УДК 004.652.4+004.827

Н.Б. Шаховська, Д.І. Угрин*
Національний університет "Львівська політехніка",
кафедра інформаційних систем та мереж,
*Буковинський університет,
кафедра інформаційних систем і технологій

АНАЛІЗ ЗАДАЧІ ОПРАЦЮВАННЯ ВІДСУТНОСТІ ТА НЕПОВНОТИ ІНФОРМАЦІЇ У СХОВИЩІ ДАНИХ

Під час реалізації проектів побудови сховищ даних виникає ряд загальних завдань, що залежать від предметної області даних: проектування структури, актуалізація агрегатних значень, розрідження гіперкуба, зниження якості рішень. У статті розглянуто можливі шляхи рішення цих завдань і способи реалізації простих та ієрархічних вимірів.

During realization of projects of construction of depositories given there is a row of general tasks which depend on the subject domain of information: planning of structure, actualization of aggregate values, dilution of hypercube, decline of quality of decisions. The ways of decision of these tasks and methods of realization of the simple and hierarchical measurings are possible in the articles considered.

Вступ

Питання вивчення невизначеності у реляційних базах даних, системах прийняття рішень тощо почало розвиватися ще у 70-х роках минулого століття та поширювалося в різних галузях. Сьогодні до розв'язання задачі опрацювання невизначеностей у сховищах даних (СД) немає єдиного підходу, що зумовлене розрізненістю наукових досліджень; недостатньо розроблені методики проектування схем СД з врахуванням невизначеності; недостатньо вивчені питання ефективного аналізу невизначених даних; комерційні реалізації інформаційних систем коректно опрацьовують лише певні типи невизначеної інформації [5].

Типовими предметними областями, у яких постає задача опрацювання невизначених та нечітко заданих значень, є, наприклад, бронювання туристичних квитків, задачі планування екскурсій, погодні умови в туристичному бізнесі.

Практика розроблення і впровадження реляційних систем збирання даних показала, що через різні причини первинні дані збирають лише частково, а тому їх не завжди можна оптимально використовувати. Це приводить до необхідності застосування багатовимірних баз даних з частковою або слабкою заповненістю. При цьому створювані багатовимірні куби даних (гіперкуби – Data Hypercube) мають низьку щільність заповнення даними, а тому є розрідженими. Тому виникають такі проблеми:

- низька ефективність пошуку і витягання інформації з розрідженого гіперкуба даних;
- некоректність використання набутих значень при агрегації розріджених гіперкубів даних.

Разом з тим, розріджені гіперкуби даних містять потенційно цінну інформацію, ефективне використання якої може зіграти значну роль при ухваленні рішення [2, 8].

Основними проблемами, які виникають в задачах аналізу усунення невизначених та нечітких даних, є розрідження гіперкуба, зниження якості розв'язків та погіршення агрегації розріджених гіперкубів даних.

1. Опис об'єкта дослідження

Сховище даних визначають як предметно-орієнтований, інтегрований, залежний від часу набір даних, призначений для підтримки прийняття рішень різними групами користувачів. Оскільки сховище має предметно-орієнтований характер, його організація націлена на змістовний

аналіз інформації, а не на автоматизацію бізнес-процесів. Ця властивість визначає архітектуру побудови сховища і принципи проектування моделі даних, відмінні від тих, що застосовуються в оперативних системах [3].

Враховуючи специфіку, до проектування сховищ даних зазвичай висуваються такі вимоги:

– повинні бути виділені статичні дані, що регулярно модифікуються;

– повинні бути спрощені вимоги до запитів з метою вилучення запитів, що могли б вимагати множинних запитів SQL у традиційних реляційних СУБД;

– повинна бути забезпечена підтримка складних запитів SQL, що вимагають послідовної обробки великої кількості записів.

Уведемо формальну модель сховища даних.

Реляційною базою даних називають трійку

$$DB = \langle r, R, Z \rangle,$$

де r – множина відношень бази даних, R – множина їх схем, Z – множина обмежень цілісності.

