

РОЗДІЛ 5

ІНТЕГРОВАНІ ТЕХНОЛОГІЇ В ГАЛУЗІ ІНТЕЛЕКТУАЛЬНИХ ОБЧИСЛЕНЬ

Комп'ютерні технології разом з організацією інтелектуальних обчислень переживають свій розквіт. Це пов'язано, головним чином, з потоком нових ідей, породжених дослідженнями у порівняно новій галузі комп'ютерних наук, яка утворилася на перетині штучного інтелекту, статистики та теорії баз даних. Зараз відбувається стрімке зростання кількості програмних продуктів, що використовують нові технології, а також типів завдань, де їх застосування приносить значний економічний ефект. Елементи автоматичної обробки й аналізу даних стають невід'ємною частиною концепції електронних сховищ даних та організації інтелектуальних обчислень [1-3].

Системи, побудовані шляхом поєднання баз даних (БД) і нечіткої логіки, дозволяють істотно розширити функціональні можливості та коло задач, що розв'язуються. У той же час, нечітка база даних розглядається як узагальнена версія традиційної бази даних, інакше кажучи, остання є спеціальним випадком нечіткої бази даних.

Проблема проектування БД зараз є об'єктом зростаючого інтересу широкого кола фахівців у різних галузях науки і техніки. Проте, у багатьох випадках інформація, що зберігається у БД, не завжди точно відображає характеристики досліджуваного об'єкта. Це пояснюється низкою об'єктивних і суб'єктивних причин. По-перше, даний об'єкт може мати складну функціональну структуру, і дослідники не в змозі повністю її з'ясувати. По-друге, через різні об'єктивні причини фахівці часто-густо не мають можливості отримати точну інформацію, а отримують лише деяку її частину. По-третє, на практиці зустрічається досить багато атрибутів об'єкта, які можна оцінити лише якісними величинами, а не кількісними. Такі типи даних часто зустрічаються в експертних системах, системах прийняття рішень та

інтелектуальних БД. Тому сьогодні проблема проектування нечіткої моделі БД і створення методик обробки неточної та абстрактної інформації стає все більш актуальною [4].

Багато вчених розглядали проблеми проектування нечітких баз даних, особливо на основі реляційної моделі. Розроблено різні варіанти нечітких моделей реляційних баз даних. Результати досліджень викладені, наприклад, у роботах: Buckles B.P., Petty F.E., Shenoі S., Melton A., Vila M.A., Lipski W Jr., Prade H., Zemankova-Leech M., Kandel A., Umано M., Ding-An Chiang, Lui R. Chow, Nan-Chen Hsien, Le Tien Vuong, Ho Thuan, Truong Due Hung та інших. Серед них нечіткі моделі Buckles B.P. і Petty F.E., Shenoі S. і Melton A.

Завдяки роботам в області баз даних таких вчених як Мартін Дж., Мейер Д., Дейт К.Дж, Codd E.F., Fagin R., Armstrong W.W. та ін., теорія класичних баз даних була достатньо розвинена з математичної точки зору [5-7]. У той же час теорія нечітких баз даних ще не може вважатися закінченою, і в ній залишається чимало питань, що потребують вирішення, та обумовлює актуальність сучасних досліджень.

5.1 Реляційні бази даних у завданнях видобування знань

Зараз відбувається стрімке зростання кількості програмних продуктів, що використовують нові технології, а також типів завдань, де їх застосування забезпечує значний економічний ефект. Елементи автоматичної обробки і аналізу даних, що називають Data Mining (видобування знань), стають невід'ємною частиною концепції баз даних, електронних сховищ даних та організації інтелектуальних обчислень.

Але традиційна математична статистика, яка довгий час претендувала на роль основного інструмента аналізу даних, пасує при розв'язуванні завдань із реального складного життя. Вона оперує усередненими характеристиками вибірки, що часто є фіктивними величинами (типу середньої температури

пацієнтів по лікарні). Тому методи математичної статистики виявляються корисними, головним чином, для перевірки заздалегідь сформульованих гіпотез. Data Mining все більш стає мультидисциплінарною галуззю, що виникла на підґрунті досягнень різних наук. Звідси і велика кількість методів та алгоритмів, що реалізовані в різних діючих системах Data Mining. Багато із таких систем інтегрують у собі кілька підходів. Проте, у кожній системі існує, як правило, такий ключовий елемент, на якому фокусується головна увага [8].

Системи, побудовані шляхом поєднання баз даних і нечіткої логіки, дають змогу істотно розширити функціональні можливості та коло завдань, що розв'язуються. Теорія нечітких баз даних ще не повністю сформована, і в ній залишається чимало питань, що потребують вирішення.

Застосування агрегатних функцій щодо ключових атрибутів дає можливість будувати логічні залежності між інформаційними одиницями. Введено поняття функціональних асоціативних правил. Семантична мережа (СМ), що побудована на основі запропонованого підходу, дозволяє підвищити ефективність систем підтримки прийняття рішень.

При переході до знань у пам'ять повинна вводиться інформація про деяку структуру інформаційних одиниць. Наприклад, машинне слово може представляти структуру, де вказано, в яких розрядах зберігається інформація про групи і спеціальності наукових працівників. При цьому повинні бути завдані спеціальні словники: «довідник груп» і «довідник спеціальностей», в яких перераховані наявні в пам'яті ІС групи та спеціальності. Ці атрибути відіграють роль імен для тих машинних слів, які відповідають рядкам таблиці. За ними можна здійснювати пошук потрібної інформації. Кожен рядок таблиці є екземпляром структури.

