

УДК 004.65

**ПОРІВНЯННЯ ЗАСОБІВ ПЕРЕВІРКИ ОБМЕЖЕНЬ У РЕЛЯЦІЙНИХ СУБД**

Помернюк М. С., Безносик О. Ю.

**СРАВНЕНИЕ СРЕДСТВ ПРОВЕРКИ ОГРАНИЧЕНИЙ В РЕЛЯЦИОННЫХ СУБД**

Помернюк М. С., Безносик А.Ю.

**COMPARISON OF THE TOOLS FOR CONSTRAINTS CHECKING IN THE RELATIONAL DATABASES**

Pomernyuk M., Beznosyk O.

Національний технічний університет України «КПІ»

[o.beznosyk@gmail.com](mailto:o.beznosyk@gmail.com)

*В даній статті розглядається ефективність застосування різних механізмів перевірки обмежень в реляційних базах даних на прикладі СУБД SQLServer, SQLite та PostgreSQL. Для перевірки обмежень використовуються стандартні засоби СУБД (процедура CHECK) та тригери. Порівнюються результати, отримані в різних операційних системах. Наводяться висновки щодо доцільності використання тих чи інших засобів в залежності від СУБД, що використовується, типів даних, що перевіряються, та кількості одночасних перевірок.*

*Ключові слова: СУБД, тестування, тригери, обмеження-перевірка CHECK*

*В данной статье рассматривается эффективность применения различных механизмов проверки ограничений в реляционных базах данных на примере СУБД SQLServer, SQLite и PostgreSQL. Для проверки ограничений используются стандартные средства СУБД (процедура CHECK) и триггеры. Сравниваются результаты, полученные в разных операционных системах. Приводятся выводы о целесообразности использования тех или других средств в зависимости от используемой СУБД, типов проверяемых данных и количества одновременных проверок.*

*Ключевые слова: СУБД, тестирование, триггеры, ограничение-проверка CHECK*

*This article deals with the issue of effectiveness of using different tools for constraints checking in the relational databases such as SQLServer, SQLite and PostgreSQL. The regular database tools (CHECK procedure) and triggers are being used to check the constraints. The results got in the different operating systems are being compared. The conclusions on appropriateness of using different tools depending on databases being used, data types being checked and a number of simultaneous tests are given.*

*Keywords: database, testing, triggers, constraint-checking CHECK*

**Вступ**

Продуктивність є одним з найважливіших показників будь-якого програмного продукту. Саме вона визначає вартість та швидкодію програми. Метою будь-якого програміста є збільшення продуктивності створюваного додатка, використовуючи

засоби-аналоги з меншим часом виконання. Зокрема, якщо говорити про бази даних, однією з найважливіших їх властивостей є цілісність даних, що забезпечує коректність та валідність збережених даних в будь-який момент часу [1]. Цілісність даних підтримується за допомогою обмежень, які можуть бути реалізовані як штатними засобами СУБД, так і за допомогою тригерів [2,3].

### **Опис поставленого завдання**

У літературі стверджується, що обмеження-перевірка CHECK, реалізоване штатними засобами СУБД, працює швидше ніж тригер, що перевіряє те ж саме [4-6].

Тригер – це спеціальний вид збережених процедур, які спрацьовують на певні події в таблиці. Тригер пов'язується з певною таблицею і найчастіше виконує захисну роль для даних.

Існують три події, на які можуть реагувати тригери – додавання, зміна та вставка даних, тобто тригер реагує на будь-які спроби вплинути на дані. Коли відбувається спроба вставки, оновлення або видалення даних у таблиці, і для цієї дії оголошений тригер, він викликається автоматично. Його не можна обійти. На відміну від вбудованих процедур, тригери не можуть викликатися безпосередньо і не отримують та не приймають параметри.

Основна перевага тригерів полягає в тому, що вони можуть містити логіку, що виконується комплексно. Вони дозволяють:

- робити каскадні зміни залежних таблиць у базі даних, забезпечуючи більш складну цілісність даних, ніж стандартна перевірка обмежень CHECK;
- оголошувати індивідуальні повідомлення про помилки;
- містити не нормалізовані дані;
- порівнювати стан даних до і після зміни.