Тоді сховищем даних, побудованим на основі реляційної моделі, назвемо трійку

$$DW = \langle DB, rf, Rf, func \rangle,$$

де DB – множина баз даних (або множина відношень, їх схем та обмежень цілісності, які можна вважати окремою базою даних та які містять інформацію про певну частину предметної області – наприклад, дані складського обліку),

rf – відношення, у якому зберігається агрегована інформація і за даними якого здійснюється прийняття рішень (відношення фактів).

Rf – схема відношення *rf*.

$func$ – множина процедур прийняття рішень.

Тоді нові дані (або рішення) – це результат застосування функцій сховища даних над відношенням фактів:

$$Design = func(rf, user\_param).$$

де *user_param* – параметри користувача (або вимоги), які ставляться до рішення.

Оскільки відношення rf містить агреговану інформацію з відношень баз даних, то зв'язок між ним і відношеннями баз даних DB приводить до утворення так званого гіперкуба даних (моделі багатовимірного подання даних) [3].

Виміром назвемо універсум відношень бази даних DB_i – $V_i : Universum(DB_i)$. Кожен вимір містить напрямки консолідації даних, що складаються із серії послідовних рівнів узагальнення (рівнів ієрархії).

Відношення між вимірами – деяке відношення, яке є зв'язком між вимірами.

$$V_1, V_2, ..., V_n \to \mathrm{Re}l.$$

Своєю чергою, *Rel* можуть бути параметрами для інших відношень між вимірами, і тим самим створювати ієрархію вимірів.

Осями багатовимірної системи координат є основні атрибути аналізованого бізнес-процесу. На перетинах вимірів (dimensions) знаходяться дані, що кількісно характеризують процес – значення (measures).

Формування відношення rf здійснюється на основі функції агрегування Agg [1]: $rf : Agg(\mathrm{Re}l_1, ..., \mathrm{Re}l_n)$

Внаслідок встановлення відношень між вимірами та операцій агрегування, гіперкуб у переважній більшості випадків є сильно розрідженим, тому проблема опрацювання невизначеності тут постає набагато сильніше ніж у реляційних базах даних.

Розглянемо складові елементи гіперкуба.

Гіперкуб даних містить одне або більше вимірів і є впорядкованим набором комірок (рис. 1). Кожна комірка визначається одним і лише одним набором значень вимірів – атрибутів. Комірка може містити дані – виміру або бути порожньою.

Під виміром розумітимемо множину атрибутів, що утворюють одну із граней гіперкуба. Прикладом часового виміру є список днів, місяців, кварталів. Прикладом туристичного виміру може бути перелік оздоровчо-туристичних об'єктів: пунктів відпочинку та оздоровлення, районів конкретного виду відпочинку тощо. Для одержання доступу до даних користувачеві необхідно вказати одну або декілька комірок шляхом вибору значень вимірів, яким відповідають необхідні комірки. Процес вибору значень вимірів називатимемо фіксацією атрибутів, а множини вибраних значень вимірів – множиною фіксованих атрибутів.

