УДК 004.655
DOI: 10.37069/1683-4720-2019-33-8

©2019. **І.С. Канарська**

# АЛГОРИТМИ, ЩО РЕАЛІЗУЮТЬ ТЕОРЕТИКО-МНОЖИННІ ОПЕРАЦІЇ НА ТАБЛИЦЯХ ТА МУЛЬТИТАБЛИЦЯХ

Робота присвячена дослідженню алгоритмів, що реалізують операції перетину, об'єднання та різниці у таблицях та мультитаблицях. Тематика роботи є актуальною, оскільки незважаючи на важливість та вживаність теоретико-множинних операцій у реляційних базах даних, внаслідок певних причин, увагу дослідників було зосереджено на оптимізації інших табличних операцій, у першу чергу – з'єднання. При цьому оптимальне виконання теоретико-множинних операцій призведе до більш швидкого виконання запиту, що містить хоча б одну таку операцію, та суттєво зменшить час обробки інформації у сучасних базах даних. У статті розглянуто базові, найбільш природні алгоритми, що реалізують теоретико-множинні операції на таблицях та мультитаблицях, та досліджувалися модифікації базових алгоритмів, які могли б зменшити кількість обчислень. У якості критерія оцінювання швидкодії алгоритмів розглядалася їх складність у середньому для найбільш загального випадку, за яким домен кожного атрибуту схеми таблиць є фіксованим і зазделегідь відомим, а розподіл значень за кожним атрибутом в кожній таблиці є рівномірним. Для кожного з шести випадків (три операції на таблицях та три операції на мультитаблицях) знайдено найбільш швидкі за цим критерієм алгоритми. Для усіх розглянутих 6 алгоритмів на таблицях (базові та найшвидші модифікації базових) знайдено точну складність у середньому, причому формули, що визначають складність запропонованих алгоритмів, не містять О-асимптотики. Для експериментального підтвердження теоретичних результатів розроблено програмну систему, яка обчислює фактичну кількість виконаних обчислювальних дій для кожного розглянутих у роботі алгоритму. Проведені експерименти підтвердили теоретичні оцінки, знайдені для таблиць, та визначили найбільш швидкі алгоритми для мультитаблиць. Результати роботи можуть використовуватись як у теорії реляційних баз даних, так і на практиці для оптимізації запитів та зменшення часу обробки інформації в системах управління базами даних.

**MSC:** 68Q25.

***Ключові слова:*** *алгоритм, складність, бази даних.*

## 1. Вступ.

В наш час світ інформаційних технологій широко розповсюдився майже у всі сфери людської діяльності. Переважна більшість сучасних інформаційних систем у своїй роботі використовує бази даних, роботу з якими забезпечують системи управління базами даних, що реалізуються різноманітними програмними продуктами. Наразі спостерігається постійний зріст обсягів інформації, що підвищує вимоги для її зберігання та обробки у реальних базах даних. Однією з цих вимог є збільшення швидкодії роботи системи з користувачем. Для більшості інформаційних систем ця вимога зазвичай реалізується через оптимізацію запитів, які є основним інструментом взаємодії між користувачем і базою даних та й взагалі з системою.

Дотепер найбільш розповсюдженими залишаються реляційні бази даних, ма-

тематична модель яких вперше була запропонована Е. Коддом [1, 2]. Табличні алгебри, що були введені В.Н. Редьком і Д.Б. Буєм [3, 4], побудовані на основі алгебр Е. Кодда, суттєво їх уточнюють та складають теоретичний фундамент мов запитів сучасних табличних баз даних. Аналіз літератури щодо оптимізації операцій у реляційних базах даних показав, що у переважній більшості випадків розглядаються підходи, які дозволяють одержати оптимізацію лише операції з'єднання; не було знайдено жодної спроби оптимізації теоретико-множинних операцій перетину, об'єднання, різниці. Проте ці операції є важливими для табличних (реляційних) алгебр та досить вживаними у запитах. Тому дослідження теоретико-множинних табличних операцій є важливою задачею, результати якої можуть бути використаними для оптимізації запитів.

У роботі досліджуються алгоритми, що реалізують перетин, об'єднання та різницю таблиць і мультитаблиць. У кожному з цих шести випадків спочатку розглядаються базові, найбільш природні алгоритми, що їх реалізують. У процесі дослідження теоретико-множинних табличних операцій нами було розглянуто значну кількість (приблизно 15 для кожного випадку) модифікацій базових алгоритмів, які могли б зменшити кількість обчислень. Для таблиць для кожного алгоритму, що було розглянуто, знайдено значення середньої кількості обчислень. Крім того, нами було розроблено програмну систему, яка для таблиць та мультитаблиць із заданими параметрами (кількість рядків, атрибутів, потужність атрибутів), знаходить фактичну кількість виконаних обчислень за кожним із розглянутих алгоритмів. Завдяки роботі цієї програмної системи були підтверджені теоретичні оцінки складності алгоритмів для таблиць, та обрані найбільш швидкі (з розглянутих) алгоритми для мультитаблиць; у цій роботі наведено ці шість алгоритмів разом з оцінками їх складності.

