнопки (*RadioButton*) і поява у верхній частині поля головного репрезентативного вікна інтерфейсу інформаційного рядка, що пояснює її суть. Вікно має заголовок “Вікно візуалізації структури створеної (модифікованої) схеми посилань”. Рядок зникає в разі виведення у вікні схеми посилань, яку створив, модифікуватиме, або переглядатиме користувач.

Розглянувши наведений на рис. 1 загальний вигляд інтерфейсу користувача, перейде-

мо до опису можливостей існуючої натепер версії КПЗ. Для зручності зазначимо, що на наведених у статті рисунках, задіяні в конкретній операції радіо-кнопки і кнопки користувацького меню, виділятимемо відповідно наявністю хрестика, або зміною фону тексту.

Кожна операція, яку здатен виконувати КПЗ, потребує своєї сукупності вихідних даних і тільки їхній окремий розгляд може забезпечити належне розуміння логіки роботи програми, що її реалізує і послідовності потрібних для цього дій. Тому початковим станом інтерфейсу є пасивність усіх радіо-кнопок, розташованих у полі прямокутника із заголовком “Види робіт, дозволених до виконання на схемі посилань”.

Для зручності, сукупність усіх наявних у КПЗ операцій розбито на дві групи.

**Група 1.** Створення нової БД і пов'язаної з нею базової схеми посилань.

Група 1 включає наступні операції:

- Вибір з наведеної у *ComboBox* сукупності (заголовок “Базові таблиці даних БД”) всіх імен *dbf*-файлів (ТД) з робочого каталогу імені тієї базової ТД, вибрані поля якої породять базові ключі, потрібні КПЗ задля організації процесу будування схеми посилань створеної реляційного типу БД.

- Маркування у списку крайнього правого інформаційного вікна, що містить імена всіх полів обраної базової ТД цих полів, які, ставши активними, породять базові ключі створюваної схеми.

- Занесення у поле редактора з заголовком “Імя файла *NEW* схеми посилань” ім'я *\*.ref*-файлу даних схеми, яку буде створено.

- Створення базового варіанта БД і супутної їй схеми посилань.

Процес створення нової БД і пов'язаної з нею схеми починається з натискання відповідної радіо-кнопки, після чого КПЗ:

- активізує текстовий редактор з заголовком “Файл *NEW* схеми посилань”, для занесення у його полі імені файлу створюваної схеми посилань. *ref*-розширення до імені цього файлу буде додано КПЗ автоматично;

- сформує список імен усіх наявних в робочому каталозі ТД (*dbf*-формат файлів) і ви-

веде у вікні із заголовком “Базові таблиці даних (ТД)” для вибору серед них базової ТД, чий імена полів породять базові 1-го X-рівня ключі створюваної схеми посилань;

— активізує кнопку “Підготувати дані для виконання операції”. Її натискання призведе до виведення у вікні із заголовком “Імена полів базової ТД” списку імен всіх  $K$  полів (номери 0, 1, 2, ...,  $K - 1$ ) базової ТД для маркування тих, які стануть базовими при організації процесу створення нової схеми посилань вже без участі в цьому користувача.

Якщо до натискання кнопки “Підготувати дані для виконання операції” не було задано ім'я створюваної схеми посилань, або задане співпадає з наявним у робочому каталозі, буде видано попередження, що конкретизує зроблену помилку. До натискання кнопки “Підготувати дані для виконання операції” користувачеві надається можливість вибору іншого типу роботи, яка зникає після активізації цієї кнопки.

У разі виконання всіх умов вибору базової ТД і коректного запису імені створюваного *ref*-файлу схеми посилань, натискання кнопки “Підготувати дані для виконання операції” активізує пару кнопок “Зберегти дані створюваної схеми посилань” і “Скасувати виконання обраної операції” і наповнює список (*ComboBox*) з заголовком “Імена полів базової ТД” іменами всіх полів, прописаних у хедері вибраної базової ТД. Одночасно з цим, доповнене *ref*-розширенням ім'я створюваної схеми буде занесено у вікно текстового редактора з заголовком “Файл *NEW* схеми посилань”.

Маркування сукупності всіх потрібних імен полів базової ТД і натискання першої з активованих кнопок, дає змогу продовжити обрану операцію. Натискання кнопки з написом “Скасувати виконання обраної операції”, перерве виконання операції і поверне програму в режим головного користувацького меню.

Черговим завданням користувача стає вибір з виведеного на екран списку всіх полів базової ТД тих імен, які породять базові ключі створюваної схеми посилань. Вибір (маркуванням) полів здійснюється розміщенням на кожному з них стрілки маніпулятора і натис-

канням його лівої кнопки. Візуально це виглядає як зміна кольору фону відповідних імен. Відмова від маркування (демаркування) раніше виділених імен полів виконується аналогічно. Якщо користувач, не вибравши жодного імені, натисне кнопку “Зберегти дані створюваної схеми посилань”, КПЗ виведе попередження про потребу неодмінно це зробити.

Дозволено маркувати будь-яку кількість будь-яких імен полів з наведеного списку. Вбудовані у КПЗ засоби контролю залишать для подальшої роботи лише ті, що реально може бути використано в разі організації процесу будівництва нової схеми. Зайві імена, буде проігноровано, щоб запобігти виведенню на екран “сміття”, яке не несе корисного смислового навантаження і лише створює незручності візуального сприйняття.

Активізація кнопки “Зберегти дані створюваної схеми посилань” створить в робочому каталозі новий файл з заданим користувачем *\*.ref*-ім'ям і наповнить його даними початкового варіанту структури створюваної схеми, який матиме вигляд графа-дерева і міститиме лише вершини нульового (*ROOT*-вершина) і першого (базові ключі) X-рівнів.

Дані створюваної “з нуля” базової структури схеми посилань буде записано у бінарному *\*.ref*-файлі. Одночасно, в інформаційному вікні з заголовком “Базові ключі схеми” буде наведено імена всіх раніше промаркованих полів базової ТД, а у інформаційному вікні з заголовком “Вікно візуалізації структури створеної (модифікованої) схеми посилань” виведено базовий варіант структури створюваної БД.

Вивівши базовий варіант структури схеми, КПЗ активізує пару кнопок “Створити нову схему посилань” і “Завершити роботу програми”. В разі натискання другої радіо-кнопки, роботу з базовим варіантом схеми буде завершено і КПЗ перейде до режиму загального користувацького меню, даючи змогу завершити роботу, або вибрати для виконання іншу операцію.

Продовжити роботу з базовими варіантами схем можна у будь-який, зручний для цього час, вивівши потрібний варіант на екран

і скориставшись операцією “Створити нову схему посилань”. Натискання цієї кнопки призведе до того, що:

1. У списку із заголовком “Файли для створення схем посилань” буде виведено імена всіх наявних в робочому каталозі *ref*-файлів, які містять дані винятково базових варіантів схем, що уникли застосування операції будування повноцінних *ref*-схем;

2. Стане активною кнопка “Вивести структуру створюваної схеми посилань”.

Додатковою ознакою активності цієї операції слугує напис “Будування нового/модифікованого варіанту схеми посилань”, який з’явиться у вікні із заголовком “Вікно візуалізації структури робочого варіанту графа”. Цей напис заміниться на “Модифікація старої структури графа”, якщо вибрано раніше створену повноцінну *ref*-схему для виконання на ній будь-якої операції модифікації.

Вибір зі списку імені базового варіанта схеми, для будування її повноцінного варіанту і натискання зазначеної в пункті 2 кнопки, призведе до наступних подій:

- заміни заголовка редактора “Імя файла *NEW* схеми посилань” на “Робочий файл схеми посилань” і виведення в його полі імені *ref*-файлу базового варіанту схеми, яку має бути перетворено на повноцінну;

- виведення в полі редактора із заголовком “Базова ТД схеми посилань” повного (з *dbf*-розширенням) імені базової ТД, чий імена полів породили ключі базового варіанту структури створюваної схеми посилань;

- активізації вікна із заголовком “Базові ключі схеми” і виведення в ньому імен усіх базових ключів обраного варіанту схеми;

- активізації вікна із заголовком “Імена полів базової ТД” з наведеним в ньому списком імен усіх полів ТД, яка є материнською ТД одночасно для усіх базових ключів, наявних і в початковому варіанті схеми, і у вікні із заголовком “Базові ключі схеми”;

- активізації кнопки “Створити нову (*NEW*) схему посилань”;

- активізації кнопки “Скасувати виконання вибраної операції”.

Натискання кнопки “Скасувати виконання вибраної операції” припиняє її виконання, очищує і деактивує всі вікна, списки і кнопки, переводячи програму в режим загального меню користувача.

Натискання кнопки “Створити нову (*NEW*) схему посилань” призводить до автоматичного формування КПЗ повноцінної схеми, що містить усі ТД, які використано у сформованій повноцінній БД. Ознакою завершення операції створення схеми слугує виведення структури графа в головному інформаційному вікні і поява у вікні з заголовком “Всі ключі схеми посилань” списку всіх активних ключів усіх наявних в робочому каталозі ТД, реально задіяних у збудованій *ref*-схемі. Результат виконання операції створення нової повноцінної схеми посилань з базовими ключами 1-го *X*-рівня 1, 4 і 9, показано на рис. 1.