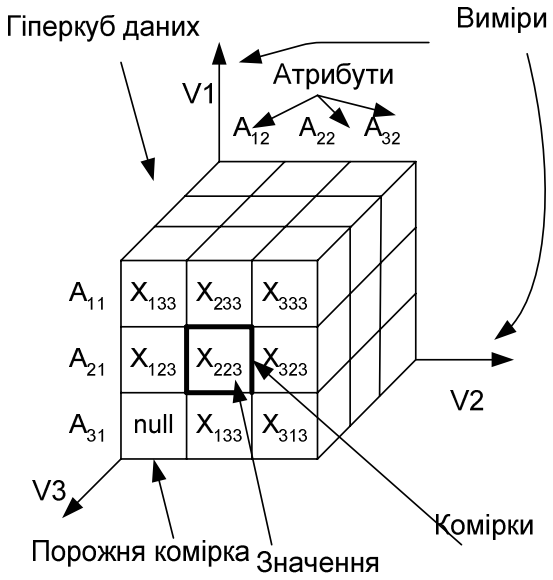

Рис. 1. Гіперкуб даних

Отже, V – множина вимірів гіперкуба, $\boldsymbol{A}_{V_i} = \{A_{1_i}, A_{2_i}, ..., A_{k_i}\}, i = 1,...,n$ – множина атрибутів виміру V_i, $\boldsymbol{A} = \boldsymbol{A}_{V_1} \cup \boldsymbol{A}_{V_2} \cup ... \cup \boldsymbol{A}_{V_n}$ – множина атрибутів гіперкуба, $\boldsymbol{V}' \subseteq \boldsymbol{V}$ – множина фіксованих вимірів, $\boldsymbol{A}' \subseteq \boldsymbol{A}$ – множина фіксованих атрибутів.

Гіперкуб даних позначимо як множину комірок, що відповідає множинам ***V***, ***A***:

rel(V,A).

Підмножина гіперкуба даних, що відповідає множині фіксованих значень, позначатимемо як $rel'(V', A')$.

Кожній комірці гіперкуба даних $rel \in \boldsymbol{rel}$ відповідає єдино можлива множина атрибутів вимірів $A_{rel} \subset \boldsymbol{A}$. Комірка може бути порожня (не містити даних) або містити значення показника.

Для отримання доступу до даних користувачу необхідно вказати множину необхідних вимірів $\boldsymbol{V}' \subseteq \boldsymbol{V}$ і значень атрибутів $\boldsymbol{A}' \subseteq \boldsymbol{A}$ (фіксувати атрибути). Множина комірок, що відповідають відповідним атрибутам та вимірам, позначимо як $rel'(V', A') \mid rel' \subseteq \boldsymbol{rel}$.

Ключем виміру назвемо атрибут, який однозначно визначає кортеж (рядок) виміру гіперкуба.

Куби підтримують ієрархію вимірів і формул без дублювання їх визначень. Набір відповідних кубів складає сховище даних.

Розглянемо приклад з поданням куба із трьома вимірами:

Рис. 2. Гіперкуб з трьома вимірами

Наявність добре розвиненої ієрархії агрегованих даних за рівнями агрегації є відмінною рисою сховища даних.

2. Постановка задачі

Проведені дослідження [2, 7, 9] показали, що більшість кінцевих користувачів не працюють з детальними даними, а в основному з агрегованими показниками. Структура сховища даних відображає цю ситуацію і дає змогу кінцевому користувачу швидко і зручно одержувати необхідну для його агреговану інформацію з подальшою навігацією за всіма рівнями агрегації.

У процесі експлуатації необхідність у деяких детальних даних може значно зменшитися, що є причиною поділу детальних даних на поточні і застарілі. Тоді як поточні дані регулярно використовуються і тому зберігаються на накопичувачах з швидким доступом, застарілі детальні дані можуть зберігається на місткіших накопичувачах з повільнішим доступом.

Зв’язок іж агрегованими та деталізованими даними подано на рис. 3.