**2. Основні визначення.**

Зафіксуємо деяку непорожню множину $A = \{A_1, \ldots, A_p\}$, елементи якої називаються атрибутами. Довільну скінченну підмножину $R = \{A'_1, \ldots, A'_n\} \subseteq A$ назвемо схемою. Рядком $s$ схеми $R$ називається множина пар $s = \{(A'_1, d_1), \ldots, (A'_k, d_k)\}$, проекція якої за першою компонентою дорівнює $R$. Таблицею схеми $R$ називається скінченна множина рядків схеми $R$. Кількість рядків у таблиці $T$ позначатимемо $|T|$.

На множині всіх таблиць схеми $R$ введено такі параметричні операції: 1) перетин $\bigcap\limits_R$ таблиць схеми $R$ – таблиця, що складається з тих і лише тих рядків, які належать одночасно всім вихідним таблицям; 2) об'єднання $\bigcup\limits_R$ таблиць схеми $R$ – таблиця, що складається з тих і лише тих рядків, які належать хоча б одній з вихідних таблиць; 3) різниця $T_1 \underset{R}{-} T_2$ двох таблиць схеми $R$ – таблиця, що складається з тих і лише тих рядків, які належать таблиці $T_1$ та не належать таблиці $T_2$.

Іншими словами операції перетину, об'єднання і різниці таблиць є обмеженням відповідно теоретико-множинних операцій перетину, об'єднання і різниці на

множині таблиць однакової схеми.

Існує цілий ряд практичних ситуацій, при яких може виникнути повторюваність даних. При застосуванні традиційних реляційних (табличних) баз даних для обробки результатів у цих ситуаціях можуть виникнути певні ускладнення, оскільки у таблицях повторюванність даних заборонена. Існує декілька шляхів усунення цих ускладнень, одним з яких є розширення поняття таблиці з метою усунення заборони повторюваності рядків. По аналогії з поняттям «мультимножина», такі таблиці дістали назву «мультитаблиці». Отже, мультитаблицею (схеми $R$) називається сукупність рядків однакової схеми ($R$).

Найбільш природними є такі два способи подання мультитаблиці:

а) мультитаблиця задається як невпорядкована сукупність рядків однакової схеми (із можливим повторенням рядків);

б) змінюється структура рядка. Вводиться новий атрибут (у нашій роботі цей атрибут назвемо лічильником, ім'я цього атрибута позначимо $K$), що не належить схемі вихідної таблиці. В якості значення цього атрибута будемо розуміти кількість повторів даного рядка заданої таблиці, це значення є додатним цілим числом.

Нескладно бачити, що ці два способи подання мультитаблиці є еквівалентними: з мультитаблиці, представленої першим способом, легко отримати (причому, однозначно) мультитаблицю, представлену другим способом, і навпаки. На наш погляд, в реальних базах даних природно задавати мультитаблиці першим способом, а для обробки мультитаблиць (у тому числі для виконання операцій над ними) більш зручним є другий спосіб подання. У разі, коли відношення кількості рядків, що зустрічаються точно один раз до загальної кількості рядків у таблиці невелике, другий спосіб подання мультитаблиці є найбільш оптимальним. Далі у нашій роботі вважатимемо, що мультитаблицю задано другим способом подання.

Через $\|T\|$ позначимо кількість унікальних рядків у мультитаблиці $T$. Нехай $s = \{(A_1, d_1), \ldots, (A_n, d_n), (K, k)\}$ – деякий рядок мультитаблиці схеми $R$. Тоді його основою $\|s\|$ назвемо обмеження $s$ за $R$, тобто $\|s\| = \{(A_1, d_1), \ldots, (A_n, d_n)\}$.

При другому способі подання мультитаблиці всі її рядки та всі основи рядків стають попарно різними, тобто вона перетворюється в таблицю схеми $R' = R \bigcup K$ з одним особливим атрибутом – лічильником.

Наведемо означення теоретико-множинних операцій для мультитаблиць. 1) Перетин двох мультитаблиць $T_1$ та $T_2$ – мультитаблиця $T_1 \overset{M}{\underset{R'}{\bigcap}} T_2$, що містить ті і тільки ті рядки, які одночасно містяться в обох вихідних таблицях. 2) Об'єднання двох мультитаблиць $T_1$ та $T_2$ – мультитаблиця $T_1 \overset{M}{\underset{R'}{\bigcup}} T_2$, що містить ті і тільки ті рядки, які містяться хоча б в одній з вихідних таблиць. 3) Різниця двох мультитаблиць $T_1$ та $T_2$ – мультитаблиця $T_1 \overset{M}{\underset{R'}{-}} T_2$, що містить ті і тільки ті рядки, які містяться в $T_1$ і не містяться в $T_2$.

Проілюструємо ці означення. Нехай $\|s\| \in T_1$, $\|s\| \in T_2$ та значення лічильника $s$ в мультитаблицях $T_1$ і $T_2$ дорівнюють відповідно $k_1$ і $k_2$.

Тоді значення лічильника $s$ в мультитаблиці $T_1 \overset{M}{\underset{R'}{\bigcap}} T_2$ дорівнює $\min(k_1, k_2)$, у мультитаблиці $T_1 \overset{M}{\underset{R'}{\bigcup}} T_2$ – $k_1 + k_2$, у мультитаблиці $T_1 \overset{M}{\underset{R'}{-}} T_2$ – $\max(0, k_1 - k_2)$.

При цьому для різниці мультитаблиць, якщо значення лічильника менше одиниці, то відповідний рядок до результуючої мультитаблиці не включається.

