УДК 681.3

Н. Б. Шаховська, А.Я Вовчина
НУ "Львівська політехніка",
кафедра "Інформаційні системи та мережі"

ОБМЕЖЕННЯ НА ЗАСТОСУВАННЯ ТРИЗНАЧНОЇ ЛОГІКИ У БАЗАХ ДАНИХ

The paper described the problems, which appeared in decision of 3-valued logic in relational databases.

У статті розглянуто проблеми, які виникають у базах даних при застосуванні тризначної логіки, та способи їх розв'язання.

Сучасні бази даних та знань можуть містити усе більшу кількість невизначеної інформації. Такий процес є закономірним, враховуючи динаміку світу та швидкість його розвитку. Тим не менше, нечітка чи неповна інформація теж повинні бути належно подані та опрацьовані, особливо це стосується предметних областей з явно вираженою динамікою розвитку (сфери, які обліковують фактор часу – музеї, картотеки; сфери, які працюють із неструктурованими даними – біржі праці, соціальні фонди тощо; сфери, діяльність яких пов'язана з ризиком у разі неточної інформації – планові відділи, ринки; виробництво та сфера послуг).

З уведенням стандарту SQL89 [6] з'явилась можливість подання невизначеної інформації у базах даних, тобто відбувся перехід від двозначної до тризначної логіки, а саме логіки Лукасевича [2], [1]. Таким чином було досягнуто більшої виразності у поданні знань про світ у реляційній моделі.

Проте перехід до тризначної логіки не вирішив усіх проблем, пов'язаних із невизначеністю інформації. Так, залишається незрозуміло, як відрізняти фактори

- відсутності значення певного атрибута для об'єкта;
- відсутності знань про значення цього атрибута.

Крім того, невідомо, як подавати інформацію про атрибути деякого об'єкта, у правильності значень яких ми не впевнені на сто відсотків (тобто, чи можна відобразити у базі даних здогадки та чутки) [3], [6].

Частково вище перераховані проблеми можна вирішити існуючими засобами тризначної логіки, здійснивши деякі зміни у структурах даних та способі трактування невизначеностей.

Розглянемо кілька випадків.

1) Невизначеність вводиться на рівні кортежу

Такий випадок подає рівень непевності знань про об'єкт, відображених кортежем. Для коректного подання та опрацювання невизначеної інформації додамо до схеми таблиці ще один атрибут (позначимо його *UNK*), який і буде позначати ступінь невизначеності. Як його домен залежно від предметної області й методу отримання інформації (автоматичний розрахунок на основі алгоритмів або ручне занесення даних) можна використовувати процентні відношення, коефіцієнти, різноманітні шкали або лінгвістичні оцінки [4]. За замовчуванням цьому атрибутові присвоюємо значення, яке означає найвищий ступінь істинності.

При такому опрацюванні даних можна працювати з k-значною логікою, у якій k визначається згідно з вимогами предметної області (визначення оптимального розміру шкали подано у [7]).

Для опрацювання та аналізу невизначеностей за допомогою запиту в реляційних операторах слід здійснювати селекцію кортежів за значеннями атрибута *UNK*. Додавши таблицю, у якій би містилися "розшифровки" значень атрибута *UNK*, або, використовуючи

лінгвістичні оцінки, ми отримуємо можливість побудови нечітких запитів (наприклад, "Вибрати усіх працівників, рівень інформованості про яких є високий"). Варто зазначити, що у разі стовідсоткової довіри до кожного кортежу ми отримуємо традиційне реляційне відношення та застосовуємо традиційні операції над ним.

Опишемо, як впливатиме введення атрибута ступеня невизначеності для кортежу на операції над відношеннями.

Нехай r та s – відношення зі схемою R, r' та s' – відношення зі схемою $R \cup UNK$. Тоді $r \cap s$, $r \cup s$ і $r - s$ є відношеннями зі схемою R, а $r' \cap s'$, $r' \cup s'$ і $r' - s'$ – відношеннями зі схемою $R \cup UNK$.

Доповнення до відношення r' працюватиме коректно у разі присвоєння усім значенням атрибута *UNK* найвищого ступеня довіри.

Спеціальні оператори над відношеннями – вибірка, проекція та з'єднання – повинні ще включати умови для відбору даних за ступенями довіри.

Оператор вибірки передбачає аналіз нечіткого значення по атрибуту *UNK*.

$$\sigma_{(UNK\ \Theta\ unk)\cup(A\ \Theta\ a)}(r') = \left\{ t \in r' \,\middle|\, t(UNK)\ \Theta\ unk,\ t(A)\ \Theta\ a \right\}, \quad (1)$$

де Θ - множина символів (знаків) бінарних відношень над парами доменів. Вважається, що кожний атрибут A порівняний по рівності і по нерівності. Як правило, будуть вживатися тільки такі знаки порівняння над одним доменом: $=, \neq, <, \leq, \geq, >$ [1,8].