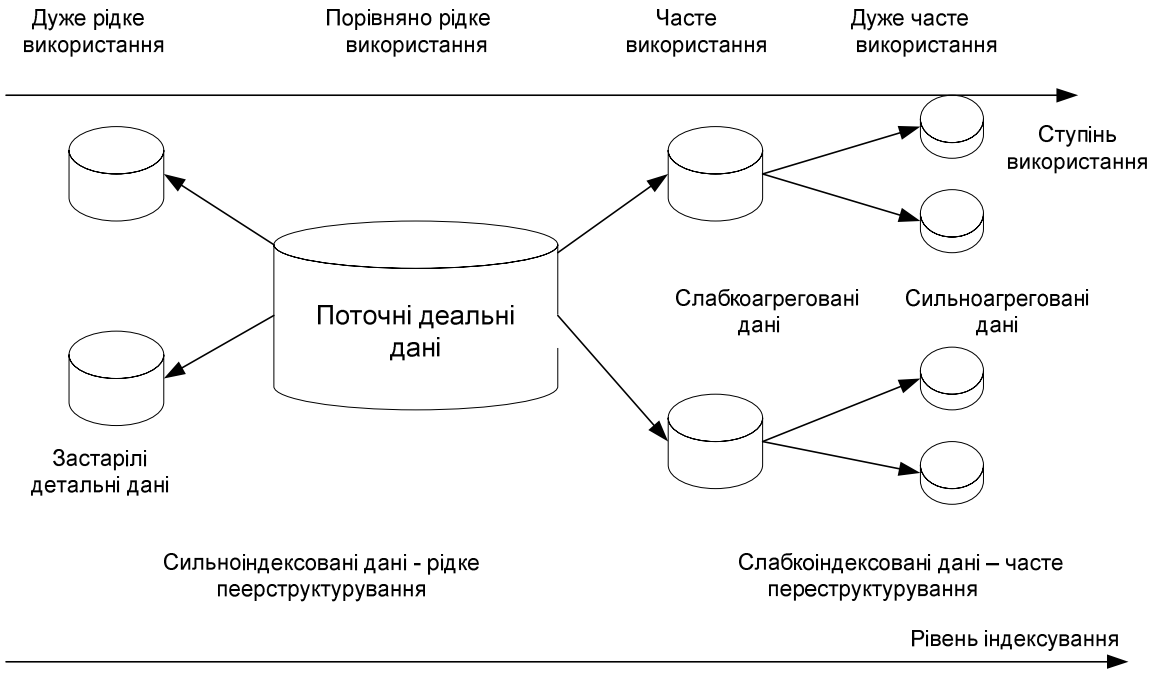

Рис. 3. Зв’язок між детальними та агрегованими даними у СД

У більшості випадків при створенні інформаційних систем, орієнтованих на аналіз даних, питання представлення інформації у розріджених гіперкубах даних обходяться стороною. Та методи роботи з щільними і розрідженими гіперкубами даних повинні істотно розрізнятися. Тому розроблення альтернативних методів пошуку і агрегації даних, що дозволяють вирішити вищезгадані проблеми, є актуальним завданням [9].

Створення оптимальних методів пошуку і агрегації інформації в розріджених гіперкубах даних та підвищення якості рішень передбачає проведення робіт у таких напрямах:

- дослідження моделі даних і формалізація методів оцінки щільності гіперкуба даних;
- дослідження і розроблення ефективних методів доступу до інформації в розрідженому гіперкубі даних;
- розроблення альтернативного методу агрегації розрідженого гіперкуба даних;
- дослідження можливостей застосування різних методів візуалізації розріджених гіперкубів даних, зокрема з використанням ГІС-ТЕХНОЛОГІЇ, двовимірної і тривимірної машинної графіки [1,2].

Причини невизначеностей у сховищах даних та проблеми, які породжуються у зв’язку з цим, наведені на рис. 4.

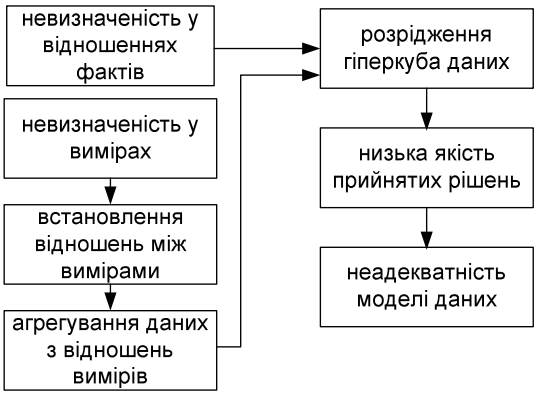

Рис. 4. Причини появи невизначеності у сховищі даних

Наповнення гіперкуба даними за недостатньої кількості початкових даних приводить до утворення порожніх комірок. Гіперкуби даних з великою кількістю порожніх комірок називають розрідженими [1,4].

Поняття агрегації в гіперкубі даних нерозривно пов'язане з поняттям ієрархічного виміру. Агрегація даних Ag – отримання значень, відповідних атрибутам деякого рівня ієрархічного виміру V на основі значень рівня l–1. Отримується у результаті виконання операції згортки. Саме агрегація призводить до виникнення зв’язків між даними.

Розглянемо ієрархічні виміри *V* з *L* рівнями (рис. 5). Первинні дані (факти) відповідають нижньому рівню ієрархії (l=0).