В роботі ми будемо порівнювати алгоритми за їх обчислювальною складністю. Слідуючи [5], обчислювальною складністю алгоритму називається функція залежності «обсягу» роботи, яка виконується їм, від «розміру» вхідних даних. Розглядають два види обчислювальної складності: часову, яка оцінює час виконання алгоритму (кількість виконуваних обчислень), і ємнісну, яка оцінює кількість елементів пам'яті, необхідних для роботи алгоритму. У даній роботі будемо досліджувати часову складність та вважаємо, що складність елементарних операцій однакова (тобто використовується так звана рівномірна вага).

Виділяють три типи часової складності алгоритму: складність у найгіршому випадку (максимальна кількість елементарних операцій, що виконує алгоритм), складність для майже всіх входів і складність в середньому (математичне сподівання кількості елементарних операцій, що виконує алгоритм). Дослідження складності алгоритму в найгіршому випадку істотно простіше, ніж дослідження інших двох типів складності, проте при аналізі часової складності обробки інформації в системах керування базами даних та обробки запитів найбільш прийнятним є використання складності в середньому. У нашій роботі швидкодія алгоритмів визначається саме за їхньою складністю в середньому.

**3. Алгоритми, що реалізують теоретико-множинні операції на таблицях.**

У якості базових розглянемо такі алгоритми перетину, об'єднання та різниці таблиць.

**Алгоритм БПТ** З кожним рядком $s_1$ таблиці $T_1$ порівнюємо всі рядки таблиці $T_2$: якщо існує такий рядок $s_2 \in T_2$, що $s_1 = s_2$, то рядок $s_1$ додаємо до таблиці $T$, у якій міститься результат, та переходимо до наступного рядка таблиці $T_1$.

**Алгоритм БОТ** Додаємо всі рядки таблиці $T_1$ до нової таблиці $T$, яка містить результат, а потім всі рядки $s_2$ таблиці $T_2$ порівнюємо з рядками таблиці $T$, і якщо не існує такого рядка $s_1$, що $s_2 = s_1$, то рядок $s_2$ додаємо до таблиці $T$, а якщо такий рядок існує, то переходимо до наступного рядка таблиці $T_2$.

**Алгоритм БРТ** З кожним рядком $s_1$ таблиці $T_1$ порівнюємо всі рядки таблиці $T_2$: якщо не існує такого рядка $s_2 \in T_2$, що $s_1 = s_2$, то рядок $s_1$ додаємо до нової результуючої таблиці $T$, а якщо такий рядок існує, то переходимо до наступного рядка таблиці $T_1$.

У процесі дослідження теоретико-множинних табличних операцій ми розглядали різні модифікації базових алгоритмів, які могли б зменшити кількість обчислень. У якості критерія оцінювання швидкодії алгоритмів розглядалася їх складність у середньому для найбільш загального, на наш погляд, випадку, за яким домен кожного атрибуту схеми таблиць є фіксованим і зазделегідь відомим, та

розподіл значень за кожним атрибутом в кожній таблиці є рівномірним. Іншими словами, для будь-яких атрибуту $A \in R$ та рядка $s$ будь-якої таблиці кожний елемент множини $dom_a = \{d_1, \ldots, d_n\}$ має однакову ймовірність бути значенням $A$ в $s$: $(A, d) \in s \to P\{d = d_1\} = P\{d = d_2\} = \ldots = P\{d = d_n\}$.

Найбільш швидкими виявилися наступні модифікації базових алгоритмів.

**Алгоритм ШПТ** Якщо $|T_1| \leq |T_2|$, то з кожним рядком $s_1$ таблиці $T_1$ порівнюємо всі рядки таблиці $T_2$, і якщо не існує такого рядка $s_2 \in T_2$, що $s_1 = s_2$, то вилучаємо рядок $s_1$ з таблиці $T_1$, а якщо такий рядок $s_2$ існує, то вилучаємо його з таблиці $T_2$ та переходимо до наступного рядка таблиці $T_1$; результат міститься в таблиці $T_1$. Якщо ж $|T_1| > |T_2|$, то таблиці $T_1$ і $T_2$ міняються місцями.

**Алгоритм ШОТ** Якщо $|T_1| \leq |T_2|$, то з кожним рядком $s_2$ таблиці $T_2$ порівнюємо всі рядки таблиці $T_1$, і якщо існує такий рядок $s_1 \in T_1$, що $s_2 = s_1$, то вилучаємо рядок $s_1$ з таблиці $T_1$ і переходимо до наступного рядка таблиці $T_2$; після цього додаємо всі рядки, що залишились, з таблиці $T_1$ до таблиці $T_2$, в якій міститься результат. Якщо ж $|T_1| > |T_2|$, то таблиці $T_1$ і $T_2$ міняються місцями.

**Алгоритм ШРТ** З кожним рядком $s_2$ таблиці $T_2$ порівнюємо всі рядки таблиці $T_1$: якщо існує такий рядок $s_1 \in T_1$, що $s_2 = s_1$, то вилучаємо $s_1$ з таблиці $T_1$, у якій міститься результат, і переходимо до наступного рядка таблиці $T_2$.

Знайдемо складність у середньому алгоритмів БПТ, БОТ, БРТ, ШПТ, ШОТ, ШРТ.