На відміну від обмеження CHECK, тригери можуть посилатися на поля в іншій таблиці [2].

Враховуючи певні переваги використання тригерів та зважаючи на велику кількість різних СУБД, що мають суттєві відмінності як з точки зору архітектури, так і призначення, а також беручи до уваги, що за допомогою обмежень можуть перевірятися дані різних типів (чисельні, строкові значення, дати тощо), доцільним є проведення досліджень, пов'язаних з порівнянням швидкодії обмежень-перевірок, що реалізовані за допомогою штатних засобів різних СУБД та за допомогою тригерів.

### **Проведення експерименту**

Порівняння продуктивності перевірки обмежень, що реалізована за допомогою штатних засобів СУБД (процедура CHECK) і за допомогою тригера, здійснювалося для СУБД SQLServer та PostgreSQL на комп'ютері Dell Inspiron 1545 (процесор: Intel Pentium Dual Core T4200 2100 МГц; частота шини: 800 МГц; кеш 2-го рівня: 1024 Кб; ОЗУ: DDR2 2048 Мб) з операційною системою Windows XP і для СУБД SQLite в операційній системі Android.

Для проведення експериментів використовувалася одна таблиця з бази даних, для якої було реалізовано 5 видів обмежень:

- перевірка поля на позитивне значення;
- перевірка поля на відповідність регулярному виразу;
- перевірка обмежень для двох полів одночасно;
- перевірка трьох строкових полів на відповідність довжині в 20 символів;

– перевірка обмежень для всіх полів.

Для заповнення таблиці тестовими даними було виконано 10000 запитів INSERT. Кожне з реалізованих за допомогою CHECK і тригера обмежень тестувалося по п'ять разів за допомогою базового набору тестових даних. Таким чином, час, витрачений на перевірку обмеження, обчислювався як усереднене значення п'яти вимірів.

### **Практичні результати**

Досліджуючи продуктивність обмеження CHECK, реалізованого штатними засобами СУБД SQLServer і за допомогою тригера, можна дійсно переконатися, що обмеження CHECK працює швидше ніж тригер, який перевіряє те саме. Результати експериментів наведено в табл. 1 і на рис. 1.

Подібні результати було отримано і для бази даних SQLite (табл. 1, рис. 2).

Але порівняння продуктивності обмеження CHECK, реалізованого штатними засобами СУБД і за допомогою тригера для тієї ж таблиці і аналогічного набору тестових даних для бази даних PostgreSQL, показали істотно інші результати. Як і в попередніх випадках, результати експериментів наведено в табл. 1 і на рис. 3.

Як видно з отриманих результатів, продуктивність тригерів значно зростає, якщо тригер задіяний для перевірки більш ніж трьох обмежень. Такий результат експериментів суперечить твердженням, що недоліком тригера є те, що він завжди працює повільніше обмежень CHECK.

Подібна відмінність у результатах експерименту для розглянутих СУБД, можливо, може бути пояснена тим, що PostgreSQL з самого початку розроблялася як база з великим числом налаштувань і більш точною відповідністю стандарту. PostgreSQL вважається найдосконалішою СУБД, що поширюється на умовах відкритих вихідних текстів. У PostgreSQL реалізовано багато можливостей, зазвичай присутніх тільки в комерційних СУБД, таких як DB2 і Oracle. PostgreSQL має ряд налаштувань, які підвищують швидкість доступу: парціальні індекси; компресія даних; виділення пам'яті; покращений кеш тощо [7].

Для того, щоб переконатися, що продуктивність тригерів дійсно зростає з кількістю обмежень, що перевіряються, практичні експерименти з базою даних PostgreSQL проводилися також на більш потужному комп'ютері Dell Vostro 3460 (процесор: Intel Core i3 2370M 2400 МГц; частота шини: 1333 МГц; кеш 3-го рівня: 3072 Кб; ОЗУ: DDR3 4096 Мб) з операційною системою Ubuntu 12.04. Як видно з результатів, наведених в табл. 2 і на рис. 4, тенденція ефективності використання тригерів в залежності від кількості обмежень, що перевіряються, зберігається і в цьому випадку.