Розміщення стрілки маніпулятора на будь-якій (окрім *ROOT*) вершині графа і маркування її правою кнопкою, дає змогу побачити імена всіх полів материнської ТД ключа, пов’язаного з цією вершиною. Імена полів буде виведено у вікні, що розміщене праворуч і має заголовок “Імена полів ТД \*\*”. Ім’я материнської ТД цих полів буде прописане в заголовку вікна замість знака “\*\*”. Демаркування ключа здійснюється аналогічно. При цьому поле вікна буде очищено, а його заголовок замінено на “Імена полів базової ТД”.

**Група 2.** Включає 4 операції, які є більшою часткою операцій КПЗ і відповідають за модифікацію кількісного і якісного складу компонентів наявних БД і схем посилань, перегляд схем і формування на них шляху, створення схем інтегрованого типу — полісхем.

1. Операція модифікації наявних у робочому каталозі структур БД зміною складу ключів їхніх схем посилань з чотирма реалізованими режимами:

- **режим 1.1.** Вилучення зі схеми посилань неактивних ключів;

- **режим 1.2.** Долучення до схеми посилань нових ключів;

- **режим 1.3.** Долучення до структури обраної (моно)полісхеми аналогічного типу



**БУДУВАННЯ І МОДИФІКАЦІЇ СХЕМИ ПОСИЛАНЬ РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ**

Вибір роботи, дозволяє до виконання на схемі посилаць

Варіанти модифікації схеми ключів схеми посилаць

- Випустили всі ключі для НОВОЇ схеми посилаць
- Випустили тільки нові ключі
- Замінити схему ключів на схемі посилаць
- Замінити тільки активні ключі
- Сформувати тільки нові ключі на схемі посилаць
- Вивести інформацію по виконуваним операціям

Вивести на екран схему, що її модифікують

Цілювати дані до виконання операції

Скасувати виконання вибраної операції

Створити / Модифікувати схему посилаць

Зберігати дані створеної схеми посилаць

Показати шукані від промеркованого ключа до кореня

Реконфігурація схеми після додавання ключів

Ім'я файлу робочої схеми посилаць

aaa.taf

Базова ТД схеми посилаць

Data0\_0.dbf

КОМПОНЕНТИ РОБОЧОЇ СХЕМИ ПОСИЛАНЬ

Імена полів БАЗОВОЇ ТД

Всі ключі схеми посилаць

KEY1  
KEY8  
KEY2  
KEY4  
KEY5  
KEY6  
KEY7  
KEY9  
KEY10  
KEY3

NAME  
KEY1  
KEY8  
KEY2  
KEY4  
KEY5  
KEY6  
KEY7  
KEY9  
KEY10  
KEY3  
INDEXS  
GOROOD  
KEY2  
ULICA  
DOM  
KVARTIRA  
TELEPHONE  
ORGANIZAC  
DATA\_ROGD  
SEX  
PULS  
KEY11  
PRESSURE  
ROST  
VES  
ZN\_ZOOIANKA  
MOON\_FAZA  
DATA\_1  
TIME\_1  
LEGK\_P1  
PERIK\_P1  
SERD\_P1  
TON\_C\_P1  
TOL\_C\_P1  
LEGK\_L1  
PERIK\_L1  
SERD\_L1

КОРІНЬ 14 моносхеми посилаць Y\_Lev = 0 (Базова ТД: Data0\_0)

```

1 KEY1 (ключ з TD = Data0_0) Y_Lev = 1 X_Lev = 1 (Y_Dist_ToRootTr = 0 | TDs_Id_KEY = 0)
2 KEY1 (ключ з TD = Data1_1) Y_Lev = 2 X_Lev = 2 (Y_Dist_ToRootTr = 1 | FISHKEY = 2 | TDs_Id_KEY = 0)
3 KEY8 (ключ з TD = Data1_1) Y_Lev = 3 X_Lev = 3 (Y_Dist_ToRootTr = 2 | TDs_Id_KEY = 0)
4 KEY8 (ключ з TD = Data8_1) Y_Lev = 4 X_Lev = 4 (Y_Dist_ToRootTr = 1 | FISHKEY = 1 | TDs_Id_KEY = 0)
5 KEY2 (ключ з TD = Data0_0) Y_Lev = 5 X_Lev = 5 (Y_Dist_ToRootTr = 0 | TDs_Id_KEY = 0)
6 KEY2 (ключ з TD = Data2_1) Y_Lev = 6 X_Lev = 6 (Y_Dist_ToRootTr = 1 | FISHKEY = 1 | TDs_Id_KEY = 0)
7 KEY4 (ключ з TD = Data2_1) Y_Lev = 7 X_Lev = 7 (Y_Dist_ToRootTr = 2 | TDs_Id_KEY = 0)
8 KEY4 (ключ з TD = Data4_1) Y_Lev = 8 X_Lev = 8 (Y_Dist_ToRootTr = 1 | FISHKEY = 1 | TDs_Id_KEY = 0)
9 KEY5 (ключ з TD = Data2_1) Y_Lev = 9 X_Lev = 9 (Y_Dist_ToRootTr = 4 | TDs_Id_KEY = 0)
10 KEY5 (ключ з TD = Data5_1) Y_Lev = 10 X_Lev = 10 (Y_Dist_ToRootTr = 1 | FISHKEY = 3 | TDs_Id_KEY = 0)
11 KEY6 (ключ з TD = Data5_1) Y_Lev = 11 X_Lev = 11 (Y_Dist_ToRootTr = 2 | TDs_Id_KEY = 0)
12 KEY6 (ключ з TD = Data6_1) Y_Lev = 12 X_Lev = 12 (Y_Dist_ToRootTr = 1 | FISHKEY = 2 | TDs_Id_KEY = 0)
13 KEY7 (ключ з TD = Data6_1) Y_Lev = 13 X_Lev = 13 (Y_Dist_ToRootTr = 2 | TDs_Id_KEY = 0)
14 KEY7 (ключ з TD = Data7_1) Y_Lev = 14 X_Lev = 14 (Y_Dist_ToRootTr = 1 | FISHKEY = 1 | TDs_Id_KEY = 0)
15 KEY9 (ключ з TD = Data5_1) Y_Lev = 15 X_Lev = 15 (Y_Dist_ToRootTr = 6 | TDs_Id_KEY = 0)
16 KEY9 (ключ з TD = Data9_3) Y_Lev = 16 X_Lev = 16 (Y_Dist_ToRootTr = 1 | FISHKEY = 1 | TDs_Id_KEY = 0)
17 KEY3 (ключ з TD = Data0_0) Y_Lev = 19 X_Lev = 19 (Y_Dist_ToRootTr = 0 | TDs_Id_KEY = 0)
18 KEY3 (ключ з TD = Data3_1) Y_Lev = 20 X_Lev = 20 (Y_Dist_ToRootTr = 1 | FISHKEY = 1 | TDs_Id_KEY = 0)

```

Результат будовання моносхеми посилаць на основі сукупності вибраних користувачем імен полів (KEY1, KEY2, KEY3) "базової" таблиці даних DATA0\_0.dbf, яка породила "активні" базові ключі схеми з іменами KEY1, KEY2, KEY3.

ПОЗНАЧЕННЯ:

- ТД - таблиця даних;
- номер в полі прямокутника - номер поля відповідної ТД.

ПОРЯДКОВИЙ ПОМЕР ТА X-РІВЕНЬ КЛЮЧА У МОНО АБО ПОЛІСХЕМІ

Задляти дані ключа схеми посилаць

Закрити роботу програми

ВІКНО ВИВЕДЕННЯ ІНФОРМАЦІЇ ЗАГАЛЬНОГО ТИПУ, А ТАКОЖ ПО КОЖНІЙ ОПЕРАЦІЇ

Отримати інформацію по АКТИВНІЙ операції

Скасувати модифікацію даних схеми посилаць

Рис. 1. Результат виконання створення КРЗ операції будовання нової моносхеми посилаць

об'єкта з робочого каталогу на зазначеному місці;

— **режим 1.4.** Вилучення моносхеми з зазначеного місця полісхеми посилань.

2. Операція перегляду існуючого варіанта структури (моно)полісхеми посилань, який щойно створено, або вже декілька разів модифікована.

3. Операція формування шляху на обраній моносхемі посилань.

4. Операція зміни даних активних ключів схеми посилань.

Долучені до операції 1 режими 1.3. і 1.4. на пряму не пов'язані зі зміною складу ключів конкретного Х-рівня (моно)полісхеми, яку переглядають чи модифікують. Їхнє віднесення до цієї групи операцій зумовлено відповідністю загальному сенсу операцій, які активує радіокнопка із заголовком “Змінити склад ключів схеми посилань”. У низці випадків, їхнє застосування дає можливість користувачеві отримати більш зручну для його візуального сприйняття БД у вигляді укрупненої, або деталізованої сукупності її окремих складових.