Нехай $|T_1| = m_1, |T_2| = m_2, |A_1| = q_1, \ldots, |A_n| = q_n$. Позначимо $m = \min\{m_1, m_2\}$ та $Q = q_1 \cdot \ldots \cdot q_n$. Існує $Q$ фіксованих рядків, кожний з яких незалежно від решти інших може належати чи не належати кожній з таблиць-аргументів. Нескладно бачити, що у випадку $m_1 + m_2 \leq Q$ мінімальна кількість рядків таблиці $T_1 \bigcap\limits_R T_2$ дорівнює 0, у протилежному випадку мінімальна кількість рядків таблиці $T_1 \bigcap\limits_R T_2$ дорівнює $m_1 + m_2 - Q$, це число позначимо через $z$, максимально можлива кількість рядків таблиці $T_1 \bigcap\limits_R T_2$ становить $m$. Оскільки всі запропоновані алгоритми використовують порівняння рядків в таблицях $T_1$ та $T_2$ на предмет їх рівності, та складність усіх цих алгоритмів залежить від кількості рядків таблиці $T_1 \bigcap\limits_R T_2$, знайдемо значення ймовірності $P(j)$ того, що кількість таких рядків буде дорівнювати $j$ ($j \in \{z, \ldots, m\}$). Далі для кожного алгоритму знайдемо значення $W(j)$ середньої кількості обчислень за умови $|T_1 \bigcap\limits_R T_2| = j$. Будемо вважати складністю алгоритму число

$$W = \sum_{j=z}^{m} P(j) \cdot W(j) (1).$$

Знайдемо ймовірність $P(j)$. Ці $j$ рядків можна обрати $C_Q^j$ способами, рядки таблиці $T_1$, які не належать таблиці $T_2$, можна обрати $C_{Q-j}^{m_1-j}$ способами, а рядки таблиці $T_2$, які не належать таблиці $T_1$, – $C_{Q-m_1}^{m_2-j}$ способами. Існує $C_Q^{m_1}$ варіантів вибору рядків таблиці $T_1$ та $C_Q^{m_2}$ варіантів вибору рядків таблиці $T_2$. Таким чином

імовірність $P(j)$ дорівнює:

$$P(j) = \frac{C_Q^j \cdot C_{Q-j}^{m_1-j} \cdot C_{Q-m_1}^{m_2-j}}{C_Q^{m_1} \cdot C_Q^{m_2}} = \frac{m_1! \cdot m_2! \cdot (Q-m_1)! \cdot (Q-m_2)!}{j! \cdot (m_1-j)! \cdot (m_2-j)! \cdot (Q-m_1-m_2+j)! \cdot Q!}$$

Для знахождения $W(j)$ спочатку знайдемо середню кількість обчислень $W'$, необхідну для порівняння рядків $s^1 = \{(A_1, d_1^1), \ldots, (A_n, d_n^1)\} \in T_1$ та $s^2 = \{(A_1, d_1^2), \ldots, (A_n, d_n^2)\} \in T_2$ у випадку, коли $s^1 \neq s^2$. Імовірність того, що $d_1^1 = d_1^2$, дорівнює $\frac{1}{q_1}$, тому імовірність того, що $d_1^1 \neq d_1^2$, дорівнює $1 - \frac{1}{q_1} = \frac{q_1-1}{q_1}$. Якщо $d_1^1 \neq d_1^2$, то перевірка на рівність рядків $s^1$ та $s^2$ завершується, при цьому виконано одне обчислення. Нехай $d_1^1 = d_1^2$. Тоді імовірність того, що $d_2^1 = d_2^2$, дорівнює $\frac{1}{q_1} \cdot \frac{1}{q_2}$, а ймовірність того, що $d_2^1 \neq d_2^2$, дорівнює $\frac{1}{q_1} \cdot (1 - \frac{1}{q_2}) = \frac{q_2-1}{q_1 \cdot q_2}$. Якщо $d_2^1 \neq d_2^2$, то перевірка на рівність $s^1$ та $s^2$ завершується, при цьому виконано два обчислення.

Аналогічно, якщо виконуються рівності $d_1^1 = d_1^2, \ldots, d_{j-1}^1 = d_{j-1}^2$, то ймовірність того, що $d_j^1 = d_j^2$, дорівнює $\frac{1}{q_1} \cdot \ldots \cdot \frac{1}{q_j}$, а ймовірність того, що $d_j^1 \neq d_j^2$, дорівнює $\frac{q_j-1}{q_1 \cdot \ldots \cdot q_j}$; у другому випадку перевірка на рівність $s^1$ та $s^2$ завершується, при цьому виконано $j$ обчислень. Оскільки за постановкою задачі випадок $s^1 = s^2$ є неможливим (його ймовірність дорівнює $\frac{1}{Q}$), то всі знайдені значення треба поділити на число $1 - \frac{1}{Q} = \frac{Q-1}{Q}$. Таким чином,

$$W' = \frac{Q}{Q-1} \cdot \left(1 \cdot \frac{q_1-1}{q_1} + 2 \cdot \frac{q_2-1}{q_1 \cdot q_2} + \ldots + n \cdot \frac{q_n-1}{Q}\right).$$

При знаходженні $W(j)$ вважаємо, що рядки в таблицях впорядковані блоками наступним чином. В таблиці $T_1$ спочатку розміщено блок з $\frac{m_1-j}{2}$ рядків, що не належать таблиці $T_2$, далі – блок з $j$ рядків таблиці $T_1 \bigcap\limits_R T_2$, після цього – блок з решти $\frac{m_1-j}{2}$ рядків, що не належать таблиці $T_2$. В таблиці $T_2$ впорядкування є аналогічним: спочатку розміщено блок з $\frac{m_2-j}{2}$ рядків, що не належать таблиці $T_1$, далі – блок з $j$ рядків таблиці $T_1 \bigcap\limits_R T_2$, після цього – блок з решти $\frac{m_2-j}{2}$ рядків, що не належать таблиці $T_1$. Оскільки в усіх алгоритмах, що розглянуто у роботі, формули, які підраховують їх складність, відрізняються лише значенням $W(j)$, далі будемо порівнювати швидкодію алгоритмів саме за цим параметром.