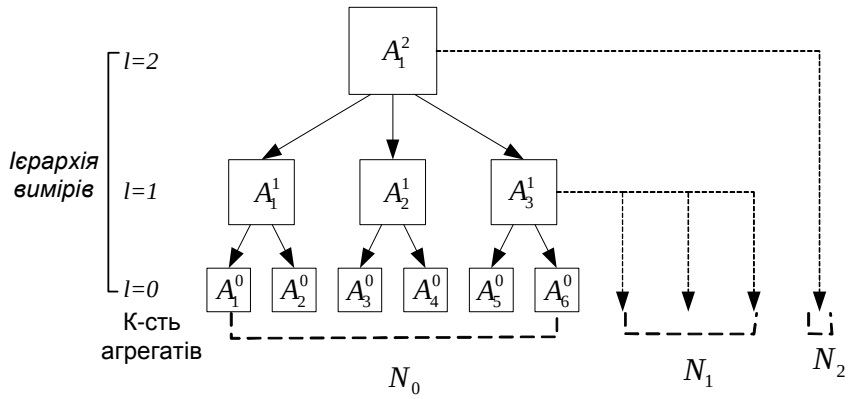

Рис. 5. Агрегація гіперкуба даних. Одномірне представлення

Обчислення агрегатів здійснюється відповідно до методу агрегації, що використовується. Наприклад, у разі підсумовування значення агрегату на рівні ієрархії l=1 може бути обчислене за формулою: $Ag_j^{\ 1} = \sum_{i=1}^{A_j} Ag_i^{\ 0}$ де – кількість фактів, що відповідають атрибутам, які є дочірніми відносно атрибута j.

Узагальнюючи, одержимо формули обчислення агрегатів за методом підсумовування на решті рівнів ієрархії: $Ag^l{}_j = \sum_{i=1}^{A_j} Ag_i^{\ l-1}, l = 1,...,L; j = 1,...,N_l$

Вісь виміру V, що спочатку містить атрибути, відповідні нижньому рівню ієрархії (l=0), може бути доповнена атрибутами, відповідними рівням ієрархії, починаючи з l=1. Отже, відмінність між атрибутами, відповідними первинним даним і атрибутами, відповідними агрегатам, є умовною.

Операція згортки даних [9] у цьому випадку являє собою побудову зрізу гіперкуба даних, що відповідає зміні мітки рівня агрегації $l_1 | 0 \le l_1 < L$ на рівень $l_2 | l_1 < l_2 \le L$. Операція деталізації відповідає зміні мітки рівня $l_1 | 1 \le l_1 \le L$ на рівень $l_2 | 0 \le l_2 < l_1$.

Кількість агрегатів для одного виміру $N_V = \sum_{i=1}^{L} N_i$. Розглянемо випадок двох вимірів (рис. 6).

Рис. 6. Агрегація гіперкуба даних. Двовимірне представлення

Кількість агрегатів, збережена в гіперкубі даних поряд з первинними даними, залежить від кількості атрибутів, що відповідають рівням ієрархії вимірів гіперкуба, починаючи з $l = 1$, і може істотно перевищувати кількість первинних даних.

У випадку двох вимірів кількість агрегатів становитиме суму значень областей: $Ag_{01}, Ag_{02}, Ag_{10}, Ag_{11}, Ag_{12}, Ag_{20}, Ag_{21}, Ag_{22}$. З іншого боку, кількість агрегатів можна обчислити як різницю кількості всіх значень гіперкуба і кількості значень, відповідних області первинних даних Ag_{00}. Кількість значень агрегата є добутком $N_0^1 \times N_0^2$. Отже, кількість агрегатів гіперкуба даних у двовимірному випадку становить:

$$N_A = \left(N_0^1 + N_1^1 + ... + N_{L_1}^1\right) \times \left(N_0^2 + N_1^2 + ... + N_{L_2}^2\right) - N_0^1 \times N_0^2 =$$

$$= \sum_{i=0}^{L_1} N_i^1 \times \sum_{i=0}^{L_2} N_i^2 - N_0^1 \times N_0^2$$

На випадок довільної кількості вимірів V одержимо: $N_A = \prod_{j=1}^{V} \sum_{i=0}^{L_j} N_i^j - \prod_{j=1}^{V} N_0^j$, де V – к-ть атрибутів і-го рівня ієрархії виміру, а L_i – к-ть рівнів ієрархії виміру j [1–4, 6].