### **Операція модифікації наявних у робочому каталозі структур БД зміною складу ключів їхніх схем посилань**

Принципово важливою особливістю процесу експлуатації БД даних є те, що досить часто, а у ряді випадків і практично постійно, доводиться стикатися з ситуаціями, які змушують змінювати як номенклатуру ТД, долучених до складу БД, так і внутрішній зміст окремих з них. У першому випадку, такою причиною є об'єктивна необхідність долучення до складу робочої БД нових таблиць, або вилучення з її складу тих, подальше використання даних яких визнано недоцільним. Другий випадок, пов'язаний з модифікацією вмісту окремих ТД і, зазвичай, виникає в разі необхідності долучення в їхню структуру нових параметрів або вилучення полів, що більше не беруть участі в роботах.

В обох випадках цю ситуацію можна охарактеризувати як наявність у робочому каталозі нікому не потрібного “сміття”. Сміття, яке

за сучасних тенденцій повсюдної інформатизації має властивість накопичення і здатне не лише зменшувати ефективність використання наявних засобів зберігання інформації (займаючи значний їхній обсяг), а й суттєво спотворювати реальний стан справ, постійно збільшуючи різного роду витрати, породжуючи небажані ситуації та наслідки.

На фоні сказаного цілком обґрунтованим і закономірним виглядає бажання мати під рукою не тільки “чистий”, позбавлений сміття варіант робочої БД, але і засоби, здатні забезпечити досягнення бажаного результату максимально оперативно.

Оскільки в разі виникнення подібної ситуації “застаріла” схема посилань буде нести неправдиву інформацію, не здатну виконувати роль об'єктивного засобу контролю, її має бути або вилучено з робочого каталогу і на основі нових даних збудовано нову схему, або за допомогою відповідних засобів модифіковано її структуру. З точки зору часових та економічних показників, другий варіант є менш витратним і в нашому випадку, подібна модернізація зведеться до зміни складу (номенклатури) ключів, що входять в схему посилань.

Слід зазначити, що вирішення питань пов'язаних з долученням або вилученням таблиць зі складу БД, дасть відповідь і на питання модернізації схем посилань в разі “простого” збільшення номенклатури типів даних конкретних ТД. Тому під час вирішення питання вилучення зі схеми довільної кількості полів даних, що належать одній материнській ТД, до обов'язку користувача має увійти лише конкретизація їхніх імен. Всі інші питання, пов'язані з безпосереднім будуванням на нових умовах модифікованої структури БД, має вирішити сам КПЗ без допомоги і будь-яких додаткових витрат з боку користувача. У нашому випадку це стає реальним завдяки наявності КПЗ, що надає користувачу можливості оперативного змінення складу активних ключів схем посилань і формування нового варіанта їхньої структури.

Опису особливостей практичного застосування подібного роду програмних засобів,

розроблених і долучених до складу створеного КПЗ, присвячено наведені нижче матеріали.

Модифікацію схеми посилань завдяки зміні складу ключів здійснює операція, яку активує натискання радіо-кнопки із назвою “Змінити склад ключів схеми посилань”. Операція надає можливість роботи в двох режимах, які об’єднано в групу під заголовком “Варіанти модифікації складу ключів схеми посилань” і мають номери 1.1 і 1.2. Перш ніж перейти до опису кожного режиму, зазначимо ряд важливих моментів:

1. Незалежно від характеру зміни складу компонентів БД, яку модифікують, і даних, які в ній містяться, отриманий після практичного застосування будь-якого з зазначених режимів новий варіант БД і схеми посилань буде повністю чистий від “сміття”, а саме від тих ТД, необхідність подальшого використання яких користувач визнав недоцільним.

2. Натискання радіо-кнопки з написом “Змінити склад ключів схеми посилань”, не активізує жодного компонента групи 1.1—1.4. Вибір потрібного режиму модифікації цілком і повністю покладено на користувача.

3. Операції вилучення і долучення ключів створена версія КПЗ дозволяє застосовувати винятково до моносхем посилань. Їхнє застосування до полісхем, дані яких також містяться в одному *ref*-файлі, тимчасово заборонено. Причиною цього обмеження є те, що коректне застосування подібних операцій до будь-якої моносхеми посилань, що входить до складу полісхеми, потребує обов’язкової корекції структури такої моносхеми в усіх інших (моно) полісхемах, особисті дані яких містяться в інших *ref*-файлах, що живуть своїм “власним” життям. Щоб запобігти виникнення принципівих небажаних наслідків, зумовлених відсутністю корекції самої моносхеми посилань (наприклад, забудькуватість користувача), визнано розумною такою послідовність дій. Спочатку здійснити корекцію структури конкретної моносхеми. Потім за допомогою описаної нижче операції (радіо-кнопка “Від’єднати схему посилань від схеми посилань із зазначеного місця”) прибрати її “старий” екземпляр з усіх полісхем,

що її містять. Нарешті, долучити модифіковану моносхему до структури відповідних полісхем, використовуючи описану нижче операцію з радіо-кнопкою “Приєднати схему посилань до схеми посилань на зазначеному місці”.

4. Заборонено вилучення ключів, які є віртуальною 0-го *X*-рівня *ROOT*-вершиною дерева, або висячими вершинами (листя). У першому випадку це позбавить схему посилань обов’язкової зв’язності. У другому, вилучення ключа-листа зі структури схеми все одно збереже її придатність, але його присутність у схемі дає користувачеві змогу отримувати вельми корисну додаткову інформацію про структуру і склад робочої БД.

Розроблений і використаний підхід до структуризації даних *ref*-файлів, значно полегшив розв’язання задач зміни складу ключів схем посилань, які модифікують, зробив схожими структури блок-схем алгоритмів реалізації режимів вилучення і долучення до моносхем ключів різних *X*-рівнів, а також створення на їхній основі блоків програмних засобів.

**Режим 1.1. Вилучення зі схеми посилань неактивних ключів.** Термін “неактивні” ключі визначає факт, що результатом проведеного аналізу стало прийняття рішення на їхнє вилучення зі схеми для надання БД вигляду, який об’єктивно відображує її склад на поточний момент часу. Ключі, що залишилися у схемі, продовжать існувати на екрані монітора, візуально засвідчуючи активність і після перетворення схеми.

Робота користувача із засобами, що змінюють склад ключів схеми посилань завдяки операціям вилучення, або долучення, не залежить від того, базовими чи вторинними є такі ключі. Для розуміння дій користувач, що він має виконувати, досить познайомитися з тим, як це відбувається, в разі вилучення з схеми посилань базових ключів 1-го *X*-рівня.

**Випадок а).** Вилучення базових ключів зі схеми посилань. Відповідні за цей випадок операції, забезпечують одночасне вилучення зі схеми, що її модифікують, будь-якої кількості (але не всіх) присутніх в хедері базової матеріальної ТД) та зазначених користувачем

активних ключів і негайне (чи відкладене на час) будування повноцінної схеми посилань робочої БД на основі їхнього нового складу.

Виконання операції починається натисканням радіо-кнопки “Змінити склад ключів схеми посилань” і результатом цього є перехід в активний стан групи радіо-кнопок, об’єднаних загальним заголовком “Варіанти модифікації складу ключів схеми посилань”.

Натискання пов’язаної із цією операцією радіо-кнопки “Вилучити зі схеми посилань неактивні ключі”, забезпечить:

1. формування і виведення у вікні із заголовком “Файли створених схем посилань” повного списку *ref*-файлів, що містяться в робочій директорії;
2. активізацію кнопки з написом “Вивести на екран схему, що її модифікують”;
3. появу інформаційного повідомлення, що регламентує послідовність подальших дій користувача для забезпечення коректного виконання операції модифікації схеми.

Вибір з загального списку *ref*-файлу даних призначеної для модифікації схеми посилань і натискання зазначеної кнопки, призведе до:

- занесення імені цього файлу у вікно із заголовком “Ім’я робочої схеми посилань” і деактивізацію вікна зі списком усіх *ref*-файлів;
- виведення у головному репрезентативному вікні структури (графа-дерева) колись сформованої і нині вибраної для модифікації схеми посилань;
- активізації пари кнопок “Підготувати дані для виконання операції” і “Скасувати виконання вибраної операції”.

Знайомство з текстом повідомлення має завершитися натисканням кнопки “ОК”. Натискання кнопки “Скасувати виконання вибраної операції”, переведе КПЗ в стан головного меню. Виконання операції буде перервано, а дані раніше вибраної для модифікації схеми посилань, залишаться незмінними.

Навпаки, встановивши стрілку маніпулятора на належну вершину графа (випадок *a*) — будь-який ключ 1-го, випадок *б*) — ключ довільного, більшого за одиницю *X*-рівня), натиснути ліву клавішу для її маркування, а по-

тім натиснути кнопку “Підготувати дані для виконання операції”, операцію модифікації складу ключів буде продовжено.

Для конкретизації материнської ТД ключів, які вилучають, досить промаркувати лише один з них. На рис. 1 для випадку *a*), таким ключем може бути будь-який ключ 1-го *X*-рівня, ( $1 = KEY1$ ,  $4 = KEY2$ ,  $9 = KEY3$ ). Належність імен полів, які породили всі ці ключі, тій самій материнській ТД (*DATA0\_0.dbf*), підтверджують вирази “ключ з ТД *DATA0\_0*”, які присутні в перших круглих дужках усіх інформаційних рядків, наступних за прямокутниками відповідних вершин. Зазначене ім’я ТД також буде продубльовано у вікні редактора із заголовком “Ім’я власної ТД ключа”, який розміщено над зазначеним вікном.

