УДК 004.891

д.т.н., проф. **Лєнков С.В.** (ВІКНУ
к.т.н., доц. **Джулій В.М.** (ХмНУ)
к.т.н., доц. **Осипа В.О.** (ВІКНУ)
Хлистун І.А. (ХмНУ)

ІДЕНТИФІКАЦІЇ ОБ'ЄКТІВ В СЛАБОСТРУКТУРОВАНІЙ БАЗІ ДАНИХ

У статті розглянуті проблеми ідентифікації об'єктів в слабоструктурованій базі даних, а також застосування результатів порівняльного аналізу для вирішення алгоритмів їх пошуку. Описаний алгоритм усунення дублювання записів в базі даних при наявності декількох джерел інформації і помилок операторського введення. Запропоновано алгоритм обчислення функції релевантності. Нечіткий пошук доцільно застосовувати при ідентифікації слів з друкарськими помилками, а також у тих випадках, коли виникають сумніви в правильному написанні персональних даних. Використані при реалізації нечіткого пошуку алгоритми засновані на особливій системі асоціативного доступу до слів, що містяться в текстовому індексі повнотекстового сховища документів. В якості одиниць пошуку використовуються ланцюжки букв, що складають слово. Для прискорення пошуку попередньо створюється спеціальний індекс, що містить фрагменти слів з посиланнями на слова, в яких ці фрагменти зустрілися. Алгоритм нечіткого пошуку дозволяє швидко відібрати всі слова, фрагменти яких співпадають з фрагментами слова в запиті, що лежать в заданому діапазоні допустимих спотворень. Алгоритм дозволяє: зберегти інформаційну цілісність, а також знизити зашумленість даних; виробляти об'єднання записів, в яких відсоток схожості по заданому набору полів вище встановленої межі; виробляти усунення дублювань як на підставі автоматично налаштованих правил, так і з втручанням людини в особливо складних випадках.

Ключові слова: база даних, нечіткий пошук, порівняння рядків, пошук даних, алгоритми, інформаційна система.

Вступ. На даний момент системи керування базами даних широко використовуються в організації сучасних інструментальних, промислових, аналітичних та інформаційних систем. Розвиток інформаційних технологій баз даних визначає також ряд нових проблем та напрями подальших досліджень у даній сфері. Програмне забезпечення на сьогоднішній день розвивається в умовах швидкого нарощування обчислювальних потужностей, апаратних можливостей, швидкості доступу до пам'яті, обсягу пам'яті, пропускної здатності та надійності каналів передачі даних. Все більшого значення набувають засоби, що забезпечують взаємодію в розподіленій системі функціонування інформаційних систем.

Ефективність управління сучасним бізнесом заснована на можливості отримання управлінським персоналом всебічної інформації з усіх напрямків діяльності компанії. При цьому важливим є встановлення контролю над зростаючими потоками інформації, прискорення процесу їх обробки, узагальнення та аналізу. Необхідність постійно забезпечувати всіх учасників процесу управління достовірною, цілісною, несуперечливою і актуальною інформацією визначає ключове завдання сьогоднішнього дня в області підвищення ефективності управління - впровадження сучасних інформаційних технологій в систему управління підприємством. Для багатьох компаній інформація є основним активом. Втрата важливої інформації може призвести до суттєвих фінансових втрат або під загрозою опиняється весь бізнес. Керувати інформацією та забезпечувати її збереження в зазначеному вигляді є вкрай непростим завданням. Для вирішення проблем, пов'язаних з організацією зберігання і централізованого управління великими обсягами різнорідних даних необхідне комплексне рішення. Впровадження даного рішення за рахунок використання сучасних інформаційних технологій дозволить не тільки підвищити якість вирішення завдання збереження інформації, а й розширять можливості використання інформації за рахунок появи нових функцій, таких, як прискорений пошук, розмежування доступу співробітників до даних, управління життєвим циклом інформації та інше. Управління дисковим простором, збільшення ємності систем зберігання і зростання їх продуктивності, міграція даних із

застарілих сховищ на нові - всі ці, і багато інших завдань доводиться постійно вирішувати компаніям, що використовують системи зберігання даних.