3. Основний матеріал

Для того, щоб мати можливість класифікувати інформацію у сховищі даних, необхідно передбачити, як простіше її реалізувати та яку класифікацію агрегації використати альтернативніше: часткову чи повну.

Ступінь агрегації куба обчислюється як:

$$\alpha = \frac{a}{a^*},$$

де a – реальна кількість агрегованих значень показників, а* – максимально можлива кількість агрегатних значень вихідних даних куба [1].

На практиці, визначаючи, яку обрати з формул для а і а* спочатку розбирають прості випадки із двома–трьома вимірами, а потім, у разі великих неточностей, переходять до узагальненого варіанта. Те саме стосується й рівнів ієрархії у вимірах: спочатку розглядаються випадки простих вимірів (з одним рівнем), а потім на прикладі вимірів з декількома рівнями виводиться узагальнена формула. Такий підхід дає змогу легше зрозуміти процес одержання агрегованих значень показників. На рис. 7 представлений спрощений приклад виміру територіальних об'єктів, що має ієрархічну структуру. Спочатку база даних містить факти, що відповідають атрибутам нижнього рівня ієрархії (первинні дані). Суть процесу агрегації полягає в обчисленні значень, що відповідають атрибутам інших рівнів ієрархії на основі фактів нижнього рівня. Отримані в ході агрегації значення називаються агрегативами. Агрегативи використовуються при аналізі даних на різних рівнях деталізації й, як правило, обчислюються на етапі формування гіперкуба даних з метою скорочення часу відгуку на запит користувача [1,7].

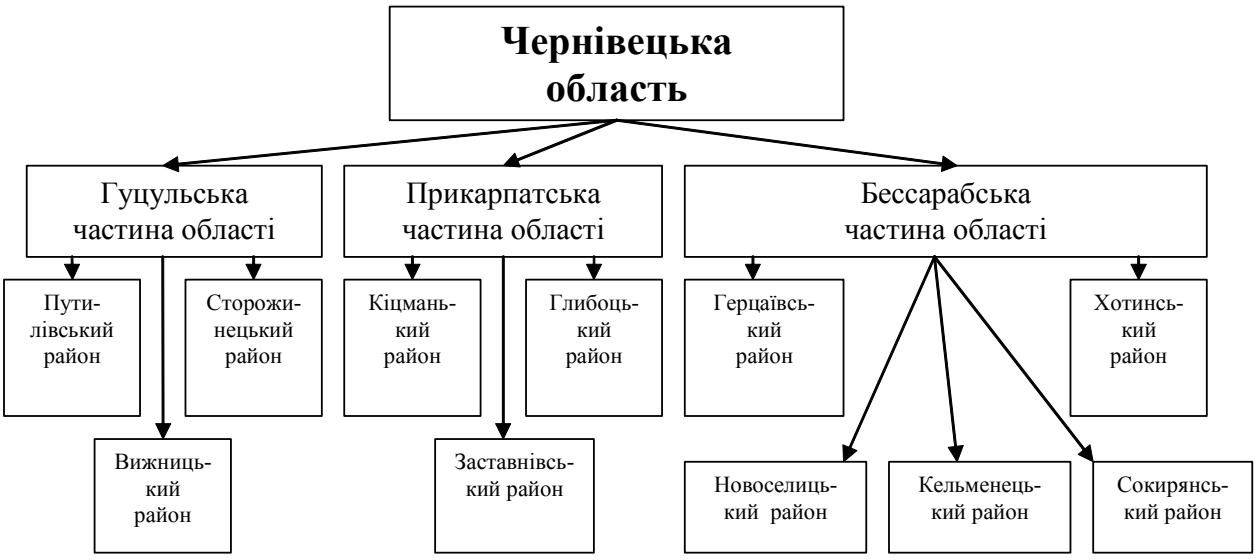

Рис. 7. Ієрархічне вимірювання Чернівецької області

Агрегація даних є одним із ключових понять технології OLAP. Саме формування агрегативів (попереднє або динамічне) уможливлює проведення операцій поглиблення (Drill-down) і згортки (Roll-up) при проведенні аналізу даних. Існує кілька традиційних методів агрегації (табл. 1).