У разі натискання кнопки вибору без попереднього маркування ключа схеми, засоби контролю КПЗ видадуть попередження про помилку, яку зробив користувач.

В разі маркування ключа, білий колір поля його прямокутника стане зеленим. Відмову від вибору ключа виконують повторним натисканням лівої кнопки маніпулятора на зеленому полі того самого прямокутника.

Після натискання останньої кнопки, в інформаційному вікні з назвою “Активні поля ТД ключа”, буде виведено перелік (для стислості — перший перелік) імен полів материнської ТД, які породили всі ключі 1-го *X*-рівня робочої схеми посилань.

В якості додаткової інформації, у вікні із заголовком “Імена полів власної ТД ключа” буде виведено перелік (для стислості — другий перелік) всіх імен полів, наявних у хедері раніше зазначеної ТД *Data0\_0.dbf*. Очевидно, що імена всіх полів першого списку є компонентами другого списку. Одночасно з цим буде активізовано кнопку “Створити/Модифікувати схему посилань”, а список в другому вікні стане неактивним.

Після цього, основним завданням користувача стає груповий вибір з першого списку імен тих полів, що підлягають вилученню зі схеми посилань, яку буде модифіковано. Якщо користувач, замість вибору хоча б одного імені

поля, натисне кнопку “Модифікувати схему посилань”, на екрані монітора з’явиться попередження, яке повідомить про зроблену помилку і вкаже, що треба робити задля її виправлення. Відмовою від вилучення зі схеми раніше промаркованого ключа є повторне натискання лівої кнопки маніпулятора після його розміщення на сірого кольору полі прямокутника, яке виділяє цей ключ.

Якщо після маркування ключів, що підлягають вилученню, натиснути кнопку “Модифікувати вибрану схему посилань”, з’явиться повідомлення, яке дає користувачеві можливість зберегти дані модифікованої схеми під новим ім’ям *ref*-файлу, що його задасть користувач, або старим, присутнім у вікні з заголовком “Ім’я робочої схеми посилань”.

В разі прийняття рішення про збереження даних в новому *ref*-файлі, чергове повідомлення КПЗ зазначить поле текстового редактора із заголовком “Ім’я модифікованої схеми”, де користувач має прописати це ім’я.

Натискання кнопки “ОК” завершить ознайомлення з повідомленням і активізує кнопку “Зберегти дані схеми під новим ім’ям”. Якщо до натискання цієї кнопки ім’я нового файлу модифікованої схеми не буде задано, вбудовані в КПЗ засоби контролю обов’язково сповістять користувача повідомленням про зроблену помилку;

Якщо ім’я задано вірно, то після натискання кнопки, в головному репрезентативному вікні буде виведено базовий варіант структури модифікованої схеми посилань, який містить тільки віртуальну 0-го *X*-рівня *ROOT*-вершину і всі збережені користувачем базові ключі. У вікні редактора із заголовком “Ім’я модифікованої схеми” буде зазначено повне (з додатним *ref*-розширенням) ім’я файлу, в якому записано дані базового варіанта модифікованої схеми, виведеного в репрезентативному вікні. У вікні із заголовком :Активні поля ТД ключа” буде виведено імена базових 0-го *X*-рівня ключів модифікованої схеми, які залишилися після вилучення, а у вікні із заголовком “Імена полів власної ТД ключа” — імена полів базової ТД, якою залишилася та сама *Data0\_0.dbf*.

Кнопка “Створити НОВУ схему посилань” стає активною. В разі її натискання КПЗ сформує з нового базового варіанта повноцінну схему посилань, що описує структуру модифікованого варіанта БД з відсутніми базовими ключами 1-го *X*-рівня, які було вилучено.

Небажання негайно формувати нову схему посилань дає змогу відкласти вирішення цього питання до будь-якого, зручного моменту часу.

В разі збереження даних модифікованої схеми у файлі зі старим ім’ям, всю інформацію, що характеризувала стару схему, буде замінено на нову без можливості відновлення першої. Всі ТД, що виключено зі складу старої БД і їхні імена, буде залишено без змін.

Активізація кнопки “Завершити роботу програми” переводить КПЗ у режим головного меню інтерфейсу користувача, надаючи можливість вирішення інших завдань.

**Випадок б).** Вилучення вторинних ключів зі схеми посилань. Цей режим призначено для одномоментного вилучення зі структури виведеної на екран схеми посилань будь-якої кількості  $q$  ( $1 \leq q \leq Q$ ) з  $Q$  наявних однорівневих, зазначених користувачем ключів  $\{KEY_q^j\}_{q=1}^Q$   $j$ -го ( $1 < j$ ) *X*-рівня, породжених іменами полів загальної для всіх них замість їхньої материнської ТД *dbf*-формату файлу.

В цьому випадку, дії користувача повністю ідентичні описаним у випадку *a*). Тому, зазначимо лише те, що в разі вилучення ключів зі схеми посилань, до тих вершин її графа-дерева, що не припускать маркування, віднесено:

- *ROOT*-вершина нульового *X*-рівня;
- вершини, які породжені іменами полів, які в хедері власної материнської ТД мають нульові значення порядкових номерів;
- висячі  $\{k_i\}$   $i$ -х *X*-рівнів вершини (листя) графа-дерева.

Останні, не мають суміжних вершин чергових  $\{k_i + 1\}$ -х *X*-рівнів. Вони лише виконують роль індикаторів, що конкретизують місця можливого долучення в наявну БД нових ТД, або розташування потенційно можливих точок розгалуження в схемі посилань. На кольоровому екрані візуальною ознакою заборони

маркування таких ключів, є червоний колір цифри '0' всередині прямокутника і інформаційного рядка, який за ним. На чорно-білих рисунках, червоний колір замінено потовщенням бічних ребер рамок прямокутників. На рис. 1 дозволено маркувати і вилучати вторинні ключі  $KEY8$  ( $Y_{Lev} = 3$ ),  $KEY4$  ( $Y_{Lev} = 7$ ),  $KEY5$  ( $Y_{Lev} = 9$ )), що є породженими полями материнських ТД  $Data1\_1$  і  $Data2\_1$  відповідно. В той же час, ключі  $KEY1$  ( $Y_{Lev} = 2$ ),  $KEY2$  ( $Y_{Lev} = 6$ ) і  $KEY3$  ( $Y_{Lev} = 20$ ), мають потовщені бічні ребра рамки, не підлягають маркуванню і вилученню бо породжені полями таблиць  $Data1\_1$ ,  $Data2\_1$  і  $Data3\_1z$  0-ми порядковими номерами.

У обох а) та б) випадках, одночасно з вилученням зі схеми посилань промаркованих ключів буде вилучено всі вершини гілок, для яких ці ключі є коренями. В свою чергу, зі складу модифікованої БД буде вилучено всі ТД, поля яких породили ці ключі, але самі ТД буде збережено недоторканими в робочому каталозі під колишніми \*.ref-іменами.

**Режим 1.2. Долучення до схеми посилань нових ключів.** Задача долучення нових даних до наявних БД (і ключів до їхніх схем посилань), виникає в разі потреби збільшення номенклатури використовуваних в роботі параметрів і показників. Зробити це можна долученням до складу робочої БД нових ТД з робочого каталогу, або додаванням (за допомогою *DBF Navigator* чи їм подібних засобів) в самі ТД нових полів. Третім варіантом вирішення задачі є комбінація (симбіоз) двох попередніх.

З одного боку, такі зміни є чисто "технічними" і не потребують значних витрат часу і технічних ресурсів. З іншого, збільшення номенклатури даних призведе до ускладнення розуміння механізму їхньої взаємодії в БД. Найефективнішим і найменш ресурсовитратним у реалізації способом отримання цілісної картини того, що реально має відбутися, є модифікація відповідних схем посилань у повній згоді зі змінами, що можуть виникнути.

На відміну від вже розглянутого вилучення даних з БД, питання врахування даних,

що додають до структури вже наявної схеми посилань, вирішується зовсім іншим способом. Причина цього в тому, що алгоритми і методи його програмної реалізації принципово залежать від місця в схемі, де саме, згідно вимогам загальної логіки внутрішніх взаємозв'язків сукупності всіх даних робочої БД, має бути додано ключі, що ініціюють подібні зміни.

Як і в пов'язаному з вилученням ключів режимі 1.1, зазначена обставина породжує необхідність розгляду наступних двох випадків.

**Випадок 1.** Додавання базових ключів у схему посилань. Відповідна за реалізацію цього випадку радіо-кнопка має напис "Додати в схему посилань нові ключі". Після її активізації у вікні із заголовком "Файли створених схем посилань" буде виведено перелік імен усіх наявних в робочому каталозі моносхем, повноцінної структури яких вже було сформовано користувачем. Як приклад схеми посилань, що підлягає модифікації, розглядатимемо структуру, наведену на рис. 2.