Постановка задачі. Сучасний підхід до операцій з даними дозволяє змінити існуючий стан речей. Багато типів даних можна розглядати не як набір чисел і символів, якими оперують обчислювальні системи, а у вигляді об'єктів, з якими оперує користувач: фінансові документи, поштові повідомлення, технічна документація, фотографії, відеоролики, звукові записи, скановані документи і т.д. У подібній ситуації значно зросла необхідність у створенні та впровадженні ефективних систем пошуку та аналізу даних. Інтегровані в СКБД системи пошуку слабо адаптовані для обробки неструктурованої інформації. За статистикою, частка структурованих даних в сучасних базах даних становить не більше 35-50%, решта ж припадають на частку різних довідників, сканованих документів та іншої розрізненої інформації. У цьому випадку виникає проблема пошуку і вибірки необхідної інформації з великого неструктурованого масиву. При організації пошуку даних в подібних базах виникають характерні проблеми, пов'язані з наявністю в запитах орфографічних і фонетичних помилок, помилок введення інформації, а також відсутністю єдиних стандартів транскрипції з іноземних мов. Внаслідок зазначених причин завдання пошуку в базах даних не може бути повною мірою вирішеним тільки методами перевірки на точну відповідність. Стає актуальною задача розробки спеціальних методів і технологій текстового пошуку з використанням нетривіальних рішень, у тому числі з використанням апарату нечіткої логіки (fuzzy logic), а також алгоритмів нечіткого пошуку. Нечіткий пошук доцільно застосовувати при ідентифікації слів з друкарськими помилками, а також у тих випадках, коли виникають сумніви в правильному написанні персональних даних. Використані при реалізації нечіткого пошуку алгоритми засновані на особливій системі асоціативного доступу до слів, що містяться в текстовому індексі повнотекстового сховища документів. В якості одиниць пошуку використовуються ланцюжки букв, що складають слово. Для прискорення пошуку попередньо створюється спеціальний індекс, що містить фрагменти слів з посиланнями на слова, в яких ці фрагменти зустрілися. Алгоритм нечіткого пошуку дозволяє швидко відібрати всі слова, фрагменти яких збігаються з фрагментами слова в запиті, що лежать в заданому діапазоні допустимих спотворень. Задаючи розмір діапазону (відсоток відмінних фрагментів і допустимі зміщення їх позицій в слові), можна легко регулювати точність і повноту пошуку - відбирати слова за ступенем близькості до запиту. Швидкість пошуку пропорційна логарифму від числа індексованих слів і становить менше однієї секунди при індексі в кілька мільйонів слів. За допомогою нечіткого пошуку можливе вирішення наступних прикладних задач: повна ідентифікація суб'єкта або об'єкта при наявності спотворень інформації в базі даних або в пошукових запитах; усунення дублікатів записів при надходженні до БД з множинних джерел зі слабоструктурованою інформацією; пошук і коректування помилок у персональних даних (фізичних та юридичних осіб), адресних даних, телефонних номерах, текстових примітках та ін. З найбільш ефективних алгоритмів слід виділити алгоритми n-грам, trie-дерев, а також сигнатурні алгоритми, які забезпечують оптимальне співвідношення між розміром індексу і швидкістю пошуку.

Основна частина. Функція релевантності реалізована на основі алгоритму порівняння підрядків і визначає близькість рядкових значень у відсотках. Аргументами функції є два рядки і параметр порівняння (N), що представляє із себе максимальну довжину підрядків, що беруть участь в порівнянні. Як результат функція повертає відсоток релевантності, де 0% вказує на абсолютну розбіжність рядків, а 100% на тотожну рівність. Порівняння відбувається за наступним алгоритмом: функція створює два набори підрядків (довжина підрядків обмежується параметром порівняння). Для підрядків однакової довжини в двох наборах функція знаходить підрядки першого рядка, які є в підрядку другої, додає кількість співпадань підрядків другого рядка з підрядками першого. Відношення суми співпадань до числа варіантів, приведене до процентного виду, запам'ятовується як проміжний коефіцієнт релевантності для даної довжини підрядків, далі береться середнє значення всіх проміжних коефіцієнтів і повертається функцією як відсоток релевантності вхідних рядків.

Функцію релевантності від двох рядків S_{t1} і S_{t2} довжиною l_1 і l_2 відповідно і максимальної довжини підрядків N визначимо так:

1. Формуємо набори всіх можливих підрядків довжиною до N:

$$G_j(i)=\{g_{j1}(i),...,g_{jk}(i),...,g_{jn}(i)\};\quad j=1,2;\quad i=\overline{1,N};\quad n=l_j-i+1; \tag{1}$$

де i- довжина підрядка; j - номер вхідного рядка; n - кількість підрядків довжиною i в j-му слові.