Зараз практично всі системи управління базами даних забезпечують внутрішню інтерпретованість всіх інформаційних одиниць, що зберігаються в БД. З іншого боку, існує низка питань, які потребують додаткових досліджень. Серед них можна виділити питання, пов'язані із впливом структурних

властивостей БД на формування знань. Подібні завдання розглядаються в роботі R. Srikant, де для обробки транзакцій, що складаються з різних типів даних, та вилучення численних асоціативних правил, був запропонований відповідний алгоритм. У роботах Л.А.Калиниченка і Е.М. Беніамінова міститься опис алгебраїчних засобів моделювання БД реляційного типу, подання знань і понять [9].

Сучасні технології обробки інформації, в яких переважає розділення інформаційних одиниць на дані та команди, створили ситуацію, при якій дані пасивні, а команди активні. Тобто, всі процеси, що протікають, ініціюються командами, а дані використовуються цими командами лише у разі потреби.

Метою проведених досліджень є аналіз особливостей інформаційних одиниць і структур даних, які впливають на технологію вилучення знань. БД можуть бути розглянуті у вигляді структур бази знань (БЗ), оскільки у даний час не існує БЗ, в яких повною мірою були б реалізовані внутрішня інтерпретованість, структуризація, зв'язність, введена семантична міра і забезпечена активність знань.

Крім однозначної можливості інтерпретації, інформаційні одиниці повинні мати гнучку структуру. Кожна інформаційна одиниця може бути включена до складу будь-якої іншої, а із кожної інформаційної одиниці можна виділити деякі складові – її інформаційні одиниці. Інакше кажучи, повинна існувати можливість встановлення між окремими інформаційними одиницями відношень типу «частина – ціле», «елемент – клас» та ін.

У БД зв'язки, встановлені між інформаційними одиницями, можуть характеризувати відповідні відношення між ними. Семантика відношень може носити або декларативний, або процедурний характер. Наприклад, дві або більше інформаційні одиниці можуть бути зв'язані відношенням «у деякий час», «причина – наслідок», «бути однаковим» та ін. Такі відношення характеризують декларативні знання. Якщо між двома інформаційними одиницями встановлене відношення «аргумент – функція», то воно

характеризує процедурне знання, пов'язане з обчисленням певних функцій.

Агрегатними функціями називаються функції, які визначають кількість записів у таблиці, підраховують у стовпці кількість значень або знаходять для нього мінімальне й максимальне значення, а також підсумують дані. До агрегатних функцій належать функції COUNT, SUM, MAX, MIN, AVG і, можливо, інші, що можуть пропонуватися розробником конкретної системи.

Для застосування агрегатних функцій при обчисленнях деякої групи однакових значень використовується параметр угруповання GROUP BY. Цей параметр «стискає» однакові значення заданого атрибута в один рядок підсумкових результатів. Наприклад, для пошуку середньої ціни деталі можна сформулювати запит мовою SQL.

`SELECT Num_detail, AVG(Price) FROM Table GROUP BY Num_detail`

Схема використання агрегатної функції AVG щодо згрупованих даних показана на рис.5.1.

Num_detail	Colors	Price
103	Green	105 грн.
103	Blue	25 грн.
103	Wait	46 грн.
106	Black	59 грн.
106	Green	102 грн.
201	Wait	15 грн.
201	Black	28 грн.
201	Blue	115 грн.

Num_detail	AVG (Price)
103	58.7 грн.
106	80.5 грн.
201	52.7 грн.

Рисунок 5.1 – Запит з угрупованням

Розробка й дослідження моделі функціональних асоціативних правил здійснюється таким чином. Нехай є БД, для доступу до якої реалізовано набір транзакцій $T = \{T_1, \dots, T_n\}$. $D = \{d_1, \dots, d_n\}$ – множина елементів з T , тобто T_i

$\subseteq D$ і $\Omega = \{\text{COUNT, SUM, MAX, MIN, AVG, ...}\}$ – набір агрегатних функцій. Кожна транзакція є бінарним вектором, де $T_i = 1$, якщо елемент d_i є присутнім у транзакції, й $T_i = 0$, якщо ні. Транзакція T_i містить набір елементів $X \subseteq D$, якщо $X \subset T_i$. Тоді продукцію виду (1) будемо називати функціональним асоціативним правилом, якщо $X \subset D$, $Y \subset D$, $X \cap Y = \emptyset$ і $\varpi \in \Omega$.

$$\{P; X \Rightarrow \varpi (Y)\}, \quad (5.1)$$

де P – умова активізації ядра правила.

Метою аналізу БД є встановлення таких залежностей: якщо в транзакції зустрівся деякий набір елементів X , то на підставі цього можна зробити висновок про те, що інший набір елементів Y також повинен з'явитися в цій транзакції. Встановлення таких залежностей дає можливість знаходити прості й інтуїтивно зрозумілі правила.

У загальному вигляді процес формування асоціативних правил можна подати двома етапами:

- виділення всіх необхідних наборів елементів;
- генерація правил з наборів елементів із застосуванням необхідних функцій.

Звернімо увагу на умову активізації правила (5.1). Для реалізації секвенції « \Rightarrow » цього правила необхідне виконання умови застосовності ядра. У теорії реляційних БД відсутність значення в атрибуті є неприпустимою. З іншого боку, якщо схема БД містить кілька зв'язаних відношень, то можлива ситуація, коли значення зв'язаного атрибута на даний момент не визначено.