Вибір зі списку імені моносхеми, яку має бути модифіковано, і натискання кнопки "Вивести на екран схему посилань", призведе до появи цього імені у вікні із заголовком "Ім'я робочої схеми посилань", візуалізації у головному репрезентативному вікні її структури, активізації пари кнопок з написами "Підготувати дані для виконання операції" і "Скасувати виконання вибраної операції". Буде також виведено повідомлення, що зорієнтує користувача в послідовності дій, які він має виконати. Цей випадок дозволяє маркування будь-якої вершини 1-го X-рівня. Тому, щоб додати до схеми дані саме базової ТД з ім'ям  $Data0\_0$ , маємо право вибрати 1 =  $KEY1$ , 4 =  $KEY2$ , або 9 =  $KEY3$ .

Призначення кнопки, пов'язаної зі скасуванням операції, було описано раніше.

Натискання кнопки "Підготувати дані для виконання операції" виведе в умовно першому праворуч вікні із заголовком "Активні поля ТД ключа" перелік усіх наявних у схемі посилань, що її модифікують, імен ключів, породжених активними полями материнської ТД тільки що промаркованого ключа, а саме,  $Data0\_0$ . Паралельно з цим, в умовно другому

праворуч вікні із заголовком “Імена полів власної ТД ключа”, з’явиться перелік усіх імен полів цієї ТД, що містить і імена полів, що породжують ключі 1-го  $X$ -рівня, імена яких також присутні в першому правому вікні. Одночасна наявність двох списків в парі вікон праворуч, спрощує вибір у другому з них імен тих полів, які, в разі вибору, породять ключі, що КПЗ долучить до схеми посилань. Також буде конкретизовано імена всіх ТД, які додатково увійдуть до складу повноцінної модифікованої БД, і виведено повідомлення, регламентуюче порядок дій користувача.

Маркування імен полів і натискання кнопки “Модифікувати вибрану схему посилань” виведе повідомлення про прийняте КПЗ рішення щодо остаточного вибору імен полів із зазначенням причин відмови іншим іменам не придатним для розв’язання завдання модифікації схеми шляхом додавання ключів. Вбудовані в КПЗ засоби контролю також відмовлять повторному використанню в модифікованій схемі вже наявних в лівому списку імен, і імен не здатних породити у схемі нові ключі 1-го  $X$ -рівня через брак у робочому каталозі ТД з такими іменами полів. В разі невиконання вимоги вибору імен полів, КПЗ також попередить користувача про зроблену ним помилку.

Завершення вибору придатних до використання імен полів надає користувачу можливості вибору варіанта збереження даних нової схеми посилань у файлі під новим або старим ім’ям. У першому випадку дані старої БД і схеми буде збережено без змін під її колишнім ім’ям, а дані нового варіанта буде записано у файл, ім’я якого має занести користувач у поле текстового редактора, який активізовано під заголовком “Ім’я модифікованої схеми”. Запис даних здійсниться після натискання кнопки “Зберегти дані схеми під новим ім’ям”.

У другому випадку дані попереднього варіанта буде безповоротно втрачено, через їхню заміну новими після натискання кнопки “Зберегти дані схеми під старим ім’ям”.

Після реалізації другого випадку, у головному репрезентативному вікні буде виведено

оновлений базовий варіант структури схеми, що містить як раніше наявні в ньому, так і додані в схему ключі, які слугуватимуть основою для будування повноцінної схеми на основі всієї сукупності присутніх в робочому каталозі ТД. Результатом натискання активованої кнопки “Створити нову схему посилань” стане повноцінна схема посилань, об’єктивно відображаюча взаємозв’язок даних в новому варіанті робочої БД. Як і завжди, будування нової повноцінної схеми можна відкласти до слушного моменту.

**Випадак 2.** Додавання вторинних ключів у схему посилань. Як було зазначено, порядок виконання дій, пов’язаних із додаванням у стару структуру БД нових, розміщених в робочому каталозі ТД, є тотожним описаному вище. Маркувати дозволено будь-який ключ (окрім *ROOT* дерева та листя, породжених полями ТД з 0-ми порядковими номерами), значення  $X$ -рівня якого в структурі схеми більше за одиницю.

Вважаючи за недоцільне повторювати раніше викладене і, вибравши для модифікації нову схему, зазначимо лише те, що результат додавання до неї вторинного ключа 3-го  $X$ -рівня  $9 = KEY10$  (ключ з *TD Data5\_1*) наведено на рис. 2.

Завершуючи опис операції додавання ключів у схему, має сенс виділити ряд важливих моментів. Перше, якщо на час додавання в схему посилань нових ключів, з робочого каталогу було вилучено ряд ТД, раніше використовуваних у БД, то в результаті описуваної операції всіх їх буде також вилучено зі структури модифікованої схеми. Друге, для об’єктивного відображення виниклої ситуації, операцію модифікації буде необхідно виконувати над усіма наявними в робочому каталозі схемами, структура яких раніше включала всі вилучені з робочого каталогу ТД, а також ті, в які тепер мають входити додані ТД. Третє, описана пара режимів модифікації КПЗ виконує окремо і лише на одній, вибраній для роботи моносхемі посилань. Тому в разі наявності кількох БД, що використовують ТД зі зміненими даними (містять ті самі ТД), модифікація кожної з них має бути виконана окремо.

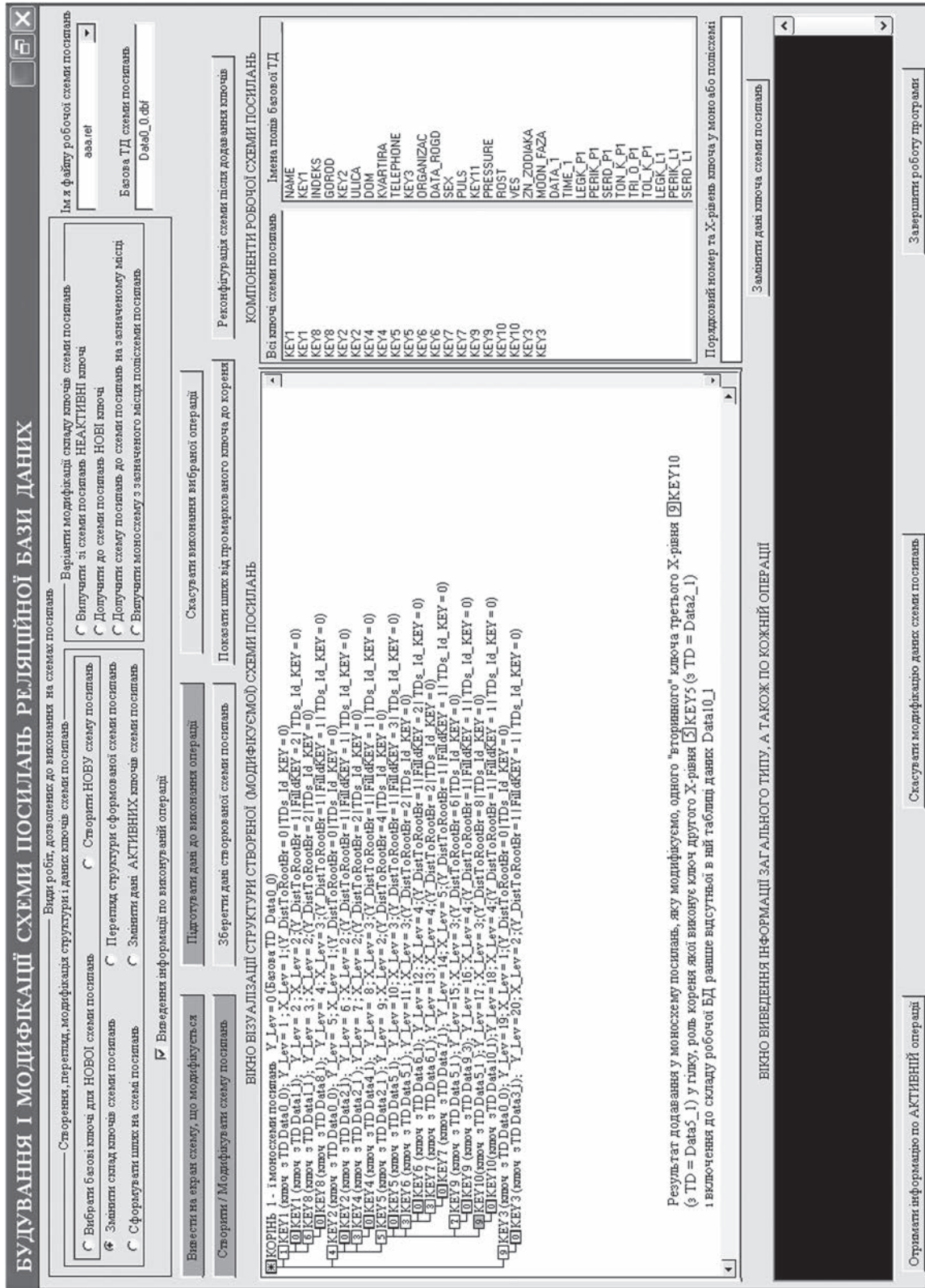


Рис. 1. Результат долучення в умовно початковий варіант моносхеми посилань ключів 3-го рівня



**Режим 1.3. Долучення до структури обраної (моно)полісхеми аналогічного типу об'єкта з робочого каталогу на зазначеному місці.** Іноді виникають ситуації, коли  $r$  ( $1 < r$ ) різних за призначенням і змістом БД має сенс об'єднати в єдину групу згідно деяких ознак. Ця обставина робить доцільним завжди мати “під рукою” засоби, що здатні коректно інтегрувати в одному файлі дані всіх пов'язаних з такими БД схем посилань і організувати їх одночасний перегляд на екрані. Операцію, яка ефективно вирішує таке завдання, називатимемо операцією приєднання схем посилань.