2. Кожному набору $G_j(i)$ поставимо у відповідність множину $G_j^*(i)$, елементи яких не повторюються із набору $G_j(i)$, тобто повторюваним елементам набору $G_j(i)$ в множині $G_j^*(i)$ буде відповідати один елемент:

$$G_j^*(i)=\{g_{j1}^*(i),...,g_{jk}^*(i),...,g_{jm}^*(i)\};\quad j=1,2;\quad i=\overline{1,N};\quad m\le l_j-i+1; \tag{2}$$

де m - кількість неповторюваних підрядків довжиною i в j-му слові.

3. Значення функції релевантності $FR=(l_1,l_2,N)$ обчислюється за наступною формулою:

$$FR=(l_1,l_2,N)=\frac{\sum_{i=1}^{N} fr(i)}{N}, \tag{3}$$

$$fr(i)=\frac{|{}^*G_1(i)|+|{}^*G_2(i)|}{|G_1(i)|+|G_2(i)|} \tag{4}$$

$${}^*G_j(i)=G_j(i)\cap G_j^*(i), \tag{5}$$

де $g_j(i)\in {}^*G_j(i)\Rightarrow \exists\, g_k^*(i): g_j(i)=g_k^*(i)$ тобто набір ${}^*G_j(i)$ складається з елементів набору $G_j(i)$, для яких є рівні у множині $G_j^*(i)$. $G_j(i)$ - набір підрядків довжиною i рядка l_j; $|G_j(i)|$ - кількість елементів у наборі підрядків $G_j(i)$; $G_j^*(i)$ - множина, в якій не повторюються підрядки набору $G_j(i)$; $|G_j^*(i)|$ - кількість елементів у наборі підрядків $G_j^*(i)$; $|{}^*G_j(i)|$ - кількість елементів у наборі підрядків ${}^*G_j(i)$; N - максимальна довжина підрядка.

Алгоритм пошуку та усунення дублювання рядків в БД представлений на рис.1.

Загальний принцип роботи алгоритму пошуку та усунення дублювання в БД наступний:

1. На вхід надходить масив даних, який потрібно додати в БД, з умовою виключення дублювання даних.

2. Проводиться обчислення значення функції релевантності кожного рядка вхідного масиву з кожним рядком БД.

3. Якщо значення функції: - вище межі автоматичної ідентифікації (Γ_{ai}), після якої кількість розпізнаних дублювань стає практично рівним 100%, то відповідний вхідний рядок оголошується повторюваним; - нижче межі ручної ідентифікації (Γ_{pi}), то рядки, від яких обчислюється функція, оголошуються різними і аналіз триває; - вище (Γ_{pi}), але нижче (Γ_{ai}), то такі рядки відправляються в лог прийняття рішень для обробки аналітиком.

4. Якщо у якому-небудь рядку вхідного масиву всі значення функції нижче (Γ_{pi}), то даний рядок оголошується новим і додається в БД.

Основними способами внесення та зміни інформації в базу даних є: безпосереднє введення користувачами; імпорт даних із зовнішніх джерел. При введенні інформації користувачем необхідно забезпечити мінімальний час роботи системи, тому що на цьому етапі алгоритм повинен працювати гранично швидко за рахунок зменшення точності. При цьому межі (Γ_{ai}) і (Γ_{pi}) можуть бути змінені в допустимих межах для забезпечення потрібної швидкості пошуку. Так як алгоритми пошуку все-таки не можуть гарантувати 100% точність, користувач має можливість залишити підказку системи без уваги і підтвердити введення даних. Таким чином, у вирішенні завдання виявлення дублювання можна виділити три етапи: виявлення дублювань на рівні введення інформації користувачами та їх відхилення; виявлення дублювань шляхом порівняння і аналізу - уже введених даних відповідно до заданого (Γ_{ai}) і автоматичне видалення дублюючої інформації; аналіз та обробка користувачем результатів, які не можуть бути оброблені автоматично.