Нехай $\mathfrak{R}(R_1(a,b,c), R_2(c,d), R_3(d,e))$ – реляційна БД (підкреслені атрибути є ключами), у схемі якої визначені зв'язки $R_1 \xrightarrow{1:M} R_2$, $R_2 \xrightarrow{1:M} R_3$ і нехай для \mathfrak{R} задана транзакція:

BEGIN TRANSATION

SELECT R₁.a, R₁.b, SUM(R₁.d) FROM R₁ GROUP BY R₁.a, R₁.b;

SELECT R₁.b, COUNT(R₂.e) FROM R₁ INNER JOIN R₂ ON

```

R1.c=R2.c GROUP BY R1.b;
SELECT R2.c, R2.d, AVG(R3.e) FROM R2 INNER JOIN R3 ON
R2.d=R3.d GROUP BY R2.c, R2.d;
SELECT R3.d, MIN(R3.e) FROM R3 WHERE d > '10'
GROUP BY R3.d;
COMMIT

```

Побудуємо систему продукцій у вигляді функціональних асоціативних правил:

1. $\{c \mid (a \ \& \ b \mid \text{SUM}(c)),$
2. $d \mid (b \mid \text{COUNT}(e)),$
3. $e \mid (c \ \& \ d \mid \text{AVG}(e)),$
4. $e \mid (\ \& \ d > '10'; d \mid \text{MIN}(e))\}$

Покажемо, що правила 5.2 і 5.3 можуть не активізуватися, тобто існує ситуація, коли не виконуються умови продукції.

Зв'язок типу «один-до-багатьох» визначає однозначну відповідність одного елемента одної множини нулю, одному або більше елементам іншої множини. Таким чином, якщо функціональне асоціативне правило будується між інформаційними одиницями, що належать різним таблицям, таке правило може бути побудоване, але можлива й ситуація, коли умова активізації ядра не виконується.

Твердження. Нехай $\mathcal{R}(R_1(a, \underline{b}), R_2(\underline{b}, \underline{c}))$ – схема реляційної БД. Функціональне асоціативне правило виду $\{c \neq \emptyset; a \Rightarrow \wp(c)\}$ існує, якщо між відношеннями R_1 і R_2 встановлено зв'язок типу 1:М.

Доведення. Грунтуючись на визначенні типу зв'язку «один-до-багатьох» і виключивши ситуацію, коли зв'язаний елемент відсутній (умова $c \neq \emptyset$ у твердженні), покажемо, що завжди можна знайти множину різних елементів однієї множини, які відповідають одному елементу іншої множини, тобто

побудувати функціональне асоціативне правило.

Нехай задані множини $A = \{a_1, \dots, a_n\}$, $B = \{b_1, \dots, b_m\}$ і $C = \{c_1, \dots, c_k\}$ і задані відношення $R_1 \subseteq A \times B$ такі, що b_i не повторюються й $R_2 \subseteq B \times C$, де не повторюються c_i (згідно з ключовими атрибутами, визначеними у схемі БД). Запишемо кортежі добутків у вигляді:

$$\begin{array}{ll}
 R_1 = & \{ \langle (a_1, \dots, a_n), b_1 \rangle \\
 & \{ \langle (a_1, \dots, a_n), b_2 \rangle \\
 & \dots \\
 & \{ \langle (a_1, \dots, a_n), b_m \rangle \\
 R_2 = & \{ \langle (b_1, \dots, b_m), c_1 \rangle \\
 & \{ \langle (b_1, \dots, b_m), c_2 \rangle \\
 & \dots \\
 & \{ \langle (b_1, \dots, b_m), c_k \rangle
 \end{array}$$

Такий запис показує, що в R_1 кожному значенню з $\{b_1, \dots, b_m\}$ може відповідати одне будь-яке значення з $\{a_1, \dots, a_n\}$, і в R_2 кожному значенню з $\{c_1, \dots, c_k\}$ може відповідати одне будь-яке значення з $\{b_1, \dots, b_m\}$.

Розглянемо можливий стан БД (рис. 5.2).

R_1		R_2	
A	<u>B</u>	B	<u>C</u>
a_1	b_1	b_1	c_1
...
a_n	b_m	b_1	c_k

Рисунок 5.2 – Схема реального стану БД

У загальному вигляді відповідність значень атрибутів A і C можна записати у вигляді $a_i \rightarrow (c_1, \dots, c_k)$.

Таким чином, можна застосувати агрегатну функцію для обчислення по атрибуту C , згрупованого по атрибуту A і побудувати відповідне функціональне асоціативне правило $\{c \neq \emptyset; a \Rightarrow \varpi(c)\}$.

Доведення закінчене.

Зауважимо, що на практиці для однозначної ідентифікації значень неключового атрибута в ядрі правила необхідно використовувати значення ключа. Для розглянутого у твердженні прикладу функціональне асоціативне правило $\{c \neq \emptyset; a \Rightarrow c\}$ має вигляд $\{c \neq \emptyset; a, B \Rightarrow c\}$.

5.2 Пошук асоціативних залежностей в реляційних базах даних

На відміну від логічних моделей подання знань, семантичні мережі дозволяють успішно структурувати інформацію. Такі моделі дозволяють розширити коло розв'язуваних проблем порівняно з множиною правил, що належать до них [10].

Використовуючи визначення функціональних асоціативних правил, під семантичною мережею (СМ) будемо розуміти вираз:

$$S = (T, \Omega), \tag{5.2}$$

де T – набір транзакцій, елементи якого виступають у ролі вершин СМ, Ω – множина агрегатних функція, що визначають відношення між вершинами (дуги графа).