На оператор проекції введення атрибута *UNK* не здійснює жодного впливу.

Для операції з'єднання слід розглянути випадки, коли відношення є повністю з'єднувальними або не повністю з'єднувальними [8].

Означення. Відношення $s_1, s_2, \ldots, s_m$ *повністю з'єднувальні*, якщо кожний кортеж в кожному відношенні є членом деякого списку з'єднувальних на S кортежах [8].

Для повністю з'єднувальних відношень введення атрибута *UNK* не впливає на операцію з'єднання (природне з'єднання або еквіз'єднання).

У випадку неповної з'єднувальності (у стандарті SQL92 [8] – це оператори LEFT JOIN або RIGHT JOIN) значення атрибута *UNK* для кортежів підлеглої таблиці, які не потрапляють у відношення, будуть вважатися рівними найвищому ступеню довіри.

$$r \underset{unk}{\triangleright\!\!\triangleleft} s' = \pi_{(R,B,NVL(UNK,\max)UNK)}\left(r \underset{\bullet}{\triangleright\!\!\triangleleft} s'\right), \quad (2)$$

де r – традиційне відношення, s' – відношення з атрибутом *UNK*, R – множина атрибутів відношення r, S – множина атрибутів відношення s', не включаючи атрибута *UNK* ($S' = S \bigcup UNK$), *B* – множина тих атрибутів з S, яких нема у відношенні r ($B \subset S$, $B \not\subset S \bigcap R$), max – значення, яке означає найвищий ступінь довіри, $NVL(UNK, \max)UNK$ – операція, яка присвоює max усім значенням атрибута UNK для нез'єднувальних кортежів відношення s', $\underset{\bullet}{\triangleright\!\!\triangleleft}$ – ліве з'єднання (включаються усі кортежі відношення r і лише ті кортежі відношення s', у яких значення по з'єднувальних атрибутах збігаються). Спочатку виконується операція лівого з'єднання для відношень зі схемами S' і R. Потім над отриманим з попередньою операції відношенням здійснюється операція проекції, за якою утвореним у результаті з'єднання порожнім значенням атрибута *UNK* присвоюється значення max.

Очевидно, що такий спосіб опрацювання невизначеностей є найпростішим для реалізації, дає змогу працювати з логікою $k \geq 3$. Водночас саме через свою простоту він має

недостатню виразність, оцінюється достовірність значень для цілого кортежу, що не відповідає реальним потребам (адже для кожної задачі існують атрибути, достовірність значень яких безпосередньо впливає на результати аналізу для одної предметної області і не настільки критична для іншої).

2) Невизначеність вводиться на рівні значень атрибутів

Засобами стандарту SQL92 можна підтримувати 4-значну логіку, використовуючи визначення ступенів істинності системи Дунна-Белнапа [2]:

1. Присутня чи відсутня достовірна інформація,
2. Присутня чи відсутня недостовірна інформація.

Присутні у таблиці значення атрибутів вважаються достовірною інформацією, відсутні значення означають фізичну відсутність даних. Для подання існуючої, але не відомої у даний час інформації використовуються значення, встановлені за замовчуванням. Для позначення відсутньої невідомої інформації використовуються маркери NULL.

Для значень за замовчуваннями використовуються значення, імовірність яких появи у вибірці є найбільшою.

При обробці таблиць, атрибути яких можуть мати 4 ступені істинності, потрібно враховувати появу NULL у вибірці і відповідним чином її опрацьовувати.

