

ДЕЯКІ АСПЕКТИ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ЗНАНЬ МОВИ SQL

Черепанова Ю.Ю.

старший викладач кафедри програмної інженерії

Широкопетлева М. С.

старший викладач кафедри програмної інженерії Харківський національний університет радіоелектроніки

Ключові слова: автоматизована перевірка, система тестування знань, критерії тестових завдань, мова структурованих запитів.

Keywords: automated testing, system testing, test task criteria, Structured Query Language

На сьогоднішній день підготовка фахівців в напрямку комп'ютерних наук практично неможлива без вивчення різних СУБД і написання запитів до бази даних. У більшості технічних навчальних закладів сфери ІТ вивчають ці питання, без цього у наш час неможливо розробляти більшість програмних продуктів. Тому важливими є закріплення та перевірка знань студентів у цій сфері. Система тестування знань мови запитів може використовуватись для спрощення завдань перевірки і навчання студентів. Автоматизація цього процесу дає багато переваг, до того ж у час технологій, в першу чергу Internet, така система буде більш відповідати вимогам сучасності. Метою роботи є розробка методики формування тестових завдань та оцінки результатів для програмної системи тестування знань мови SQL. Тестування повинне проводитися за допомогою тестів у відкритій формі, а саме студенту пропонується вирішити задачу –

написати текст запиту до бази даних. Система повинна оцінити правильність запиту.

В існуючих дослідженнях розглянуті питання формування тестового набору, але не враховані особливості перевірки знань з мов програмування, зокрема, мови запитів.

Для наведення методики формування тестових завдань окрім стандартних критеріїв [1] (критерій дискримінативності, надійності, валідності, складності тесту) запропонуємо критерії, яким повинні задовольняти тестові набори:

1. Критерій повноти – наведені у тестовому наборі питання повинні покривати усі елементи теми, за якою проводиться тестування;

2. Критерій відсутності перетину тестових питань: перевірки підлягають лише фрагменти завдання, за яким сформульоване питання.

Докладніше опишемо критерій повноти. Для цього наведемо терміни «зміст поняття» та «об'єм поняття»:

Таблиця 1 –
Тематичні одиниці підрозділу DDL

Підрозділ	Тема	Тематична одиниця	Рівень
CREATE	Table	Структура таблиці	1
		Декларативні обмеження на дані	1
		Декларативні обмеження цілісності	2
		Завдання каскадного видалення	3
	Index	Первинні індекси	1
		Некластеризовані індекси	2
		Складові індекси	3
		Покривні індекси	3
	View	Завдання представлення	1
		Іменування стовпців	2
ALTER TABLE	Зміна типу даних		1
	Додавання / вилучення обмежень полів		2
	Додавання / вилучення / модифікація стовпців		3
DROP TABLE	Просте вилучення об'єктів		1
	Каскадне вилучення об'єктів		2

змістом поняття є сукупність суттєвих ознак, які характеризують даний предмет, а об'єм поняття – сукупність або множина предметів, які входять до складу поняття. Провівши аналогію між поняттям (основним об'єктом предметної галузі) та дисципліною, за якою проводиться тестування, можна зауважити, що до змісту дисципліни входить увесь перелік тематичних одиниць (тем, розділів, підрозділів, тощо), а об'єм складає множина тестових питань, які покривають увесь зміст кожної теми. Формалізуємо наведене:

$$S = \cup R^i,$$

де S – об'єм всієї дисципліни; R^i – i -а тематична одиниця, $i \in [1; n]$, n – кількість тематичних одиниць в дисципліні.

Для запобігання «обміну думками» під час проведення тестування тест по-

винен містити не менш ніж 10 питань за кожною темою.

Наприклад, для перевірки знань з мови SQL, виділимо розділи: DML, DDL, DCL. В кожному з них визначимо тематичні одиниці, взаємозв'язок яких для розділу DDL наведено в табл.1. Наведений рівень складності може застосовуватися для автоматизації формування тестових наборів для стійкої групи користувачів або дискримінаційного тестування (для отримання максимальної оцінки «задовільно» за успішне проходження тесту першого рівня складності).

Тобто для формування тесту з цього підрозділу необхідно скласти не менш ніж 150 питань відповідно до наведеної тематики.