Нехай задана система функціональних асоціативних правил:

$$\begin{aligned} &\{Quantity ((; Product (SUM(Quantity) \\ &Price ((; Product (MIN(Price) \\ &Product ((; Name_Customer (COUNT(Product) \\ &Quantity ((; Name_Customer, Product (MAX(Quantity) \\ &Price ((; Material (MIN(Price) \\ &Quantity ((; Price (MIN(Quantity) \} \end{aligned} \tag{5.3}$$

Для ідентифікації складових лівих частин ядра правила в мережі використовується позначення ϖ^i ($i = \overline{1, \infty}$), де i – показує відношення складених

вершин.

Використовуючи структуру СМ, розглянемо зворотну задачу. Сформулюємо запит, ґрунтуючись на аналізі шляхів у СМ. Наприклад, яку інформацію можна одержати, маючи дані про імена постачальників (Name_Customer). Виходячи із системи правил (5.3), запит можна сформулювати наступним чином:

```
SELECT Name_Customer, COUNT(Product) FROM Table  
GROUP BY Name_Customer
```

Аналіз СМ може розширити набір знань. Визначивши необхідний шлях у мережі, запит можна скласти таким чином:

```
SELECT Name_Customer, Product, MIN(Price) FROM Table  
GROUP BY Name_Customer, Product
```

Так, із виразів:

- якщо є постачальник, то можна знайти кількість його поставок;
- і якщо є поставки, то можна знайти мінімальну вартість однієї,

можна одержати вирази:

- якщо є постачальник і його поставки, то можна знайти мінімальну вартість цих поставок.

Останнім десятиріччям спостерігається гібридизація методів інтелектуальної обробки інформації. М'які обчислення поєднують такі області як нечітка логіка, штучні нейронні мережі, видобування знань, бази даних, імовірнісні міркування, еволюційні алгоритми та низку інших. Вони доповнюють одне одного й використовуються у різних комбінаціях для створення гібридних інтелектуальних систем.

Не залишилися осторонь від цікавого й сучасного напрямку дослідники в галузі баз даних. Розробляється нечітка реляційна алгебра й спеціальні розширення структурованої мови (SQL) для нечітких запитів. У цій області інтенсивні дослідження проводять європейські вчені Д. Дюбуа й Г. Праде. [11] Формується перспективний напрямок у сучасних системах обробки інформації

– нечіткі запити до баз даних (fuzzy queries) [12–14].

У цьому контексті можна розглядати два основних питання, найбільш актуальних в цей час: як проектувати, де й у яких структурах зберігати нечіткі дані систем такого класу. Вирішення цих проблем відкриє шляхи інтеграції реляційних баз даних, що накопичили колосальні обсяги інформації, і систем на основі нечіткої логіки.

Метою проведених досліджень є розробка методів зберігання й обробки нечітких даних засобами реляційної моделі, орієнтованої на реалізацію в середовищі сучасних систем управління базами даних. Особлива увага приділена обґрунтуванню вибору схеми реляційної моделі даних для подання функцій належності лінгвістичних змінних.

Розглянемо класичний підхід до побудови реляційного відношення, запропонований Е.Ф. Коддом, і визначимо основні властивості відношень при розширенні множини доменів.

Основним структурним компонентом даних у реляційній моделі даних (РМД) є n -арне відношення, що є підмножиною кортежів декартового добутку доменів, тобто множини значень елементів даних. Для заданих кінцевих множин D_1, \dots, D_n (не обов'язково різних по типу) декартовим добутком $D_1 \times \dots \times D_n$ називається множина добутків виду d_1, \dots, d_n , де $d_1 \in D_1, \dots, d_n \in D_n$. Відношенням R , визначеним на множинах D_1, \dots, D_n , називається підмножина добутку (декартовий добуток) $D_1 \times \dots \times D_n$, тобто $R \subseteq D_1 \times D_2 \times \dots \times D_n$. Множина $D = \{D_1, \dots, D_n\}$ називається доменом.

Домени – це однотипні семантично однозначні (однакові за змістом) значення елементів даних. Елементи декартового добутку d_1, \dots, d_n називаються кортежами, число n визначає ступінь відношення; кількість кортежів визначає потужність відношення.

Нехай $a = \{a_1, \dots, A_n\}$ – множина імен, тоді однозначне відображення виду $\rho : A_i \rightarrow D_i$, де пару $A_i = \langle \rho(A_i), D_i \rangle$ називають атрибутами відношення. Схемою відношення R будемо називати вираз $S(A_1, \dots, A_n)$ в якому всі атрибути

A_i різні. При цьому екземпляр відношення $R(S)$ визначається як підмножина декартового добутку доменів $r_i \subseteq \rho(A_i) \times \dots \times \rho(A_n)$. Екземпляр відношення зі схемою R_i будемо позначати як $R_i(r_i)$. Зазначимо, що перестановка атрибутів у схемі не породжує нового стану БД. Таким чином, множина атрибутів $\{A_1, \dots, A_n\}$ завдає тип відношення й визначає його властивості [15].

Схему БД будемо позначати як множину схем відношень $U = \{R_1, \dots, R_n\}$, де $R_i \in R$ і всі R_i різні. Відповідно, екземпляр БД будемо позначати множиною екземплярів відношень $U(r_1, \dots, r_n)$.