### **Висновки**

Таким чином, на підставі отриманих результатів практичних досліджень, можна зробити висновок, зокрема для бази даних PostgreSQL, що не завжди реалізація обмежень за допомогою тригерів працює повільніше, ніж за допомогою стандартної процедури CHECK. Отримані практичні результати показують, що продуктивність перевірки обмежень, реалізованої штатними засобами СУБД і за допомогою тригерів, залежить від бази даних, що використовується. Зокрема, для СУБД PostgreSQL при кількості задіяних обмежень більше трьох для їх перевірки рекомендується використовувати механізм тригерів.

Таблиця 1

Час перевірки обмежень

№	Приклад обмеження	Реалізація обмеження		Час, витрачений на перевірку обмеження в SQLServer, мс	Час, витрачений на перевірку обмеження в SQLite, мс	Час, витрачений на перевірку обмеження в PostgreSQL (Windows XP), мс	Час, витрачений на перевірку обмеження в PostgreSQL (Ubuntu 12.04), мс
		а	б				
1	Перевірка поля на позитивне значення	а	CHECK	65,8	112,6	265,8	86,6
		б	TRIGGER	545,2	1332,8	290,6	104,0
2	Перевірка поля на відповідність регулярному виразу	а	CHECK	92,4	258,2	196,6	99,8
		б	TRIGGER	662,8	1467,6	427,8	102,3
3	Перевірка двох попередніх обмежень одночасно	а	CHECK	92,2	343,2	507,5	142,4
		б	TRIGGER	704,2	1657,4	181,0	88,0
4	Перевірка трьох строкових полів на відповідність довжині в 20 символів	а	CHECK	99,2	407,0	714,4	237,2
		б	TRIGGER	659,6	1762,8	387,0	132,4
5	Перевірка обмежень для всіх полів	а	CHECK	485,0	723,2	1090,8	418,8
		б	TRIGGER	1258,2	2398,8	476,5	203,6