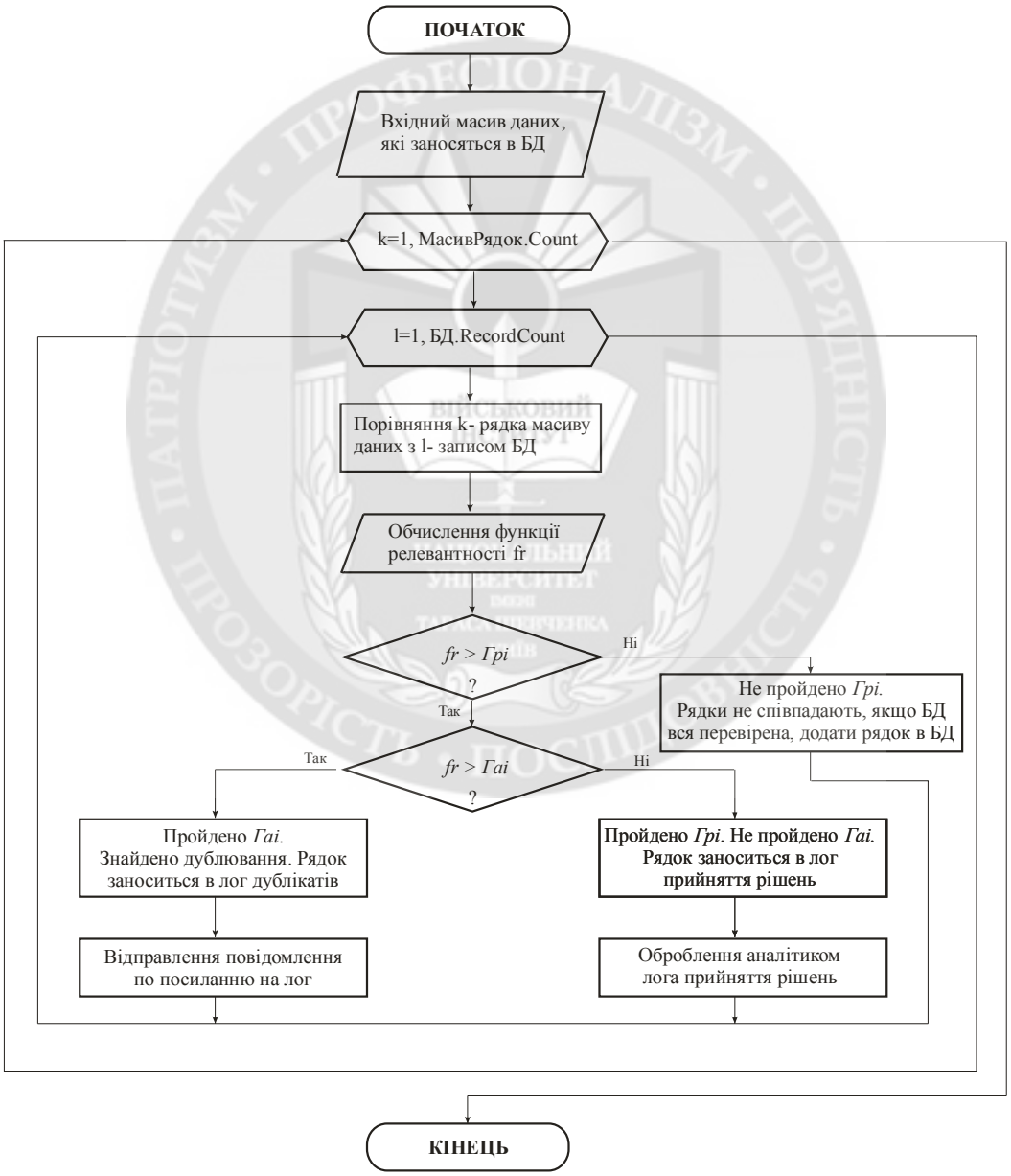

Рис. 1. Алгоритм пошуку та усунення дублювання в БД

Висновки. Запропонований алгоритм дозволяє: зберегти інформаційну цілісність, а також знизити зашумленість даних, обумовлену наявністю помилок операторського

введення; виробляти об'єднання записів, в яких відсоток схожості по заданому набору полів вище встановленої межі; виробляти усунення дублювань як на підставі автоматично налаштованих правил (автоматичний режим), так і з втручанням людини в особливо складних випадках (ручний режим).

ЛІТЕРАТУРА:

1. Ахо А. Структуры данных и алгоритмы./ А. Ахо, Д. Хопкрофт, Д.Ульман. – М.: Вильяме, 2009. – 400 с.

2. Вирт Н. Алгоритмы и структуры даннях / Н. Вирт. – М.: ДМК Пресс, 2010. –272 с.

3. Гагарина Л.Г. Разработка и эксплуатация автоматизированных информационных систем : учеб. пособие / Л.Г. Гагарина, Д.В. Киселев, Е.Л.Федотова. – М.: ИД «Форум»: Инфа-М, 2007. – 384 с.

4. Гагарина Л.Г. Алгоритмы и структуры данных / Л. Г. Гагарина, В.Д. Колдаев. – М.:Инфра-М, 2009. – 304 с.

5. Гайдамакин Н.А. Автоматизированные информационные системы, базы и банки данных./ Н.А. Гайдамакин // Москва «Гелиос АРВ», 2002. – 368 с.

6. Кнут Д.Э. Искусство программирования / Кнут Д.Э. // Том 4. Выпуск 2. Генерация всех кортежей и перестановок. – Ml:: Вильяме, 2008. – 160 с.