Концептуально реляційна БД є інформаційною моделлю предметної області (ПО), такою, що кожний екземпляр відповідає стану ПО у певний момент часу. Кожний стан моделюється впорядкованою сукупністю значень елементів даних, що відповідають значенням властивостей об'єктів ПО. Об'єкту певного типу відповідає кортеж відношень.

Для задач аналізу даних визначимо додатковий тип відношень, що визначає приналежність існуючих даних до деякого заданого числового відрізка, що характеризує інформаційний об'єкт – відношення фаззифікації.

Будь-яку лінію на координатній площині можна подати у вигляді бінарного відношення, де $Dom R$ визначається значеннями осі абсцис, а $Im R$ визначається значеннями осі ординат. У задачі фаззифікації діаграма містить три показники, які необхідно враховувати при формуванні відношення.

Під нечіткою змінної будемо розуміти набір (N, X, Y) , де N – назва змінної, X – область міркувань, Y – нечітка множина на X . Використовуючи таке визначення, задамо три домена, що відповідають елементам змінної. Нехай $N = \{n_1, \dots, n_m\}$, $Y = \{0, 0.1, \dots, 1\}$, $X = \{x_0, \dots, x_k\}$.

Значення X і Y відповідають обраній шкалі дискретизації координатних осей і описують область належності до параметра N . Для кожного параметра N робимо вибірку даних за значеннями X і будемо діаграму належності, наведену на рис. 5.3.

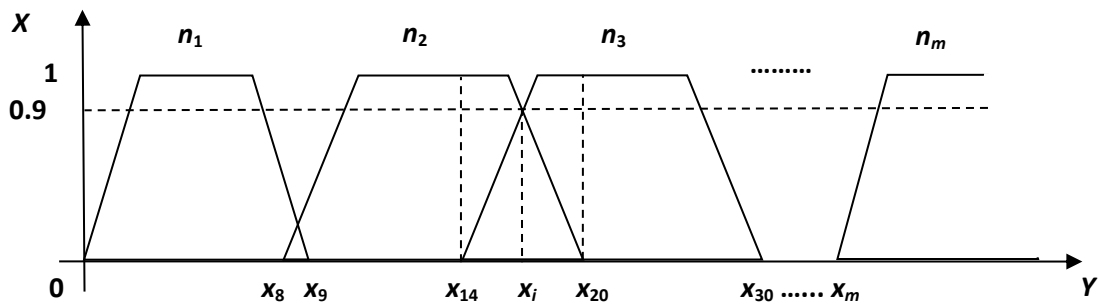


Рисунок 5.3 – Діаграма функції належності

Для розглянутого випадку визначимо відповідні домени для подання нечіткої змінної.

$$D_1 = \{n_1, n_2, n_3, \dots, n_m\};$$

$$D_2 = \{x_0, \dots, x_8, x_9, \dots, x_{14}, \dots, x_i, \dots, x_{20}, \dots, x_{30}, \dots, x_m\};$$

$$D_3 = \{0, 0.1, \dots, 1\}.$$

Задамо множину імен доменів і побудуємо відображення. Для множини імен $A = \{A_1, A_2, A_3\}$ відображення $\rho : (A_1 \rightarrow D_1; A_2 \rightarrow D_2; A_3 \rightarrow D_3)$ визначають множину атрибутів $A = \{A_1, A_2, A_3\}$ і відповідну схему відношення $S(A_1, A_2, A_3)$. Структура відношення R^f у табличному вигляді буде називатися відношенням фаззифікації. Можливі перетини діаграм фаззифікації визначають той факт, що всі значення можуть повторюватися відносно одне одного.

Наприклад, розглянемо фрагмент відношення для показників $\{x_{14}, x_i, x_{20}\}$, наведений на рис. 5.4.

Таким чином, можна говорити про відношення, що включає повний набір кортежів декартового добутку доменів $D_1 \times D_2 \times D_3$ (повний декартовий добуток, у цьому випадку, визначається скінченністю значень системи координат). Виходячи із цього факту, можна визначити, що ключем відношення буде множина всіх атрибутів $K = \{A_1, A_2, A_3\}$.

Очевидно, що інформативність кортежів визначається значеннями на діаграмі фаззифікації. У розглянутої задачі, необхідно враховувати ще один

показник – множину значень вибірки із БД, для якої будується діаграма. Тобто необхідно встановити зв'язок між доменом відношення R^f , що відображує значення осі абсцис і доменом із БД, що відображує значення параметрів фаззифікації [16–18].

A_1	A_2	A_3
n_2	x_{14}	0
n_3	x_{14}	1
n_2	x_i	0.9
n_3	x_i	0.9
n_2	x_{20}	1
n_3	x_{20}	0
...

Рисунок 5.4 – Фрагмент даних для обраних показників

Таким чином, наступна задача, яку необхідно розглянути, пов'язана з інтеграцією відношення R^f із БД, що зберігає основні показники.

Розглянемо задачу в загальному вигляді. Нехай $U(R_1, \dots, R_n)$ – БД, що зберігає основні дані, $R^f(A_1, A_2, A_3)$ – відношення фаззифікації. Задача має сенс, якщо в БД U існує параметр, для якого виконана фаззифікація.