Вибір того або іншого методу агрегації даних залежить від конкретної розв'язуваної задачі. Технологічно процедура підрахунку агрегативів виконується з використанням мап агрегації, що містять стандартні методи агрегації, зазначені в таблиці. У багатьох популярних OLAP-системах як

метод агрегації "за замовчуванням" використовують метод підсумовування, що припускає наявність первинних даних на нижньому рівні ієрархії. Однак виникає питання про застосовність даних методів при агрегації цих у розріджених гіперкубах [2].

Методи агрегації

SUM	Підсумовування деталізованих даних	$P = \sum_{i=1}^{N} x_i$
WSUM	Зважена сума	$P = \sum_{i=1}^{N} p_i x_i$
MIN (MAX)	Мінімальне (максимальне) значення	$P = \min_{l \in N}(x_i)$
AVERAGE	Середнє значення	$P = \frac{\sum_{i=1}^{N} X_i}{N}$
WAVERAGE	Зважене середнє	$P = \frac{\sum_{i=1}^{N} p_i x_i}{\sum_{i=1}^{N} p_i}$

Очевидно, що в стандартних методах агрегації не враховується ситуація відсутності первинних даних, що відповідають деяким міткам нижнього рівня ієрархічного виміру. Але ж саме таку ситуацію являє собою агрегація даних у розрідженому гіперкубі. Розглянемо цю проблему на простому прикладі.

На рис. 8 представлена картограма деякого показника, що характеризує рівень розвитку туризму в Чернівецькій області. Кольорове підфарбування відповідає величині показника, білим кольором позначено відсутність даних за цим показником ("білі плями"). Значення показника є первинними даними на рівні суб'єктів області в ієрархічному вимірі туристичних об'єктів.

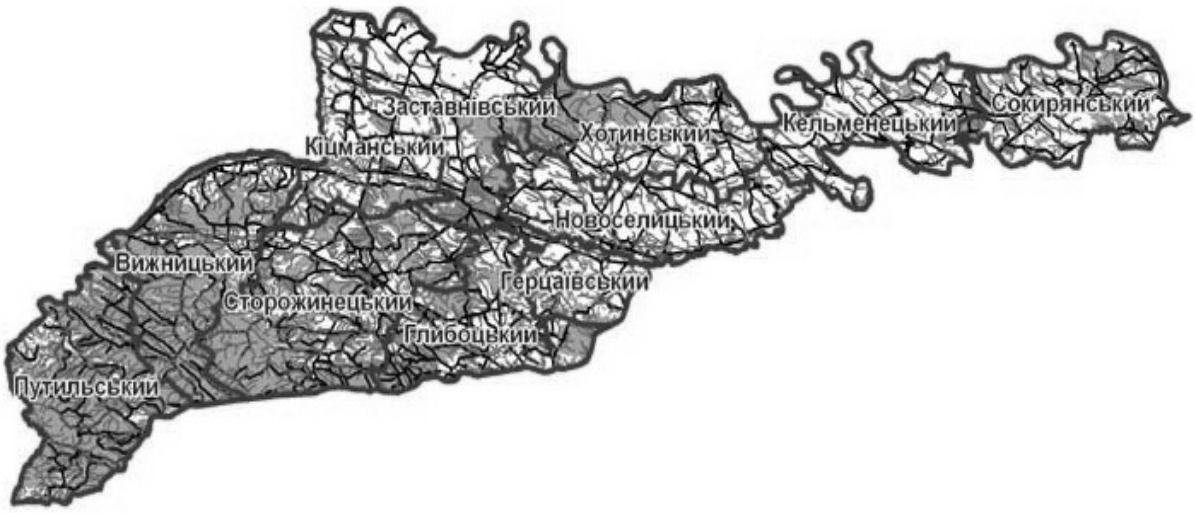

Рис. 8. Розвиток туризму в Чернівецькій області

Використання стандартних методів агрегації в цьому випадку приводить до досить сумнівних результатів. Так, наприклад, застосування методу підсумовування є неможливим через відсутність даних щодо об'єктів, неточної одночасної належності певного району до різних частин області тощо. Тому і отриманий результат буде досить далекий від дійсності. Можливе застосування