Час, витрачений на перевірку

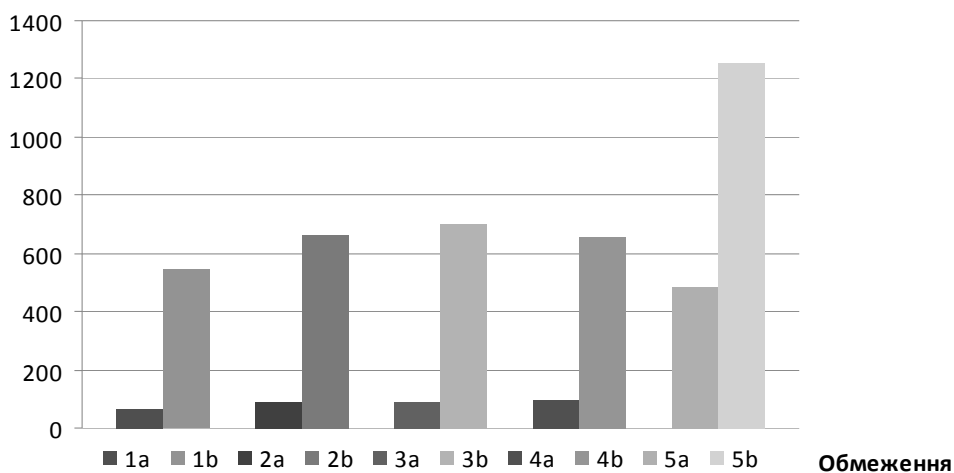


Рис. 1. Обмеження для бази даних SQLServer

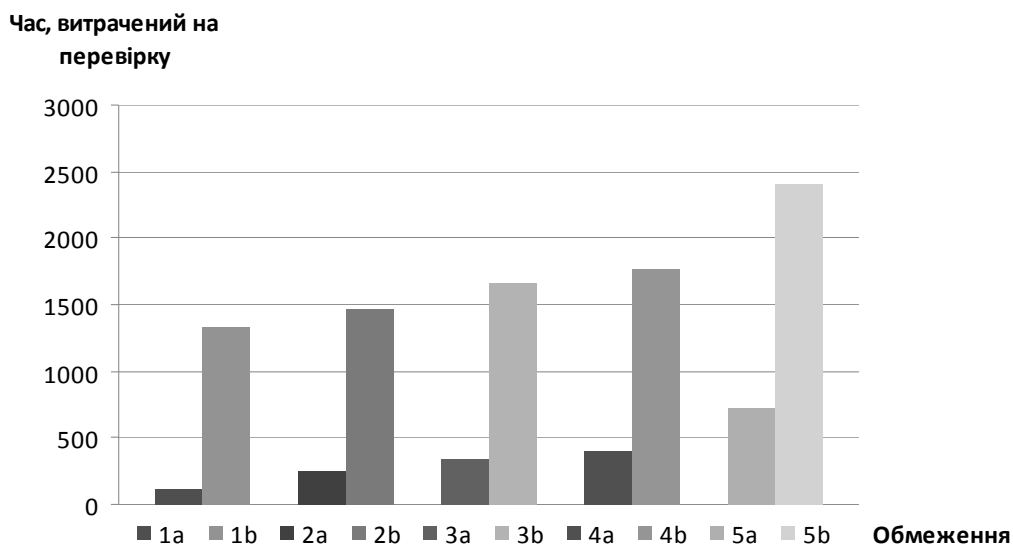


Рис. 2. Обмеження для бази даних SQLite

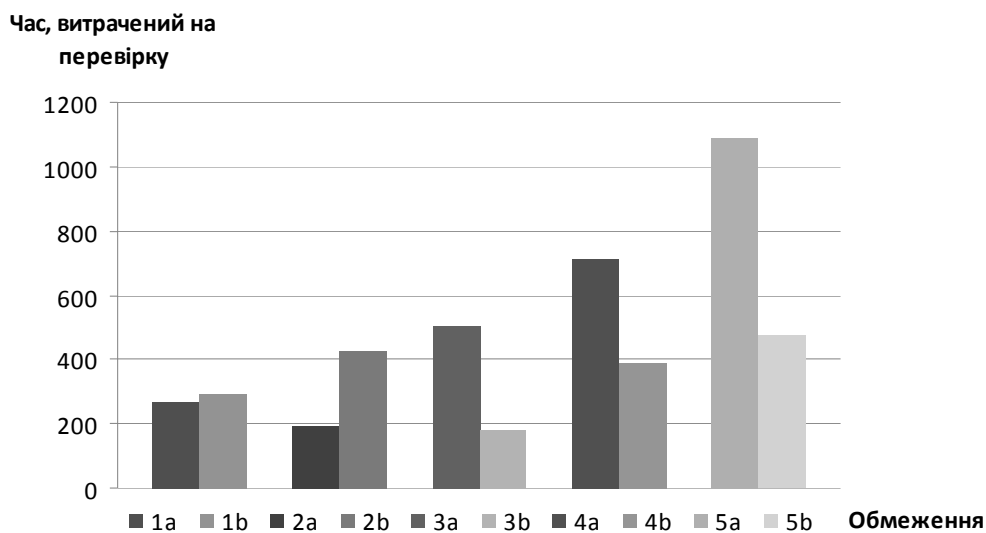


Рис. 3. Обмеження для бази даних PostgreSQL (Windows XP)