Наведемо приклад запиту SQL, який опрацьовує наявність у вибірці NULL-значень та ілюструє застосування чотиризначної логіки (запит узято з бази даних клієнта “Фенікс”). Даний запит призначений для одержання зведеної інформації про контракти, заключені з певним підрозділом у певний період та за визначених підстав. Запит $p_dca_contract містить параметри відбору, причому не усі з них можуть бути задані (тобто можуть бути NULL), а dcr_contract – необхідні дані. Для виконання запиту, який описується, проводиться порівняння значень відповідних атрибутів підлеглих запитів. Якщо значення однойменних атрибутів не рівні між собою, то перевіряється умова “OR *x* IS NULL”, де *x* – ім’я відповідного атрибута.

```
SELECT [$p_dca_contract].contract_id AS p_contract_id, [$p_dca_contract].dept_id AS p_dept_id, [$p_dca_contract].begdate AS p_begdate, [$p_dca_contract].enddate AS p_enddate, [$p_dca_contract].optype AS p_optype, [$p_dca_contract].inreason_id AS p_inreason_id, [$p_dca_contract].outreason_id AS p_outreason_id, "Обороти" AS periodtype, dcr_contract.contract_id, dcr_contract.dept_id, (dcr_contract.optype) AS stroptype, SUM(dcr_contract.suma) AS suma, suma AS abssuma
FROM [$p_dca_contract], dcr_contract
WHERE
( dcr_contract.contract_id  LIKE [$p_dca_contract].contract_id OR [$p_dca_contract].contract_id IS NULL )
AND  ( dcr_contract.dept_id LIKE  [$p_dca_contract].dept_id OR [$p_dca_contract].dept_id IS NULL )
AND  ( dcr_contract.optype LIKE  [$p_dca_contract].optype OR [$p_dca_contract].optype IS NULL )
AND (dcr_contract.evdate>=[$p_dca_contract].begdate OR [$p_dca_contract].begdate IS NULL)
AND (dcr_contract.evdate<=[$p_dca_contract].enddate OR [$p_dca_contract].enddate IS NULL)
AND (  ( dcr_contract.optype = 1 AND  ( reason_id LIKE [$p_dca_contract].inreason_id OR [$p_dca_contract].inreason_id IS NULL ) )
 OR ( dcr_contract.optype = -1 AND  ( reason_id LIKE [$p_dca_contract].outreason_id OR [$p_dca_contract].outreason_id IS NULL ) ) )
GROUP BY [$p_dca_contract].contract_id, [$p_dca_contract].dept_id, [$p_dca_contract].begdate, [$p_dca_contract].enddate, [$p_dca_contract].optype, [$p_dca_contract].inreason_id, [$p_dca_contract].outreason_id, dcr_contract.contract_id, dcr_contract.dept_id, dcr_contract.optype;
```

Рис. 1. *SQL-запит, який опрацьовує NULL значення*

До переваг такого способу опрацювання невизначеностей належать:

- чітке розмежування понять відсутніх та невідомих даних;
- простий та читабельний спосіб подання умов відбору;

- зручно використовувати для виконання звітів та аналітичних вибірок.

Одним із недоліків є важкодокументованість як наслідок використання спеціальних значень за замовчуванням.

3) Невизначеності вводяться для групи атрибутів

Атрибути у відношенні групуються за функціональним значенням (наприклад, ім'я, прізвище, по батькові становлять групу з назвою *ПІБ*; місто, вулиця, будинок складають групу *АДРЕСА*). Для позначення ступеня істинності групи використовують додатковий атрибут UNK, описаний у випадку 1. Опрацювання значень таких атрибутів здійснюється аналогічно.

До переваг такого методу належить зручність опрацювання та аналізу інформації (особливо це стосується випадків, коли в результаті аналізу потрібно визначити якусь комплексну оцінку довіри до інформації тощо). Проте явним недоліком є значне зростання структур даних та залежність від предметної області при створенні груп.

Основні доводи на користь використання невизначених значень і тризначної логіки:

По-перше, наявність виділених значень NULL деякою мірою вирішує проблему представлення неповної інформації.

По-друге, тризначна логіка дає змогу у деяких випадках не піклуватись про наявність невизначених значень при формуванні запитів. Тобто невизначені значення і тризначна логіка дозволяють підвищити виразність мови запитів.

З іншого боку, тризначної логіки у тому вигляді, у якому вона введена в SQL, є недостатньо у тих випадках, коли за змістом значення unknown умови вибірки є недозволеними. Іншими словами, якщо користувача не цікавить неповна інформація, то автоматичне трактування NULL при розрахунку арифметичних виразів і вироблення логічного значення unknown при розрахунку логічних виразів дійсно дозволяє коротко й виразно формулювати запити. Але для вибірки неповної інформації потрібно явно використовувати NULL як спеціальне значення.

1. Codd, E.F. The Relational Model for Database Management Version 2. Reading, Mass.: Addison-Wesley, 1990 2. Hähnle, R. (1999) Deduction in Many-Valued Logics: Surway, http://i12www.ira.uk.de 3. К. Дж. Дейт. Введение в системы баз данных. – 1999. – С. 581 – 599 4. Дюбуа А., Прад А. Теория вероятностей. Приложение к представлению знаний в информатике. – М.: Радио и связь, 1990. – С. 42 – 70 5. Кузнецов С. Д. Неопределенная информация и трехзначная логика // СУБД #05-06/97, http://www.osp.ru./dbms/1997/05_06/65.html 6. Кузнецов С. Д. Стандарты языка реляционных баз данных SQL: краткий обзор // СУБД #2/96, http://www.citforum.ru/database/articles/art_2.shtml#part_17 7. Лилвак Б.Г. Экспертная информация: Методы получения и анализа. – М.: Радио и связь, 1982. – С. 184. 8. Мейер Д. Теория реляционных баз данных: Пер. с англ. – М.: Мир, 1987. – 608 с., ил*