Зазначений режим вирішує питання створення полісхем посилань, які інтегрують в одному *ref*-файлі дані  $r$  ( $1 < r$ ) довільно упорядкованих структур моносхем з метою отримання можливості їхнього одночасного перегляду в головному репрезентативному вікні.

Кількість моносхем, інтегрованих в конкретних полісхемах і місця розташування кожної з них на екрані визначає користувач. Встановлений ним порядок розміщення моносхем зверху до низу, засвідчують інформаційні рядки (КОРІНЬ \*-ї моносхеми посилань), що в якості параметра використовують порядкові номери, які починаються з одиниці.

Для всіх чотирьох можливих варіантів приєднання: 1) моносхема до моносхеми; 2) моносхема до полісхеми; 3) полісхема до моносхеми і 4) полісхема до полісхеми, зазначені дії є і правомірними і коректними за постановкою. Програмна реалізація режиму є універсальною, і правила роботи в ньому користувача не залежать від приналежності схем посилань, що беруть участь в операції, до типу моносхем або полісхем.

Робота в режимі починається з натискання радіо-кнопки “Змінити склад ключів схеми посилань”. Після цього серед чотирьох активованих і об'єднаних заголовком “Варіанти модифікації складу ключів схеми посилань” радіо-кнопок треба вибрати з написом “Приєднати схему посилань до схеми посилань на зазначеному місці”.

Вибір зі списку, який розкривається під заголовком “Схеми посилань що модифікують”,

імені файлу робочої схеми посилань, що підлягає модифікації, і натискання кнопки “Вивести на екран схему, що її модифікують” забезпечить:

- виведення в головному репрезентативному вікні обраної для модифікації схеми;
- виведення у вікні із заголовком “Видимі ключі схеми” списку імен усіх, наявних у обраній схемі ключів;
- виведення у вікні “Приєднані схеми посилань” переліку імен *ref*-файлів усіх присутніх в робочому каталозі повноцінних схем посилань для можливості вибору з них тієї, яку буде приєднано до виведеного на екрані об'єкта;
- можливість вибору маркуванням місця додавання (моно)полісхеми, з метою подальшого долучення її структури до наведеної на екрані сукупності структур;
- активізацію кнопки “Підготувати дані для виконання операції”.

Місце вставки визначає маркування будь-якого ключа тієї схеми посилань, безпосередньо (візуально) під якою буде розміщено структуру, що буде додано в наведену на екрані сукупність. Заборонено маркувати ключі, які є *ROOT*-вершинами графів-дерев, або прямокутники яких мають цифру ‘0’ у внутрішній частині поля.

Вибір із активізованого вікна другого імені файлу схеми, яку буде приєднано до раніше вибраної і натискання кнопки “Підготувати дані для виконання операції” активізує кнопку “Модифікувати вибрану схему посилань”. Порядок вибору другої схеми і маркування місця її долучення в першу схему не має значення.

За відсутності вибору схеми, яку приєднують (в каталозі була єдина, вже вибрана, схема, або її ім'я збігається з уже наявною на екрані), вбудовані в КПЗ засоби контролю видадуть відповідне попередження.

Якщо вибір схеми, яку приєднують, не зумовив негативної реакції КПЗ, активізація кнопки “Модифікувати вибрану схему посилань”, призведе до виведення в головному репрезентативному вікні оновленого варіанта схеми посилань, який містить в своїй структурі

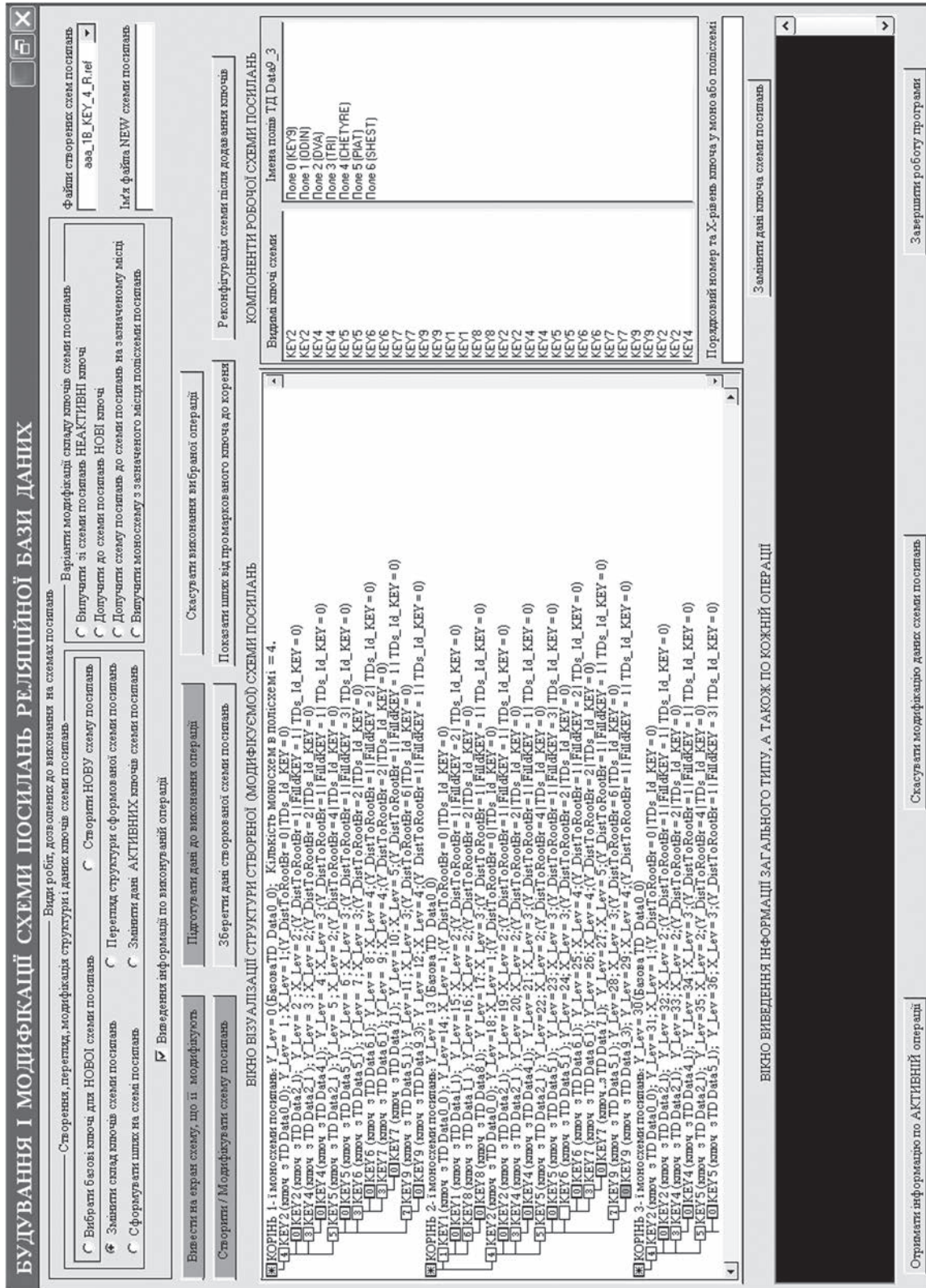


Рис. 3. Результат послідовного долучення (одне до одного) трьох схем посилань

все, що містилося у ній до операції і все долучене в процесі її виконання.

На рис. 3 показано результат послідовного приєднання чотирьох, різного обсягу моно-схем, з яких лише повні дві візуально присутні на екрані. Сказане підтверджує напис у першому згорі рядку екрана, який конкретизує загальну кількість моно-схем, інтегрованих у складі полісхем, отриманих в результаті використання описаного режиму. Використання вертикального *scroll*-движка легко доводить справдливність сказаного.

**Режим 1.4. Вилучення моно-схеми з зазначеного місця полісхеми посилань.** Задачу, що вирішує цей режим, можна умовно визначити як “спрощений” варіант загального вигляду оберненої задачі, яку вирішує режим 1.3. У цій версії КПЗ дозволене одномоментне вилучення лише однієї моно-схеми з декількох наявних, незалежно від їхнього розташування у полісхемі.

Q вхідних в полісхему моно-схем послідовно (зверху до низу) пронумеровано натуральними числами  $q$  ( $1 \leq q \leq Q$ ). Оскільки черговість вилучення  $r$  ( $1 \leq r < Q$ ) окремих моно-схем не має значення, найбільш доцільною визнано  $r$ -крокову процедуру, яка є позбавленою від принципів недоліків і здатна успішно вирішити це завдання.