Щоб організувати спільну роботу з базами даних U і R , формалізуємо процедуру інтеграції, спираючись на поетапну нормалізацію. Структура БД U отримана на підставі функціональних залежностей $F = \{M_i \rightarrow N_i\}$, де $M_i, N_i \in U$. Виділимо одну залежність, яка містить атрибут з параметрами фаззифікації, і позначимо її як $W \rightarrow V$, причому W і V можуть бути множинами. Відношення R^f містить одну залежність виду $F' = \{A_1, A_2, A_3 \rightarrow A_1, A_2, A_3\}$. Спираючись на аксіоми виводу, можна одержати еквівалентну множину

$$F' = \{A_1, A_2, A_3 \rightarrow A_1; A_1, A_2, A_3 \rightarrow A_2; A_1, A_2, A_3 \rightarrow A_3\}.$$

Нехай параметр фаззифікації відповідає атрибуту A_2 , тоді для визначення

типу зв'язку необхідно одержати множину $F = F \cup F'$ і розглянути два випадки, що впливають на правила нормалізації.

1. $A_2 \in W$ – пошук неповних залежностей: якщо виконуються функціональні залежності $\xi \rightarrow \zeta$ і $\omega \rightarrow \zeta$, причому $\omega \subseteq \xi$, тоді залежність $\omega \rightarrow \zeta$ є неповною.

2. $A_2 \in V$ – пошук транзитивно залежних елементів: якщо виконуються функціональні залежності $\xi \rightarrow \omega$ і $\omega \rightarrow \zeta$, тоді елемент ζ є транзитивно залежним.

Існування таких залежностей дозволить виконати коректну декомпозицію й встановити зв'язок між базами даних U і R^f .

Якщо $A_2 = W$ або $A_2 = V$, то процес декомпозиції призводить до другої або третьої нормальної форми. Якщо рівняння не виконуються, то неможливо організувати підтримку однозначності зв'язаних даних, тому що асоціація між відношеннями буде відповідати типу «багато-до-багатьох».

Як правило, на практиці умови рівняння не виконуються і для нормалізації необхідно виділити базис F і повторити процедуру декомпозиції. Враховуючи той факт, що структура БД не повинна змінюватися, необхідно зв'язати відношення фаззифікації R^f і БД U без реструктуризації схеми даних. Використовуючи діаграму моделі «сутність-зв'язок», представимо R^f і U у вигляді сутностей (рис. 5.5).

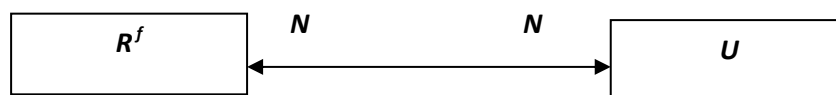


Рисунок 5.5 – Діаграма «сутність-зв'язок» між БД U і R^f

Для усунення зв'язку «N:N» запровадимо додаткову сполучну сутність, яка вирішить проблему цілісності даних за рахунок визначення нових типів зв'язків. «Сутність-зв'язок» буде містити один атрибут – сполучний для R^f і U ,

причому з об'єктивних причин він буде ключовим. На рис. 5.6 показана діаграма інтеграції БД і відношення фаззифікації з підтримкою однозначності зв'язку.

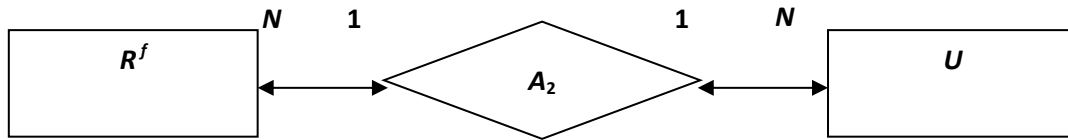


Рисунок 5.6 – Усунення зв'язку «багато-до-багатьох»

Виходячи з опису концептуальної схеми ПО можна бачити, що для коректного з'єднання R^f і U необхідно побудувати проміжну таблицю. Такий підхід гарантує погодженість даних для будь-яких параметрів фаззифікації [19].

Покажемо, що для даної задачі цілком коректні результати при виконанні з'єднання відношень з асоціацією типу «N:N». Можливі значення атрибута $A_1 \in U$ можуть повторюватися стільки разів, скільки це значення перетинає границі діаграми фаззифікації по осі ординат. Тобто кожному значенню атрибута A_1 відповідає рядок унікальних даних.

Якщо A_1 не є ключем, і значення повторюються, то, за визначенням множини, у рядку повинно бути хоча б одне відмінне значення. У термінах розв'язуваної задачі необхідно аналізувати всі такі рядки. В атрибуті A_1 відношення R^f також повторюються значення, які необхідно проаналізувати, причому в різних комбінаціях.

Таким чином, у загальному вигляді для аналізу даних, що накопичуються в реляційних базах даних, досить побудувати відношення фаззифікації й встановити зв'язок з атрибутом (атрибутами), за значеннями якого необхідно провести відповідний аналіз.

5.3 Інтегровані інформаційні системи

Інтеграція розподілених баз даних виявляється ключовою проблемою, розв'язання якої багато в чому залежить від рівня розвитку сучасних методів і засобів комп'ютеризованих інформаційних технологій. Як категорія інтегрована розподілена база даних визначає клас складних і неоднозначних інформаційних об'єктів, механізми побудови і управління якими сьогодні є найактуальнішими.

Багатоаспектність інтегрованих баз даних породжує необхідність компромісного використання цілої низки методів і засобів комп'ютерних технологій для реалізації концепції розподілених баз даних [20].