Знайдемо значення $W(j)$ для алгоритму БПТ. Для кожного з $m_1 - j$ рядків таблиці $T_1$, що не належать таблиці $T_2$, було виконано $m_2 \cdot W'$ порівнянь. Для $j$ рядків таблиці $T_1$, що також належать таблиці $T_2$, в середньому було виконано $j \cdot \frac{m_2-1}{2} \cdot W'$ порівнянь з тими рядками таблиці $T_2$, що не належать таблиці $T_1$, та $j \cdot n$ порівнянь, які встановили факт рівності рядків; також було виконано $j \cdot n$ додавань рядків до нової таблиці $T$, тому:

$$W(j) = \left((m_1 - j)m_2 + j\frac{m_2-1}{2}\right) W' + 2jn = \left(m_1 m_2 - \frac{j}{2}(m_2+1)\right) W' + 2jn.$$

Знайдемо значення $W(j)$ для алгоритму ШПТ. Для порівняння значень $m_1$ та $m_2$ необхідно одне обчислення. Нехай $m_1 \leq m_2$. Для кожного рядка першого блоку таблиці $T_1$ (їх кількість дорівнює $\frac{m_1-j}{2}$) було виконано $m_2 \cdot W'$ порівнянь та $n$ вилучень з таблиці $T_1$. Для першого рядка другого блоку таблиці $T_1$ в середньому було виконано $\frac{m_2-1}{2} \cdot W' + n$ порівнянь та $n$ вилучень з таблиці $T_2$, для другого рядка другого блоку таблиці $T_1$ – $\frac{m_2-2}{2} \cdot W' + n$ порівнянь та $n$ вилучень (кількість рядків у таблиці $T_2$ після розглядання попереднього рядка зменшилась на одиницю), ..., для $j$-го – $\frac{m_2-j}{2} \cdot W' + n$ порівнянь та $n$ вилучень. Для кожного рядка третього блоку таблиці $T_1$ (їх кількість дорівнює $\frac{m_1-j}{2}$) було виконано $(m_2-j) \cdot W'$ порівнянь та $n$ вилучень з таблиці $T_1$. Тому для випадку $m_1 \leq m_2$:

$$W(j) = 1 + \frac{m_1-j}{2} m_2 W' + \frac{m_1-j}{2} n + \frac{m_2-1}{2} W' + \ldots + \frac{m_2-j}{2} W' + jn + \frac{m_1-j}{2} n +$$

$$+ \frac{m_1-j}{2}(m_2-j)W' = \left( m_1 m_2 - \frac{j}{2}(m_1 + m_2 + \frac{1}{2} - \frac{j}{2}) \right) W' + jn + m_1 n + 1.$$

Для загального випадку:

$$W(j) = \left( m_1 m_2 - \frac{j}{2}(m_1 + m_2 + \frac{1}{2} - \frac{j}{2}) \right) W' + jn + \min\{m_1, m_2\} n + 1.$$

Знайдемо значення $W(j)$ для алгоритму БОТ. При додаванні всіх рядків таблиці $T_1$ до нової таблиці $T$ було виконано $m_1 \cdot n$ обчислень. Далі, для кожного з $m_1 - j$ рядків таблиці $T_1$, що не належать таблиці $T_2$, було виконано $m_2 \cdot W'$ порівнянь. Для $j$ рядків таблиці $T_1$, що також належать і таблиці $T_2$, в середньому було виконано $j \cdot \frac{m_2-1}{2} \cdot W'$ порівнянь з рядками, що не є рівними ним та $j \cdot n$ порівнянь, які встановили факт рівності рядків. При додаванні рядків з таблиці $T_2$ до таблиці $T$ було виконано $(m_2 - j) \cdot n$ додавань, тому:

$$W(j) = m_1 n + \left( (m_1 - j) m_2 + j \frac{m_2 - 1}{2} \right) W' + (m_2 - j) n + jn =$$

$$= \left( m_1 m_2 - \frac{j}{2}(m_2 + 1) \right) W' + m_1 n + m_2 n.$$

Знайдемо значення $W(j)$ для алгоритму ШОТ. Для порівняння значень $m_1$ та $m_2$ необхідно одне обчислення. Нехай $m_1 \leq m_2$. Для кожного рядка першого блоку таблиці $T_2$ (їх кількість дорівнює $\frac{m_2-j}{2}$) було виконано $m_1 \cdot W'$ порівнянь. Далі, аналогічно алгоритму ШПТ, для першого рядка другого блоку таблиці $T_2$ в середньому було виконано $\frac{m_1-1}{2} \cdot W' + n$ порівнянь та $n$ вилучень з таблиці $T_1$, ..., для $j$-го – $\frac{m_1-j}{2} \cdot W' + n$ порівнянь та $n$ вилучень. Далі для кожного рядка третього блоку таблиці $T_2$ (їх кількість дорівнює $\frac{m_2-j}{2}$) було виконано $(m_1-j) \cdot W'$ порівнянь. При додаванні рядків, що лишилися в таблиці $T_1$, до таблиці $T_2$ виконано $(m_1-j) \cdot n$ обчислень, тому для випадку $m_1 \leq m_2$:

$$W(j) = 1 + \frac{m_2-j}{2} m_1 W' + \frac{m_1-j}{2} W' + \ldots + \frac{m_1-j}{2} W' + jn + jn + (m_1 - j)n =$$

$$= \left( m_1 m_2 - \frac{j}{2}(m_1 + m_2 + \frac{1}{2} - \frac{j}{2}) \right) W' + m_1 n + jn + 1.$$

Для загального випадку:

$$W(j) = \left( m_1 m_2 - \frac{j}{2}(m_1 + m_2 + \frac{1}{2} - \frac{j}{2}) \right) W' + \min\{m_1, m_2\} n + jn + 1.$$

Знайдемо значення $W(j)$ для алгоритму БРТ. Для кожного з $(m_1 - j)$ рядків таблиці $T_1$, що не належать таблиці $T_2$, було виконано $m_2 \cdot W'$ порівнянь. Для $j$ рядків таблиці $T_1$, що також належать таблиці $T_2$, в середньому було виконано $j \cdot \frac{m_2 - 1}{2} \cdot W'$ порівнянь з тими рядками таблиці $T_2$, що не належать таблиці $T_1$, та $j \cdot n$ порівнянь, які встановили факт рівності рядків; також було виконано $(m_1 - j) \cdot n$ додавань рядків до нової таблиці $T$, тому:

$$W(j) = \left( (m_1 - j) m_2 + j \frac{m_2 - 1}{2} \right) W' + jn + (m_1 - j) n = \left( m_1 m_2 - \frac{j}{2}(m_2 + 1) \right) W' + m_1 n.$$

Знайдемо значення $W(j)$ для алгоритму ШРТ. Для кожного з $\frac{m_2 - j}{2}$ рядків першого блоку таблиці $T_2$ було виконано $m_1 \cdot W'$ порівнянь. Для першого рядка другого блоку таблиці $T_2$ в середньому було виконано $\frac{m_1 - 1}{2} \cdot W' + n$ порівнянь та $n$ вилучень цього рядка з таблиці $T_1$, ..., для $j$-го – $\frac{m_1 - j}{2} \cdot W' + n$ порівнянь та $n$ вилучень. Далі для кожного з $\frac{m_2 - j}{2}$ рядків третього блоку таблиці $T_2$ було виконано $(m_1 - j) \cdot W'$ порівнянь, тому:

$$W(j) = \frac{m_2 - j}{2} m_1 W' + \frac{m_1 - j}{2} W' + \ldots + \frac{m_1 - j}{2} W' + 2jn + \frac{m_2 - j}{2}(m_1 - j) W' =$$

$$= \left( m_1 m_2 - \frac{j}{2}(m_1 + m_2 + \frac{1}{2} - \frac{j}{2}) \right) W' + 2jn.$$

Оскільки одержані формули складності алгоритмів є досить громізкими, їх аналіз зробимо нижче у шостому розділі цієї роботи.

**4. Алгоритми, що реалізують теоретико-множинні операції на мультитаблицях.**

У якості базових розглянемо такі алгоритми перетину, об'єднання та різниці мультитаблиць.

**Алгоритм БПМ** З кожною основою $\|s_1\|$ кожного рядка мультитаблиці $T_1$ порівнюємо всі основи рядків мультитаблиці $T_2$: якщо існує така $\|s_2\| \in T_2$, що $\|s_1\| = \|s_2\|$, то порівнюємо значення лічильників цих рядків: рядок із мінімальним значенням лічильника додаємо до нової мультитаблиці $T$, у якій міститься результат, та переходимо до наступного рядка мультитаблиці $T_1$.

**Алгоритм БОМ** Додаємо всі рядки мультитаблиці $T_1$ до нової мультитаблиці $T$, яка містить результат, а потім всі основи $\|s_2\|$ рядків мультитаблиці $T_2$ порівнюємо з основами рядків мультитаблиці $T$, і якщо не існує такої $\|s_1\|$, що

$\|s_2\| = \|s_1\|$, то рядок $s_2$ додаємо до мультитаблиці $T$, а якщо така основа існує, то її лічильнику у $T$ присвоюємо значення суми лічильників $\|s_1\|$ і $\|s_2\|$ та переходимо до наступної основи рядка мультитаблиці $T_2$.

**Алгоритм БРМ** З кожною основою $\|s_1\| \in T_1$ порівнюємо всі основи рядків мультитаблиці $T_2$: якщо не існує такої $\|s_2\| \in T_2$, що $\|s_1\| = \|s_2\|$, то рядок $s_1$ додаємо до нової результуючої мультитаблиці $T$, а якщо така основа існує, то у випадку, коли $k_1$ перевищує $k_2$, додаємо $\|s_1\|$ до $T$ та присвоюємо їй нове значення лічильника $k_1 - k_2$; далі переходимо до наступної основи мультитаблиці $T_1$.