методу обчислення середнього значення, однак використання отриманого результату в процесі аналізу буде не цілком коректним.

Розв'язуючи важливі аналітичні задачі, аналітик має знати не тільки значення показника, але й те, наскільки отримане значення є достовірним. Обчислення агрегатива за методом середнього значення за наявності первинних даних за всіма значеннями нижнього рівня в ієрархії дає абсолютну вірогідність, тому що немає причин вважати, що це середнє значення могло бути спотворено. У нашому випадку вірогідність отриманого результату можна оцінити лише як наближену, а отже, далеко не точну.

Отже, при проведенні агрегації в розрідженому гіперкубі за методом обчислення середнього необхідне введення додаткового параметра, що характеризує рівень вірогідності отриманого результату. Технологічно цю операцію можна здійснити створенням додаткової карти агрегації, що містить розрахунок рівня вірогідності для кожного отриманого в ході агрегації значення.

Обчислення агрегату на наступному рівні ієрархії (l=1) здійснюється за формулою: $Ag_j^1 = \frac{\sum_{i=1}^{V_{ij}} ag_i^0}{V_j}$, де V_j – кількість фактів, які відповідають атрибутам, що є дочірніми відносно атрибута j.

Узагальнюючи, одержимо формули обчислення агрегатів на решті рівнів ієрархії: $Ag_j^l = \frac{\sum_{i=1}^{V_{ij}} ag_i^{l-1}}{V_j}, l = 1,..., N$.

Розглянутий метод може бути застосований при побудові карт агрегації в розріджених гіперкубах даних і дає можливість оцінити рівень достовірності одержаних результатів на етапі аналізу.

Висновки

Під час реалізації проектів побудови сховищ даних виникають задачі, що залежать від самої предметної області: проектування структури ієрархічних вимірів, опрацювання відсутності, нечіткості та неповноти даних.

Наукова новизна.

Досліджено ефективні методи доступу до інформації в розрідженому гіперкубі даних.

Практична цінність. Наукові результати, отримані у цій статті, дають змогу проводити подальші практичні дослідження за методами розрідження гіперкуба, підвищення якості рішень та покращання агрегації з метою усунення невизначеності.

1. Хрусталёв Е.М. Агрегация данных в OLAP-кубах // Алекс Консалтинг & Софт. – 2006. 2. Заботнев М.С. Методы представления информации в разреженных гиперкубах данных ФГНУ // "Госинформобр". 3. Шаховська Н.Б., Кісь Я.П. Використання класифікаційних правил для зменшення невизначеності у сховищах даних, побудованих на основі реляційної моделі // Вісник Нац. ун-ту "Львівська політехніка". – 2005. – № 546. – С. 155–162. 4. Шаховська Н.Б. Методи усунення невизначеності у сховищах даних // Тези доповідей міжнародної конференції «Комп'ютерні науки та інженерія CSE-2006». – Львів, 2006. 5. Мейер Д. Теория реляционных баз данных. – М.: Мир. – 1987. 6. Raden N. Данные, Данные и только данные // ComputerWeek-Москва. – 1996. – №8. 7. Тарасов Д.О., Шаховська Н.Б. Опрацювання нечіткостей на різних етапах формування замовлення // Вісник Нац. ун-ту "Львівська політехніка". – 2001. – № 438. 8. Стулов А.П. Особенности построения информационных хранилищ // Открытые системы. – 2003. – № 4. 9. Вон Ким Три основных недостатка современных хранилищ данных // "Открытые Системы". – 2003. – №2.