7. Макленнен Д. Microsoft SQL Server 2008 / Макленнен Д., Танг Ч., Криват Б.// Data Mining - интеллектуальный анализ данных. – СПб.: БХВ-Петербург, 2010. – 700 с.

REFERENCES:

1 Aho Data Structures and Algorithms. / A. Aho, J. Hopcroft, D..Ulman // - M .: Williams, 2009, - 400.

2.Virt N. Structures and Algorithms dannyah / Wirth // - M .: DMK Press, 2010 - 272 p.

3. Gagarin LG / design and operation of automated information systems / LG Gagarin, DV, Kiselev E.L.Fedotova //: Proc. allowance. M .: ID "Forum": Infa-M, 2007. 384 p.

4. Gagarin LG Algorithms and Data Structures. /L.G. Gagarin,VD Koldaev// - Moscow: Infra-M, 2009.- 304.

5. Gaydamakin NA Automated information systems, databases and data banks. / NA Gaydamakin // Moscow "Helios ARV" 2002. 368 p.

6. DE Knuth Art of Computer Programming. / DE Knuth // Volume 4 Issue 2 Generating all tuples and permutations. - Ml :: Williams, 2008 - 160 p.

7. Macclenny D. Microsoft SQL Server 2008 / Macclenny, D., Tang, C. Krivat B .// Data Mining - Data Mining - St. Petersburg .: BHV-Petersburg, 2010 - 700 p.

Без рецензії.

д.т.н., проф. Ленков С.В., к.т.н., доц. Джулий В.М., к.т.н., доц. Осыпа В.А., Хлистун И.А.

ИДЕНТИФИКАЦИИ ОБЪЕКТОВ В СЛАБОСТРУКТУРИРОВАННОЙ БАЗЕ ДАННЫХ

В статье рассмотрены проблемы идентификации объектов в слабоструктурированной базе данных, а также применение результатов сравнительного анализа для решения алгоритмов их поиска. Описанный алгоритм устранения дублирования записей в базе данных при наличии нескольких источников информации и ошибок операторского ввода. Предложен алгоритм вычисления функции релевантности. Нечеткий поиск целесообразно применять при идентификации слов с опечатками, а также в тех случаях, когда возникают сомнения в правильном написании персональных данных. Использованы при реализации нечеткого поиска алгоритмы основаны на особой системе ассоциативного доступа к словам, содержащиеся в текстовом индексе полнотекстового хранилища документов. В качестве единиц поиска используются цепочки букв, составляющих слово. Для ускорения поиска предварительно создается специальный индекс, содержащий фрагменты слов со ссылками на слова, в которых эти фрагменты встретились. Алгоритм нечеткого поиска позволяет быстро отобрать все слова, фрагменты которых совпадают с фрагментами слова в запросе, лежащие в заданном диапазоне допустимых искажений. Алгоритм позволяет сохранить информационную целостность, а также снизить зашумленность данных; производить объединение записей, в

которых процент сходства по заданному набору полей выше установленного предела; производить устранение дублирования как на основании автоматически настроенных правил, так и с вмешательством человека в особо сложных случаях.

Ключевые слова: база данных, нечеткий поиск, сравнение строк, поиск данных, алгоритмы, информационная система.

Prof. Lenkov S.V., Ph.D. Giulio V.M., Ph.D. Ossypa V.O., Hlistun I.A.

IDENTIFICATION OF OBJECTS IN DATABASE SEMISTRUCTURED

The article deals with the problem of identification of objects in semistructured database, and application of the results of a comparative analysis of solutions for their search algorithms. The described algorithm to eliminate duplication of records in the database when there are multiple sources of information and operator input errors. An algorithm for calculating the relevance function. Fuzzy search is advisable to apply for identification of misspellings, as well as in cases where there are doubts about the correct spelling of personal data. Used to implement fuzzy search algorithms are based on a special system of associative access to the words contained in the text index, full-text repository of documents. As the search units used letters of the chain that make up the word. Special pre-created index for faster search, containing fragments of words with reference to the words in which these fragments were met. fuzzy search algorithm allows you to quickly select all of the words, fragments of which spyvpadayut with fragments of words in the query, lying in a predetermined range of acceptable distortion. The algorithm preserves the integrity of the information, as well as reduce noise pollution data; make association records in which the percentage of similarity for a given set of fields above the limit; perform deduplication on the grounds of automatically configured rules and with human intervention in particularly complex cases.

Keywords: database, fuzzy searches, a string comparison, data mining, algorithms, information system.