Початком реалізації режиму є натискання радіо-кнопки “Змінити склад ключів схеми посилань”. Скасувати його з метою заміни операції або режиму роботи, можна натисканням будь-якої з п’яти розміщених поруч радіо-кнопки меню, що залишилися активними.

Продовжити роботу в цьому режимі можна після вибору користувачем імені робочого *ref*-файлу зі списку, наведеному у вікні із заголовком “Схеми посилань, що їх модифікують”. Зроблений вибір і наступне натискання кнопки “Вивести на екран схему посилань, що її модифікують”, виведуть в полі головного репрезентативного вікна структуру робочої схеми для конкретизації моно-схеми, яку з неї вилучатимуть.

Ознакою вибору моно-схеми, яка підлягатиме вилученню є зелений (на рисунку — сірий) колір поля будь-якого її прямокутника,

що його промаркує користувач лівою кнопкою маніпулятора. Скасування ухваленого рішення з метою заміни призначеної для вилучення моно-схеми візуально виглядає як демаркування підсвіченого раніше ключа (зміна кольору поля прямокутника на первинний) за допомогою тією самої кнопки маніпулятора.

В результаті послідовного натискання кнопок “Підготувати дані для виконання операції” і “Модифікувати вибрану схему посилань” КПЗ, виконавши всі необхідні дії щодо підготовки та подальшого оброблення даних структури вибраної для перетворення полісхеми посилань, виведе на екран її модифікований варіант.

### **Операція перегляду існуючого варіанта структури (моно)полісхеми посилань, який щойно створений, або вже декілька разів модифікований**

Призначення операції впливає з самої назви. Її використання надає змогу оглянути результати раніше виконаних операцій з метою їхньої можливої корекції. Вона ефективно працює як з моно (рис. 1), так і з полісхемами (рис. 3) посилань, що інтегрують в собі довільну кількість моно-схем.

Після натискання радіо-кнопки “Переглянути структуру сформованої схеми посилань”, відповідальна за виконання цієї операції програма автоматично сформує і виведе у вікні із заголовком “Файли створених схем посилань” список всіх раніше створених схем. Пропонуючи користувачу вибрати зі списку ім’я потрібного *ref*-файлу, програма одночасно активізує кнопку з написом “Вивести структуру створеної схеми посилань”. Результатом послідовного маркування імені файлу і натискання цієї кнопки є візуалізація у головному репрезентативному вікні структури потрібної схеми і активізація кнопки “Скасувати виконання вибраної операції”, призначення якої вже було сформульоване вище.

В разі переміщення стрілки маніпулятора по полю головного репрезентативного вікна і потрапляння на прямокутне поле будь-якого

ключа схеми, в нижньому правому вікні із заголовком “Порядковий номер і  $X$ -рівень ключа в моносхемі та полісхемі”, КПЗ виводить пару параметрів, які конкретизують місце розташування ключа в робочій схемі посилань. А саме, значення  $Y$ -рівня (Ключ № \* з \*\*) і  $X$ -рівня ключа з загальною кількістю  $X$ -рівнів ( $X$ -рівень \*\*\*) (з \*\*\*\*) серед усіх  $Y$ - і  $X$ -рівнів графа-дерева моносхеми, структурним компонентом якого він є. У вікні із заголовком “Видимі ключі схеми” також буде виведено перелік усіх ключів, які входять в структуру вибраної для перегляду (моно)полісхеми.

Розміщення стрілки на прямокутнику будь-якого ключа схеми (крім *ROOT*-вершин графів-дерев) і натискання його правої клавіші призводить до окраса поля цього ключа зеленим кольором (сірим — на рисунку). В інформаційному вікні із заголовком “Імена полів ТД \*”, з'являться реальні номери й імена полів материнської ТД, одне з полів якої породило промаркований і зазначений в дужках ключ.

### Операція формування шляху на обраній моносхемі посилань

Операція призначена для автоматизованого формування і маркування на створеній (моно) полісхемі посилань шляху, що відображає специфіку взаємозв'язку обраної користувачем вершини з конкретним (базовим) ключем 1-го  $X$ -рівня. Вона забезпечує візуальний контроль механізму доступу до конкретного типу даних робочої БД.

Початком виконання операції є активізація радіо-кнопки “Сформувати шлях на схемі посилань” і виведення у вікні із заголовком “Файли створених схем посилань” списку всіх *ref*-файлів, що містять дані повністю сформованих (моно)полісхем посилань, наявних в робочому каталозі. Вибір імені *ref*-файлу для роботи і натиснення кнопки “Підготувати дані для виконання операції” призводить до виведення в полі головного репрезентативного вікна структури схеми, а у вікні із заголовком “Видимі ключі робочої схеми” переліку активних ключів усіх моносхем, інтегрованих у по-

лісхемі. Якщо розміри поля головного репрезентативного вікна недостатні для розміщення всієї структури схеми, у вікні із заголовком “Видимі ключі робочої схеми” буде виведено перелік тих активних ключів, які в кожен конкретний момент часу є видимими на екрані. Переглянути структури всіх наявних у вибраному файлі моносхем посилань і сукупностей всіх вхідних в такі структури ключів дозволяють *scroll*-движки відповідних вікон.

Вивівши на екран згадані об'єкти, програма перейде в режим вибору ключа, що є умовно початковим пунктом шляху і очікування натискання кнопки “Показати шлях від промаркованого ключа до кореня”. Виконання в зазначеному порядку обох дій, дає змогу візуально ознайомитися з особливостями сформованого шляху, який на екрані буде позначено зеленого кольору полями прямокутників відповідних вершин графа  $\{A_{ij}^d\}$ . “Початковий” (промаркована вершина) і “кінцевий” (*ROOT*-вершина) пункти шляху на екрані виділено жовтим кольором полів прямокутників. У чорно-білих тонах друку, цей шлях має сірого кольору поля прямокутників вершин дерева.

Якщо без вибору вершини, натиснути кнопку “Показати шлях від промаркованого ключа до кореня”, програма видасть повідомлення з попередженням.

У разі правильного вибору ключа в розміщених праворуч інформаційних вікнах із заголовками “Видимі ключі схеми” і “Всі шляхи полісхеми посилань” буде виведено розділені заголовками “Всі ключі \*\* - і схеми” списки всіх наявних у головному інформаційному вікні ключів і всі шляхи  $d$ -ї ( $1 \leq d \leq D$ ) моносхеми, інтегрованої у полісхемі.

Задля зручності візуального сприйняття, кожній моносхемі, сукупності ключів її структури і всім наявним на ній шляхам надано той самий порядковий номер  $d$  ( $d = 1, 2, 3, \dots, D$ ). Кожна сукупність ключів має заголовок “Усі ключі схеми № \*” з зазначенням порядкового номера (№ \*) пов'язаної з ним моносхеми, а сукупності відповідних їй шляхів передре заголовком “Всі шляхи моносхеми № \*”. В свою чергу, сформований шлях з промаркованою користу-

вачем вершиною, виділено синім кольором фону символного рядка.

Порядкові номери моносхем також продубльовано в головному репрезентативному вікні. Вони присутні в рядках коментарів при відповідних *ROOT*-вершинах і розміщуються безпосередньо за словом “корінь”.

Програмний модуль, який реалізує цю операцію, збудовано таким чином, що умовно початковими пунктами шляхів можна обирати лише ізольовані вершини графа (листя). Доцільність введення подібного обмеження зобов'язана наступним аргументам:

- для оперативного перегляду, вся корисна для користувача інформація може бути надана в максимально компактній, загального вигляду і змісту формі;

- будь-яка (не *ROOT* і не лист) вершина графа-дерева є внутрішнім (проміжним) елементом хоча б одного шляху, умовно початковим пунктом якого є конкретна всяча вершина, а кінцевим — загальна для всього графа віртуальна *ROOT*-вершина;

- одночасна наявність на екрані ряду схем посилянь, сукупності вхідних до їхнього складу ключів і явного вигляду шляхів, формалізованих інтуїтивно усвідомлюваними символними рядками з конкретними значеннями *Y*-рівнів ключів цих шляхів, дає змогу користувачеві проаналізувати будь-яку ділянку будь-якого шляху, зрозуміти всі нюанси його побудови і складу компонентів. Зокрема, серед вершин виділеного шляху знайти потрібну вершину і, зважаючи на розв'язувану задачу, оцінити всі наявні на відповідній схемі посилянь шляхи, для яких остання є проміжним пунктом.

Під парою інформаційних вікон правої частини інтерфейсу, виведено додаткові параметри ключа, призначеного умовно початковим пунктом збудованого шляху. Ці параметри конкретизують значення його власних *Y* і *X*-рівнів серед усіх *Y* і *X*-рівнів ключів робочої схеми посилянь з головного репрезентативного вікна.