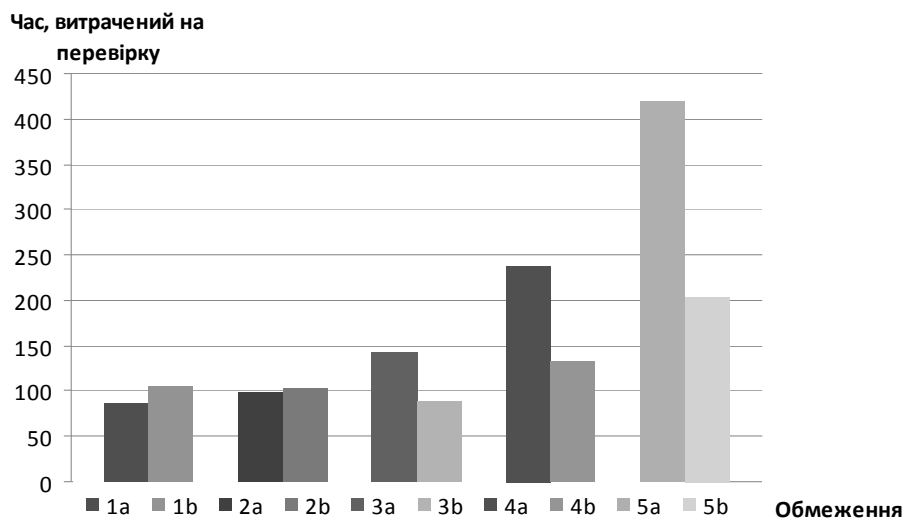


Рис. 4. Обмеження для бази даних PostgreSQL (Ubuntu 12.04)

### Література

1. Дейт, К.Дж. Введение в системы баз данных [Текст] / К.Дж. Дейт. – М. : Вильямс, 2001. – 354 с.
2. Кириллов, В.В. Введение в реляционные базы данных [Текст] / В.В. Кириллов. – СПб. : БХВ-Петербург, 2008. – 451 с.
3. Кузнецов, С.Д. Базы данных [Текст] / С.Д. Кузнецов. – М. : Академия, 2012. – 496 с.
4. Производительность триггеров [Электронный ресурс]. – Режим доступа : <http://blackball.lv/m/mag/read.aspx?id=27013>. – Дата доступа : 25.11.2014.
5. Флёнов, М. Transact-SQL [Электронный ресурс]. – Режим доступа : <http://www.flenov.info/books.php?contentid=43>. – Дата доступа : 25.11.2014.
6. MySQL and PostgreSQL [Электронный ресурс]. – Режим доступа : [https://www.ibm.com/developerworks/ru/library/os-01\\_sql/](https://www.ibm.com/developerworks/ru/library/os-01_sql/). – Дата доступа : 25.11.2014.
7. Уорсли, Дж. PostgreSQL. Для профессионалов [Текст] / Дж. Уорсли, Дж. Дрейк. – СПб. : Питер, 2003. – 496 с.

УДК 378.1

### ДО ПИТАННЯ ПРО ПЛАНУВАННЯ НАВЧАЛЬНОГО ПРОЦЕСУ ПІДГОТОВКИ БАКАЛАВРІВ ЗА НАПРЯМОМ «АВТОМАТИЗАЦІЯ ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ»

Бойко Т.В., Складаний Д.М.

### К ВОПРОСУ О ПЛАНИРОВАНИИ УЧЕБНОГО ПРОЦЕССА ПОДГОТОВКИ БАКАЛАВРОВ ПО НАПРАВЛЕНИЮ «АВТОМАТИЗАЦИЯ И КОМПЬЮТЕРНО-ИНТЕГРИРОВАННЫЕ ТЕХНОЛОГИИ»

Бойко Т.В., Складанный Д.Н.

### ON THE EDUCATIONAL PROCESS' PLANNING FOR BACHELORS IN "AUTOMATION AND COMPUTER INTEGRATED TECHNOLOGIES"

Bojko T.V., Skladannyu D.M.

Національний технічний університет України «КПІ», м. Київ, Україна,  
[kxtp@list.ru](mailto:kxtp@list.ru)

*Проаналізовано систему планування навчального процесу в частині дисциплін професійної та практичної підготовки бакалаврів за напрямом Автоматизація та комп'ютерно-інтегровані технології. Як розширення структурно-логічної схеми, запропоновано ланцюгову схему планування таких дисциплін.*

**Ключові слова:** вища освіта, планування навчального процесу

*Проанализирована система планирования учебного процесса в части дисциплин профессиональной и практической подготовки бакалавров по направлению Автоматизация*