Найбільш швидкими виявилися наступні модифікації базових алгоритмів.

**Алгоритм ШПМ** Якщо $\|T_1\| \leq \|T_2\|$, то з кожною $\|s_1\| \in T_1$ порівнюємо всі основи рядків мультитаблиці $T_2$, і якщо не існує такої $\|s_2\| \in T_2$, що $\|s_1\| = \|s_2\|$, то рядок $s_1$ видаляємо з мультитаблиці $T_1$, а якщо така основа існує, то при її виявленні порівнюємо значення лічильників рядків $s_1$ і $s_2$, присвоюємо новому значенню лічильника рядка $s_1$ менше з порівнюваних значень та видаляємо рядок $s_2$ з мультитаблиці $T_2$; далі переходимо до наступної основи рядка $T_1$, у якій міститься результат. Якщо ж $\|T_1\| > \|T_2\|$, то $T_1$ і $T_2$ міняються місцями.

**Алгоритм ШОМ** Якщо $\|T_1\| \leq \|T_2\|$, то з кожною основою $\|s_2\| \in T_2$ порівнюємо всі основи рядків мультитаблиці $T_1$, і якщо існує така $\|s_1\| \in T_1$, що $\|s_2\| = \|s_1\|$, то її лічильнику у $T_2$ присвоюємо значення суми лічильників $\|s_1\|$ і $\|s_2\|$, видаляємо рядок $s_1$ з таблиці $T_1$ і переходимо до наступної основи мультитаблиці $T_2$; після цього додаємо всі рядки, що залишились, з мультитаблиці $T_1$ до мультитаблиці $T_2$, в якій міститься результат. Якщо ж $\|T_1\| > \|T_2\|$, то мультитаблиці $T_1$ і $T_2$ міняються місцями.

**Алгоритм ШРМ** З кожною основою $\|s_2\| \in T_2$ порівнюємо всі основи рядків мультитаблиці $T_1$: якщо існує така $\|s_1\| \in T_1$, що $\|s_2\| = \|s_1\|$, то у випадку, коли $k_1$ перевищує $k_2$, присвоюємо $\|s_1\|$ у $T_1$ нове значення лічильника $k_1 - k_2$, а у випадку $k_1 \leq k_2$ вилучаємо рядок $s_1$ з мультитаблиці $T_1$ та переходимо до наступної основи $T_1$.

**5. Експериментальне порівняння швидкодії алгоритмів.**

Для експериментального підтвердження теоретичних оцінок складності для таблиць та знаходження найшвидших алгоритмів для мультитаблиць було розроблено програмну систему, яка для таблиць із заданими параметрами (кількість рядків, атрибутів, потужність атрибутів), знаходить фактичну кількість виконаних обчислень за кожним із запропонованих алгоритмів та порівнює їх із знайденими теоретичними оцінками (для таблиць), а також знаходить середнє значення фактичної кількості виконаних обчислень для будь-якої серії експериментів. Багаточисленні експерименти підтвердили теоретичні оцінки складності в середньому алгоритмів для таблиць, точність знайдених теоретичних оцінок виявилась достатньо високою, у жодній серії експериментів відхилення не перевищувало 0,25%. У Табл. 1 наведемо дані з двох серій експериментів: перша серія складалася з 500 експериментів з параметрами $n = 10$, $|T_1| = |T_2| = 1000$, $|A_1| = 3, |A_2| = \ldots = |A_{10}| = 2$, друга – з 2000 експериментів з параметрами $n = 5$, $|T_1| = |T_2| = 100$, $|A_1| = |A_2| = |A_3| = 3, |A_4| = |A_5| = 4$.

*Табл 1. Експериментальні значення складності для таблиць.*

| Алгоритм | Кількість обчислень у серії I | | | Кількість обчислень у серії II | | |
|---|---|---|---|---|---|---|
| | теорет. | експерим. | відх. % | теорет. | експерим. | відх. % |
| БПТ | 1132071,9 | 1131741,0 | 0,029 | 13324,34 | 13336,58 | 0,092 |
| ШПТ | 771415,3 | 771135,0 | 0,036 | 12205,89 | 12231,50 | 0,210 |
| БОТ | 1139051,1 | 1139279,6 | 0,020 | 14092,86 | 14094,35 | 0,011 |
| ШОТ | 771415,3 | 771304,5 | 0,014 | 12205,89 | 12212,60 | 0,055 |
| БРТ | 1129051,1 | 1128895,6 | 0,014 | 13592,86 | 13595,13 | 0,017 |
| ШРТ | 767924,7 | 767386,5 | 0,070 | 11820,63 | 11819,86 | 0,007 |

У Табл.1 дані з другого та п'ятого стовпців отримані за допомогою формули (1), дані з третього та шостого стовпців є середнім арифметичним фактичної кількості обчислень, дані з четвертого та сьомого стовпців обчислюються за формулами $\frac{|(2)-(3)|}{(2)} \cdot 100\%$ та $\frac{|(5)-(6)|}{(5)} \cdot 100\%$ відповідно.

У Табл. 2 наведемо дані з двох серій експериментів з мультитаблицями з такими ж самими параметрамии даних, як і для таблиць.