Існує кілька підходів до визначення поняття «неоднорідні бази даних». Найбільшого інтересу заслуговують два з них, що інтенсивно обговорюються останнім часом. З одного боку, бази даних, які реалізовані засобами різних СУБД є неоднорідними за відповідними їм моделями даних. З іншого боку, бази даних, які підтримуються однією СУБД, але визначаються різними концептуальними схемами, є інформаційно неоднорідними. У подальшому ми будемо використовувати поняття неоднорідності, що містить в собі обидва види неоднорідності.

Головну мету створення систем інтеграції неоднорідних баз даних можна сформулювати таким чином: інтегрована неоднорідна база даних дає змогу прикладній програмі одночасно і спільно використовувати декілька баз даних як єдине ціле.

Інтегрована сукупність різних баз даних з точки зору прикладної програми має логічно виглядати як єдина база даних.

Основна ідея інтеграції даних полягає у запровадженні уявного подання даних (віртуальної бази даних), на яке повинна відображатися кожна із баз даних, що інтегруються. Цьому рівню подання відповідає цілком визначена модель даних, в яку ефективно можуть бути перетворені моделі даних

довільних СУБД. Така модель даних у подальшому називається концептуальною моделлю інтегрованої системи. Концептуальні моделі даних, що підтримуються різними СУБД, виступають внутрішніми моделями стосовно загальної моделі [21].

Суттєвим є те, що для забезпечення умов інтеграції довільних баз даних концептуальна модель даних системи повинна містити в собі засоби, що дозволяють працювати одночасно як із структурованими, так і з неструктурованими даними. Слід зазначити, що над концептуальною моделлю даних інтегрованої інформаційної системи можуть надбудовуватися моделі даних зовнішнього рівня. Тут досить домовитися, що мови програмування і відповідні прикладні програми спираються безпосередньо на концептуальний рівень системи інтеграції.

Разом зі здатністю неоднорідних баз даних до інтеграції іншим, не менш важливим аспектом архітектури є можливість досягнення високого ступеня незалежності й мобільності прикладних програм від типу СУБД.

Відомо, що моделі даних відіграють значну роль у системах управління. Їх основні функції полягають у тому, що вони:

- є ключовими компонентами архітектури СУБД;
- служать основою розробки родин мов високого рівня для взаємодії з базами даних (мов програмування, мов запитів, мов діалогу);
- є основою розробки загальної методології проектування баз даних;
- є засобом забезпечення еволюції баз даних.

Об'єктивні обставини, такі як різні способи формального математичного опису об'єктів; різноманітність структур даних і засобів маніпулювання даними, розвинутих у мовах програмування; різноманітність предметних областей, відображених у базах даних, сприяли необмеженому зростанню кількості моделей даних і СУБД, що їх підтримують.

Однією із причин такого стану слід вважати відсутність методів формального опису й аналізу моделей даних, методів оперування моделями

даних як самостійними об'єктами. З іншого боку, розвиток архітектур СУБД, таких як трирівнева структура або архітектура систем інтеграції неоднорідних баз даних, свідчить про те, що створення таких методів, перш за все методу перетворення однієї моделі даних в іншу, є досить актуальним завданням. Саме перетворювачі моделей даних складають основу архітектур, які розглядаються.

При цьому перспективним вважається такий підхід:

- перетворення вихідної моделі даних у цільову полягає у перетворенні схем і станів конкретних баз даних у схеми і стани баз даних у цільовій моделі даних;
- перетворення операторів мови маніпулювання даними цільової моделі у послідовність операторів вихідної моделі даних.

Існуючі технології інтеграції баз даних здебільшого орієнтовані на використання реляційної моделі даних (ODBC, BDE та ін.), що забезпечує досягнення схемної однорідності за визначенням. Застосування названих технологій дає змогу стверджувати можливість формування професійного рівня схемної інтеграції розподілених баз даних. На схемному інтеграційному рівні розв'язуються завдання табличного подання даних незалежно від специфіки визначення локальних баз даних у середовищах СУБД, таких як Paradox, FoxPro, Oracle, Access та ін.

На цьому рівні інтеграції принципово важливим є існування подання всіх розподілених баз даних у вигляді реляційних схем, що гарантує досягнення схемної модельної однорідності й розв'язання цільової інтеграційної проблеми.

Іншим важливим аспектом побудови інтегрованих розподілених баз даних є та обставина, що у середовищі нових інформаційних технологій бази даних повинні володіти єдиною інфологічною та концептуальною основою [22].

Схемна однорідність реляційних баз даних не стосується проблем забезпечення єдності смислового змісту розподілених даних. Будь-яка сучасна інформаційна система оперує системами локальних баз даних як

технологічними інструментами, при цьому на перший план висувається проблема досягнення семантичної однорідності всіх інформаційних ресурсів.

Традиційно важливим аспектом побудови інтегрованих розподілених баз даних є визначення професійного інтерфейсу на рівні використання мережевих технологій доступу до розподілених баз даних.

Очевидно, що інтеграція баз даних, що перетинаються, в інтересах кількох інформаційних технологій визначає необхідність розгляду методів і засобів управління й доставки однорідних інформаційних ресурсів на основі суперпозиції механізмів маршрутизації й комунікації інформаційних потоків.

Висновки до розділу

У розділі монографії розглянуто результати досліджень у галузі інтелектуального аналізу в системах реляційних баз даних, які у сукупності розв'язують важливе науково-прикладне завдання створення, розробки та використання технології інтелектуального аналізу даних і систем нечіткої логіки з метою підвищення ефективності процесів вилучення знань в інтегрованих реляційних базах даних.

