

типовою методикою. Розбивши проміжок інтегрування в цих інтегралах вибірковими значеннями з частотної таблиці отримуємо інтервальні оцінки теоретичних моментів через статистичні. В результаті в системі (4) і величини $m_i (i = 0, 1, 2, 3, 4)$ також є відомими інтервалами.

Частотна таблиця *групує* експериментальні дані з неперервної генеральної сукупності. Моменти цього "групового" розподілу відрізняються від моментів неперервного розподілу [2]. Інтервальний варіант поправок Шеппарда отримуємо реалізуючи інтервальну оцінку правої частини рівності

$$\int_{y_i - \frac{h}{2}}^{y_i + \frac{h}{2}} x^k p(x) dx = h(y_i^k p(y_i)) + h^3 \frac{1}{2 \cdot 2!} (y_i^k p(y_i))' + \dots + h^{2j-1} \frac{1}{2^{j-1} (2j-2)!} (y_i^k p(y_i))^{(2j-2)},$$

де y_i – середини інтервалів таблиці; h – крок таблиці.

1. Больше в Л. Н., Асимптотически пирсоновские преобразования, Теория вкроятностей и ее применения, том УІІІ, вып. 2, 1963, с.129-155. 2. Крамер Г., Математические методы статистики., Мир., М., 1975. 3. Романовский В., Математическая статистика., ГОНТИ НКТП СССР, 1938.

УДК 519.689

Д.О. Тарасов

ЗАСОБИ ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ ДАНИХ НА ОСНОВІ ПРОМИСЛОВИХ СТАНДАРТІВ

Різнопланові інформаційні системи на базі реляційних СУБД широко використовуються для автоматизації обліку, опрацювання та аналізу інформації. Метою роботи таких систем є спрощення процесів прийняття рішень за рахунок використання інформації з БД. Однак рішення має ґрунтуватися на несуперечливій інформації. Таким чином, для забезпечення результативності дії інформаційної системи СУБД повинна забезпечувати цілісність даних у БД. Різні СУБД використовували власні засоби для забезпечення цілісності даних. Їх засоби розвивались відповідно до специфіки застосування, забезпечення сумісності версій та політики фірми виробника.

Упродовж останніх років гостро стало питання інтеграції та обміну даними між інформаційними системами. Ця обставина та швидкий розвиток СУБД і прикладних програм зумовлює необхідність використання узагальнених стандартів. Їх складовою частиною є засоби опису реляційної моделі, правил функціонування інформаційної системи та правил цілісності даних.

Переважає більшість сучасних реляційних СУБД та інструментальні засоби розробки наділені засобами використання мови SQL для роботи з традиційними реляційними БД.

Сучасний стандарт мови SQL - SQL 92, прийнятий у 1992 році Міжнародною Асоціацією по Стандартах (ISO) та Американським Інститутом по Національних Стандартах (ANSI), розроблено на базі SQL 89, SQL 86 та розробленої у 1970 році співробітниками компанії IBM мови SQL.

Наявні на ринку СУБД мають різний рівень сумісності зі стандартом, тому для перевірки відповідності конкретної реалізації мови SQL стандарту використовують рівні відповідності (Full 92, Intermed 92, Entry 92, SQL 89, SQL 86 - перелічені по спаданню відповідності до SQL 92). Існують СУБД, які, при належній якості проектування БД та процедур опрацювання даних, дозволяють не застосовуючи деякі елементи SQL 92 (наприклад тимчасові таблиці) забезпечувати належний рівень цілісності даних. Це свідчить про певну надлишковість стандарту, орієнтовану на старі версії. Як недолік SQL можна навести недостатнє забезпечення цілісності переходу між станами та операцій над доменами.

Розглянемо декларативні засоби збереження цілісності стану БД у стандарті SQL 92. Для цього наведемо основні види порушень цілісності даних у реляційних структурах. До основних можна віднести порушення:

1. функціональних залежностей;
2. обмежень на значення атрибутів та домени;
3. обмежень на значення в межах відношення;
4. обмежень на інформацію у БД.

Обмеження, перелічені у пунктах 3)-4), переважно зумовлені правилами функціонування конкретної системи та не пов'язані безпосередньо з моделлю БД.

Для огляду можливостей SQL 92 наведемо приклад реалізації найбільш вживаних обмежень – на значення атрибутів та домени і на значення в межах відношення.

Приклад 1. Для визначення атрибута *customer_num* первинним ключем відношення *customer* (у складі команди визначення

відношення) мовою SQL (у стандарті SQL92) використовуються наступні команди:

```
CREATE DOMAIN dm_name AS CHAR(30);  
CREATE TABLE customer (customer_num NUMBER(10)  
    CONSTRAINT Customer_pkey_constraint PRIMARY KEY,  
    fname dm_name, lname dm_name CONSTRAINT  
    LName_notnull_constraint NOT NULL);
```

Перша команда описує домен *dm_id* та *dm_name*. Друга створює відношення *customer(customer_num, fname, lname)* з обмеженням - значення атрибута *lname* повинно бути визначено. Назви обмежень записуються після параметра CONSTRAINT.

Приклад 2. Для створення іменованих користувачем обмежень цілісності (точніше цілісності стану БД) використовується команда:

```
CREATE ASSERTION ім'я_обмеж CHECK (предикат)  
[умови_перевірки];
```

Параметр *ім'я_обмеж* задає назву, за якою можна визначити, що відбулася спроба порушення правила цілісності стану БД заданого параметром *предикат*. Необов'язковий параметр *умови_перевірки* визначає вид перевірки обмеження (відкладена, безпосередня).

Незважаючи на чітку розробку декларативної цілісності SQL 92 допускає потенційні ситуації порушення цілісності. Це стосується роботи з курсорами, ізоляції транзакцій.

Поширеність та значні можливості SQL дозволили використати його як прототип стандартів мов у таких проблемно-орієнтованих областях, як часові та об'єктно-орієнтовані БД.

1. Мартин Грабер. Справочное руководство по SQL. "ЛОРИ", 1997. 2. Стивен Бобровски. Oracle7 и вычисления клиент/сервер. "ЛОРИ", 1995. 3. С. Уинкуп. Microsoft SQL Server 6.5 в подлиннике: пер. с англ. - СПб.: BHV - Санкт-Петербург, 1998. - 896 с. 4. Date C.J. An Introduction to Database Systems. - Addison-Wesley Longman, Inc., 1995. 5. The Informix Guide to SQL. Informix Software, Inc. 1991.

УДК 519.8

І.П. Твердохліб, Г.Г. Цегелик

МЕТОД ВИЗНАЧЕННЯ СТАБІЛІЗОВАНОЇ ФУНКЦІЇ СПОЖИВАННЯ

В [1] розглядалась оптимізаційна задача апроксимації функції споживання довільної ринкової економічної системи з