На підставі критерію відсутності перетину тестових питань, пропону-

ється обмежити перевірку відповідей лише тематикою питання, наприклад, при перевірці знань за тематичним розділом «Декларативні обмеження на дані» необхідно перевірити створений об'єкт на відповідність структури та обмежень на запропоновані стовпці без врахування обмежень цілісності даних та типів інших стовпців. Формалізуємо:

$$\forall i, j: R^i \cap R^j = \emptyset,$$

де $R^i, R^j - i(j)$ - тематична одиниця, $i, j \in [1; n]$, n - кількість тематичних одиниць в дисципліні.

Таким чином, уся тестова множина може бути представлена:

$$S = \cup Q_i^l,$$

де Q_i^l - l -те питання тестового набору, яке входить до складу $R^i, l \geq 10$.

Так, для формування тестового набору з одного тематичного розділу, слід навести однотипні питання, наприклад, для тематичного розділу «Декларативні обмеження на дані» наведемо таблицю Students (id integer, Name varchar (30), DOB date, Group varchar (15), Address varchar (50), Rating double) та запити:

створіть таблицю Students з таким обмеженням даних: не можна зараховувати студентів, молодших за 10 та старших за 120 років та обов'язковим заповненням поля Address;

створіть таблицю Students з таким обмеженням даних: рейтинг студентів не нижчий за 150 та назва групи повинна містити «-17-», тощо.

Для визначення коректної відповіді слід для кожного запиту сформувані еталонні запити, за допомогою яких

можна отримати результуючу вибірку з використанням еталонного запиту на створення об'єкта та службової інформації (словника даних). Також необхідно розробити показники оцінки результуючих запитів для кожної з тематичних одиниць і ввести їх до бази тестових питань з вказівкою тематичної одиниці, а всі завдання повинні відноситися до певної одиниці для здійснення перевірки.

Тестовий набір для підсумкового контролю знань повинен містити питання з кожної тематичної одиниці, але при досить великій кількості тем та обмеженні часу пропонується проводити тестування, призначене для окремих груп користувачів, а в тестові набори включати питання одного рівня складності. Наприклад, в першу чергу можна сформувати тест з питань рівнем складності 2 (оцінка «добре»), після складання якого можна сформувати 2 набору з рівнем складності 1 (для перевірки володіння базовими знаннями для тих, хто не склав першу частину) та рівнем 3 - для користувачів, які можуть претендувати на відмінну оцінку. Також можна передбачити ведення поточного контролю шляхом автоматичного формування тестових наборів на підставі питань з обраного розділу / підрозділу без урахування складності завдань. Кількість питань в завданні викладач може самостійно визначити, але ця кількість не повинна бути меншою, ніж кількість тематичних одиниць. Тобто при ієрархічній структурі організації завдань полегшується процес формуван-

ня тестових наборів з урахуванням критерію валідності за змістом.

Для оцінки результатів тестування при автоматизації перевірки знань з мови SQL необхідно створити набір вимог за кожною тематичною одиницею та алгоритм порівняння еталонного та запропонованого користувачем запитів.

Наприклад, для теми «Декларативні обмеження даних» результати отриманих вибірок пропонується порівнювати за такими критеріями: з використанням службової інформації перевіряється співпадіння створеного обмеження (constraint) з еталонними без урахування наявності інших обмежень та необов'язковим збіганням назв інших стовпців. Але обмеження можуть не повністю співпадати, тоді рекомендується перевірити наявність обмеження та виконати перевірку роботи обмеження з використанням запитів на додавання інформації та сценаріїв, що дозволяють змінювати поточне оточення (за необхідності). Аналогічно перевірка коректності

створення запитів на вилучення та модифікацію об'єктів проводиться з використанням службової інформації та еталонних запитів.

При проведенні поточного контролю і формуванні завдань з рівним рівнем складності для оцінювання можна застосовувати коефіцієнти, які відповідають рівням складності. Пропонується загальну оцінку розраховувати за формулою: $M = k * A / P$, де k – коефіцієнт складності (для середнього рівня складності – $k = 1$, для низького – $k = 0.7$, для підвищеного – $k = 1.3$). За наявності більшої кількості рівнів можна коефіцієнт складності задавати самостійно.

Література:

1. Карпенко, А. П. Тестовый контроль качества обучения и критерии качества образовательных тестов. Обзор [Электронный ресурс] / А. П. Карпенко, А. С. Домников, В. В. Белоус // Наука и образование: электронное научно-техническое издание. – 2011. – № 04. – Режим доступа: <http://technomag.edu.ru/doc/184741.html>. – 15.06.2016г.