На основі застосування сучасних комп'ютерних технологій, теорії штучного інтелекту, методів нечіткої логіки, математичного апарату реляційних баз даних, методів системного аналізу, дискретної математики, теорії формальних та алгебраїчних систем наведені аналітичні засоби, застосування яких дозволило суттєво підвищити ефективність і поліпшити показники якості функціонування системи підтримки прийняття управлінських рішень за допомогою розподілених інформаційно-аналітичних систем.

Список використаних джерел

1. Filatov V.O., Yerokhin A.L., Zolotukhin O.V., Kudryavtseva M.S. Methods of intellectual analysis of processes in medical information systems. *Information Extraction and Processing*. 2020 (48). P. 92–98.

doi:10.15407/vidbir2020.48.092

2. Аврунін О.Г., Бодяньський Є.В., Семенець В.В., Філатов В.О., Шушляпіна Н. О. Інформаційні технології підтримки прийняття рішень при визначенні порушень носового дихання. Харків : ХНУРЕ, 2018. 132 с. URL: <https://doi.org/10.30837/978-966-659-235-7>

3. Аврунін О.Г., Філатов В.О., Тимкович М.Ю., Кухаренко Д.В., Пятикоп В.О. Комп'ютерне планування малоінвазивних втручань в офтальмології та нейрохірургії. Харків : ХНУРЕ, 2020. 160 с. DOI: 10.30837/978-966-659-283-8

4. Касаткина Н.В. Танянский С.С., Филатов В.А. Методы хранения и обработки нечетких данных в среде реляционных систем. *Автоматика. Автоматизация. Електротехнічні комплекси та системи*. Херсон : ХНТУ. 2009. Випуск 2 (24). С. 84–90.

5. Date K. Introduction to database systems: trans. from English. M. : Publishing House «Williams», 2001. 1072 p.

6. Martin J. Database Organization in Computing Systems : Tr. from English. M. : Mir, 1980. 662 p.

7. Cыcritis D., Lokhovskiy F. Data Models : Trans. from English. M. : Finance and Statistics, 1985. 344 p

8. Filatov V., Semenets V. and Zolotukhin O. «Data mining in relational systems». *Innovative Technologies and Scientific Solutions for Industries*. 2020. №3(13). pp. 65–76. doi: 10.30837/ITSSI.2020.13.065.

9. Maltsev A.I. Algebraic systems. M. : Nauka, 1970. 392 p.

10. Gavrilova T.A., Khoroshevskiy V.F. Knowledge Base of Intellectual Systems. SPb : Peter, 2000. 384 p.

11. Dubois D., Prades A. Theory of opportunities Applications to the representation of knowledge in computer science. M. : Radio and communication, 1990. 288 p.

12. Zade L.A. The concept of a linguistic variable and its application to making

approximate decisions. M. : Mir, 1976. 165 p.

13. Zadeh L.A. Basics of a new approach to the analysis of complex systems and decision-making processes. *Math Today*. M. : Znanie, 1974. P. 5–49.

14. Asai K., Vatada D., Iwai S. et al. Applied fuzzy systems / Ed. T. Terano, C. Asai, M. Sugeno. M. : Mir, 1993. 368 p.

15. Filatov V., Avrunin O., Vlasov O. Model of semantic integration of information systems properties in relay database reengineering problems. *Innovative Technologies and Scientific Solutions for Industries*. 2020. № 4 (14), P. 5–12. doi:10.30837/itssi.2020.14.005

16. Filatov V., Radchenko V. Reengineering relational database on analysis functional dependent attribute. *Proceedings of the X Intern. Scient. and Techn. Conf. «Computer Science & Information Technologies» (CSIT'2015)*, 14-17 sept. 2015. Lviv, Ukraine. P. 85–88.

17. Filatov V., Kovalenko A. Fuzzy Systems in Data Mining Tasks. In: Mashtalir V., Ruban I., Levashenko V. (eds) *Advances in Spatio-Temporal Segmentation of Visual Data. Studies in Computational Intelligence*. Springer, Cham. 2020. Vol 876.. URL: doi.org/10.1007/978-3-030-35480-0_6

18. Filatov V. Fuzzy models presentation and realization by means of relational systems. *Econtechmod : an international quarterly journal on economics in technology, new technologies and modelling processes*. Lublin; Rzeszow, 2014. Vol. 3, № 3. P. 99–102. URL: http://www.pan-ol.lublin.pl/wydawnictwa/Econtechmod2014/Econ_2014_3.pdf

19. Filatov V., Doskalenko S.N. On the Approach to Searching for Functional Dependences of Data in Relational Systems. *Innovative technologies and scientific solutions for industries*. Kharkiv. 2018. No. 3 (1). P. 54–58. doi:10.30837/2522-9818.2018.3.054

20. Filatov, V., Semenets, V. & Zolotukhin, O. Synthesis of Semantic Model of Subject Area at Integration of Relational Databases. *2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL)*. 2019.

doi:10.1109/caol46282.2019.9019532

21. Glava M., Malakhov V. Information Systems Reengineering Approach Based on the Model of Information Systems Domains. *International Journal of Software Engineering and Computer Systems (IJSECS)*, University Malaysia Pahang. 2018. Vol. 4. P. 95–105. DOI: 10.15282/ijsecs.4.1.2018.8.0041

22. Filatov V.O., Yerokhin A.L., Zolotukhin O.V., Kudryavtseva M.S. Information space model in tasks of distributed mobile objects managing. *Information Extraction and Processing*. 2019. № 47(123). P. 80–86. URL: DOI:<https://doi.org/10.15407/vidbir2019.47.080>