Як і у режимі перегляду структури схеми посилянь, скориставшись правою кнопкою маніпулятора до моменту натискання кнопки

“Показати шлях від промаркованого ключа до кореня”, можна отримати інформацію, виведену у вікнах із заголовками “Видимі ключі робочої схеми” і “Імена полів ТД \*”. Активізація кнопки “Показати шлях від промаркованого ключа до кореня” унеможливує отримання цієї інформації, оскільки в даному випадку операції маркування ключа і будівництва шляху взаємно виключатимуть одна одну.

Натискання кнопки “Показати шлях від промаркованого ключа до кореня” сформує шлях і у відповідних інформаційних вікнах виведе усі пов'язані з цією операцією характеристики самого ключа і робочого варіанта схеми посилянь. Результат виконання цієї операції на полісхемі посилянь, що інтегрує три моносхеми, показано на рис. 4.

Питання будівництва шляху з промаркованою вершиною графу, яка не є листом, має вирішувати вже розроблений і включений до складу КПЗ програмний модуль, питання доцільності активізації якого поки є відкритим. Аргументи, які розглянуто і проаналізовано, недостатньо переконливі для прийняття позитивного рішення.

### **Операція зміни даних активних ключів схеми посилянь**

Причиною включення такої операції до складу КПЗ стала орієнтація на подальші розробки, які передбачають можливість створення спеціалізованої інформаційної *software*-технології і запису (у вигляді структурованих) символних рядків сукупності значень параметрів, що їх використовують як додаткові характеристики самих ключів схеми посилянь і зв'язків компонентів окремих БД, ТД тощо. Особливості конкретизації сукупності додаткових параметрів, що підлягає збереженню в бінарного типу файлі і специфіка модифікації їхніх значень безпосередньо залежать від галузі використання конкретної БД. Тому довжина рядка повинна мати здатність адаптування під відповідний клас об'єктів і кількість характерних для нього параметрів, що підлягатимуть обов'язковому врахуванню.

Тимчасовий брак належного рішення, став об'єктивною причиною, яка примусила перенести на зручніший момент часу розроблення програмного модуля, що реалізує цю операцію.

## Висновки

Запропоновано новий підхід та наведено результат його програмної реалізації у вигляді “пілотного” варіанта комплексу програмних засобів автоматизованого будування схем посилань реляційних баз даних. Основу підходу складають способи, методи та алгоритми, які розроблено і програмно реалізовано у новій, створеній універсальній інформаційній *software*-технології для роботи з (поки що) неорієнтованими графами виду “змішаного лісу”. КПЗ робить “формальною” участь користувача в процесі будування схем посилань. Багаторазово скорочуючи потрібний для цього час і забезпечуючи більше, ніж 90% економію технічних ресурсів використовуваного комп'ютера, КПЗ дає змогу враховувати таблиці даних, загальна кількість полів яких є обмеженою лише його спроможністю.

Наведені результати є “пілотним” варіантом, дослідження якого має бути ретельно проаналізовано. Позитивний висновок щодо придатності такого підходу, стимулюватиме його подальше удосконалення у вигляді спеціалізованої інформаційної *software*-технології як ефективного засобу розв'язання задачі створення схем посилань реляційних БД.

Наведені результати є “пілотним” варіантом, дослідження якого має бути ретельно проаналізовано. Позитивний висновок щодо придатності такого підходу, стимулюватиме його подальше удосконалення у вигляді спеціалізованої інформаційної *software*-технології як ефективного засобу розв'язання задачі створення схем посилань реляційних БД.

## ЛІТЕРАТУРА

1. Дейт К.Дж. Введение в системы баз данных. Киев; Москва: Диалектика, 1998. 781 с.
2. Ivaneshkin A.I. Universal Information Software Technology for Non-Oriented Mixed Forests. *Cybernetics and Systems Analysis*. 2022. Vol. 58, № 3. May, pp. 465-472. <https://doi.org/10.1007/s10559-022-00477-s>
3. Ivaneshkin A.I. The Solution of the Question of the Isomorphism of Non-Oriented Trees by the Method of Generating Isomorphic Structures. *Automation and Information Sciences*. 2020. Vol.52, № 4, pp. 68-79. DOI: 10.1615/JAutomatInfScien.v52.i8.60
4. Ivaneshkin A.I. A New Approach to Operating with Undirected Trees. *Cybernetics and Systems Analysis*. 2021. Vol. 57, № 1, pp. 124-132. <https://doi.org/10.1007/s10559-021-00335-z>.
5. Ivaneshkin A.I. Undirected mixed-forest graphs. Constructing isomorphic analogs and selective structure packing, *Cybernetics and Systems Analysis*. 2023. Vol. 59, No. 3, May, pp. 503-510. DOI: <https://doi.org/10.1007/s10559-023-00585-z>

Надійшла 05.04.2024

## REFERENCES

1. Deyt, K.Dzh. (1998). *Vvedeniye v sistemy baz dannykh*. Kyiv; Moskva: Dialektika, 781 p.
2. Ivaneshkin, A.I. (2022). “Universal Information Software Technology for Non-Oriented Mixed Forests”. *Cybernetics and Systems Analysis*. 58 (3), pp. 465-472. <https://doi.org/10.1007/s10559-022-00477-s>
3. Ivaneshkin, A.I. (2020). “The Solution of the Question of the Isomorphism of Non-Oriented Trees by the Method of Generating Isomorphic Structures”. *Automation and Information Sciences*. 52 (4), pp. 68-79. DOI: 10.1615/JAutomatInfScien.v52.i8.60
4. Ivaneshkin, A.I. (2021). “A New Approach to Operating with Undirected Trees”. *Cybernetics and Systems Analysis*. 57 (1), pp. 124-132. <https://doi.org/10.1007/s10559-021-00335-z>
5. Ivaneshkin, A.I. Undirected mixed-forest graphs. Constructing isomorphic analogs and selective structure packing, *Cybernetics and Systems Analysis*. 2023. Vol. 59, No. 3, May, pp. 503—510. DOI: <https://doi.org/10.1007/s10559-023-00585-z>

Received 05.04.2024

*O.I. Ivaneshkin*, Doctor of Technical Sciences, Leading Researcher,  
International Research and Training Center for Information Technologies and Systems NAS and MES of Ukraine,  
Acad. Glushkov Ave., 40, Kyiv, Ukraine, 03187,  
ORCID: <https://orcid.org/0009-0006-6800-2944>,  
[al.ivaneshkin@gmail.com](mailto:al.ivaneshkin@gmail.com)

## **ON ONE APPROACH TO SOLVING THE PROBLEM OF AUTOMATED CONSTRUCTION OF RELATIONAL DATABASE LINK SCHEMES**

**Introduction.** The current trend of continuous integration of various areas of scientific and practical human activity contributes to the constant increase in the complexity of problems, the successful solution of which becomes possible only through the use of large volumes of information, as a rule, geographically distributed across multiple database files. In this case, the need to know the locations and features of the interaction of its components becomes of fundamental importance when organizing the most convenient and efficient ways to access it.

In some cases, to meet such needs, you can use the services of existing products (Microsoft Access, Microsoft SQL Server, MySQL, FileMaker Pro, Oracle Database, dBASE, etc.). Sometimes, using a specialized SQL language or its dialects, it is more convenient to create a “handmade” one that will satisfy the stated requirements.

If the user desires or needs to know the location of the information used in the work, understand the logic of the mechanism for obtaining it, and evaluate the effectiveness of the ways of its delivery to constantly monitor its current and future nomenclature and quantity, its developments can help, which can provide more detailed information, than known means.

As experience shows, obtaining an integral and effective solution to this set of closely interrelated issues becomes possible only with the availability of a purely specialized software tool for the automated construction of link diagrams, which is a multifunctional, operational, easy-to-learn, and easy-to-use user environment. Endowed with the ability at any time and with minimal user participation to visualize on the screen the current structure of the created database with the required degree of clarity, detail, and ease of perception, such an environment should become a means of operational analysis and assessment of ways of possible modification of the database itself in the direction of creating a more effective one in practical terms, regarding the variant of its structure.

**Purpose.** Research and practical testing of the existing capabilities of approaches, methods, algorithms, and software modules that were developed during the creation of a universal information software technology for working with undirected graphs of the “mixed forest” type. To evaluate the efficiency and feasibility of using these approaches to improve a set of integrated software tools for the automated construction of relational database link diagrams with minimal user participation in this process.

**Methods.** Database systems, apparatus of graph-tree theory.

**Result.** A new approach is proposed, and the result of its software implementation is presented in the form of a “pilot” version of a set of software tools for the automated construction of relational database link schemes. The basis of the approach is techniques, methods, algorithms, etc. which were created and implemented programmatically in a new, universal information software technology for working with (temporarily) undirected graphs of the “sour cream forest” type. Software package (SP) makes “formal” the user’s participation in the process of building link diagrams. By greatly reducing the time required for this and providing more than 90% savings in the technical resources of the computer used, SP makes it possible to take into account data tables, the total number of fields of which is limited only by the latter’s capabilities. A positive conclusion about the suitability of such an approach will become an incentive for its further improvement in the form of specialized information software technology as an effective means of solving the problems of constructing relational database link schemes.

**Keywords:** *Relational databases, data tables, link scheme, undirected tree graphs, software package, information software technology.*