*Табл 2. Експериментальні значення складності для мультитаблиць.*

| Алгоритм | Кількість обчислень у серії I | Кількість обчислень у серії II |
|---|---|---|
| БПМ | 689766,2 | 10822,1 |
| ШПМ | 530267,7 | 9840,3 |
| БОМ | 698374,4 | 11688,1 |
| ШОМ | 530935,3 | 10173,9 |
| БРМ | 691356,5 | 11141,5 |
| ШРМ | 525921,6 | 9733,5 |

З даних Табл.1 та Табл.2 можна зробити висновок, що алгоритми ШПТ, ШОТ, ШРТ, ШПМ, ШОМ, ШРМ за швидкодією суттєво перевищують відповідно алгоритми БПТ, БОТ, БРТ, БПМ, БОМ, БРМ причому це перевищення помітно зростає зі зростанням величин значень параметрів вихідних таблиць.

**6. Висновки.**

У роботі досліджено алгоритми, що реалізують теоретико-множинні операції на таблицях та мультитаблицях. Розглянуто базові, найбільш природні алгоритми, що реалізують ці операції, знайдено модифікації базових алгоритмів, які дозволяють зменшити кількість обчислень. Для усіх розглянутих алгоритмів на таблицях знайдено точну часову складність у середньому, завдяки якій вдалося визначити найшвидші модифікації базових алгоритмів на таблицях. Для експериментального підтвердження теоретичних результатів на таблицях та визначення найшвидших алгоритмів на мультитаблицях розроблено програмну систему, яка обчислює фактичну кількість виконаних обчислень для кожного з запропонованих алгоритмів і порівнює їх із знайденими теоретичними оцінками (для таблиць); проведені експерименти підтвердили знайдені теоретичні оцінки. Результати можуть бути використані для оптимізації запитів у реляційних базах даних та для зменшення часу

обробки інформації у системах управління базами даних. За темою дослідження в подальшому планується дослідити алгоритми, що реалізують інші табличні операції, а також алгоритми, що реалізують теоретико-множинні операції із використанням сортування, індексації, хешування та специфікації таблиць.

**Цитована література**


1. *Codd E.F.* A Relational Model of Data for Large Shared Data Banks // Communications of the ACM. – 1970. – 13, No 6. – P. 377–387.
2. *Codd E.F.* The Relational Model for Database Management: Version 2. – Addison-Wesley, 1990. – 541 p.
3. *Red'ko V.N., Bui D.B.* Foundations of the theory of relational database models // Cybernetics and Systems Analysis. – 1996. – Vol. 32, Iss. 4. – P. 471–478.
4. *Редько В.Н., Брона Ю.Й., Буй Д.Б., Поляков С.А.* Реляційні бази даних: табличні алгебри та SQL-подібні мови. – Київ: «Академперіодика», 2001. – 198 с.
5. *Буй Д.Б., Скобелев В.Г.* Сложность операций в базах данных (обзор) // Радіоелектронні і комп'ютерні системи. – 2014. – №. 6. – С. 53–59.


**References**


1. Codd, E.F. (1970). A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM, 13*(6), 377-387.
2. Codd, E.F. (1990). *The Relational Model for Database Management: Version 2.* Addison-Wesley.
3. Red'ko, V.N., Bui, D.B. (1996). Foundations of the theory of relational database models. *Cybernetics and Systems Analysis, 32*(4), 471-478.
4. Red'ko, V.N., Brona, Yu.Y., Bui, D.B., Polyakov, S.A. (2001). *Relational databases: table algebras and SQL-like languages.* Kyiv: Academperiodica (in Ukrainian).
5. Bui, D.B., Skobelev, V.G. (2014). Complexity of operations in database systems (a survey). *Radioelectronic and Computer Systems, 6*, 53-59 (in Russian).


**I.S. Kanarskaya**

**Algorithms, that implement set-theoretic operations on tables and multitables.**

The paper is devoted to the research of algorithms implementing intersection, union and difference in tables and multitables. The subject of the work is relevant, since despite the importance and applicability of set-theoretical operations in relational databases, for some reason, the attention of researchers was focused on optimizing other table operations, first of all, the join. Meanwhile, the optimal implementation of set-theoretical operations will lead to a faster execution of the query, which containing at least one of set-theoretical operations, and will significantly reduce the time of processing information in the database management systems. For each set-theoretical operation algorithms that implement them on tables, in which strings are not repeat, and on multi-tables, in which the strings can be repeated, are considered. After that the modifications of the basic algorithms, that we found, which allow to significantly reduce the number of computations are considered. As an average case, we understand the most general case in which the domain of each attribute of the table schema is fixed and known above, and the distribution of values for each attribute in each table is uniform. For each of the six cases (three table operations and three multi-table operations), the fastest algorithms by this criterion were found. For all 6 algorithms considered on the tables (basic and fastest modifications of the basic ones) we found exact complexity on average. The found formulas defining the complexity

of the proposed algorithms do not contain O-asymptotics. For the experimental confirmation of the results we developed the software system, which, for tables with given parameters, finds the actual number of computations performed for each of the proposed algorithms. The experiments carried out confirmed the theoretical estimates found for the tables and identified the fastest algorithms for the multitables. The results of the work can be used both in relational databases theory and in practice in queries optimization and to reduce the processing time in database management systems.

***Keywords:*** *algorithm, complexity, databases.*

*ВП «Слов'янський коледж Луганського національного аграрного університету», Слов'янськ*
*iren_kanar@ukr.net*

*Отримано 02.09.19*