

ДИНАМІЧНЕ ФОРМУВАННЯ ЗАПИТУ ІЗ ВИКОРИСТАННЯМ SQL/OLAP У T-SQL ПРИ АВТОМАТИЗАЦІЇ КРОСС-ТАБЛИЦЬ НА БАЗІ РЕЛЯЦІЙНИХ ДЖЕРЕЛ ДАНИХ

У статті в узагальненому вигляді розглядається архітектура сучасної інформаційно-аналітичної системи. Особливу увагу приділяється складнощам, пов'язаним із розробкою, впровадженням та підтримкою ІУС на базі СД. Під час аналізу підходів щодо реалізації аналітичної обробки даних без фізичного сховища даних розглянуто такі підходи, як вітрини даних, віртуальне сховище даних та підходи, пов'язані з використанням настільних OLAP-систем, наведено перелік переваг та недоліків кожного з них.

На підставі виконаного аналізу щодо реалізації аналітичної обробки даних без фізичного сховища із наведенням переваг та недоліків кожного з них, автором висувається гіпотеза, що використання концепції сховищ даних та OLAP є в більшості випадків неприйнятним для організацій з невеликим прибутком. Задача дослідження сформульована в розробці підходу щодо реалізації системи онлайн аналітичної обробки даних, яка б використовувала переваги попереднього розрахунку агрегатів, але не була б прив'язана до сховищ даних.

Перший етап реалізації поставленої задачі полягає в динамічному формуванні SQL-запиту при онлайн перебудові аналітичного звіту. На наступному етапі отримані дані мають бути перетворені в кросс-таблицю із обчисленням проміжних агрегатів. Запропонований до розгляду підхід реалізації підсистеми онлайн аналітичної обробки даних має на меті використання переваг попереднього розрахунку агрегатних даних із мінімальними часовими та фінансовими затратами без використання концепції сховищ даних та багатовимірних БД.

Ключові слова: сховище даних; багатовимірна модель; оперативно-аналітичний аналіз; звідна таблиця; SQL-запит; розраховані агрегати; реляційна модель.

Постановка проблеми

Завданнями будь-якої інформаційно-аналітичної системи є ефективне зберігання, обробка і аналіз даних. Архітектура сучасної інформаційно-аналітичної системи (ІАС) організації в узагальненому вигляді може бути представлена у вигляді наступних рівнів:

- збір і первинна обробка даних;
- витягання, перетворення і завантаження даних;
- складування даних;
- представлення даних у вітринах даних;
- аналіз даних.

До першого рівня архітектури ІАС відносяться джерела даних, що, як правило, іменуються транзакційними або операційними джерелами (базами) даних, що є частиною так званих oltp-систем (*online transactional processing*). Інформація в таких базах даних орієнтована на конкретне застосування і управляється транзакціями, вона сильно деталізована і часто коректується. Транзакційні бази даних відмінно справляються з обсягом повсякденної інформації, яка повинна

рутинно оброблятися щодня, але не дозволяють отримати загальну картину положення справ в організації в цілому і рідко можуть служити джерелами для проведення комплексного аналізу.

До останнього рівня архітектури ІАС організації відносяться сучасні програмні засоби, що іменуються інструментами інтелектуального або ділового аналізу даних (*Business Intelligence (BI) Tools*), або BI-інструменти. Дані можуть бути організовані у вигляді багатовимірних баз даних OLAP (*Online Analytical Processing*), де довідкова інформація представляється у вигляді вимірів, а кількісна – у вигляді показників. Інформація в багатовимірній моделі представляється у вигляді, максимально доступному кінцевим користувачам, що дозволяє істотно знизити час на здобуття потрібної для прийняття рішень інформації.

Наявність інших етапів зумовлено прагненням об'єднати в одній архітектурі системи підтримки прийняття рішень (СППР) можливості OLTP-систем і систем аналізу.

Серед складнощів реалізації та підтримки ІУС на базі сховища даних (СД) можна виділити наступні:

- необхідність інтеграції даних з неоднорідних джерел в розподіленому середовищі;
- потреба в ефективному зберіганні і обробці дуже великих об'ємів інформації;
- необхідність наявності багаторівневих довідників метаданих;
- підвищені вимоги до безпеки даних.

Отже, реалізація концепції сховищ даних, яка об'єднує в собі етапи витягання, перетворення, завантаження даних із оперативних джерел даних (ОДД), накопичення даних у єдиному СД із подальшим представленням у вигляді багатовимірної моделі вимагає великих часових та фінансових витрат на впровадження такого роду систем.

Подібні витрати, що включають у себе придбання потужних апаратних серверних ресурсів, специфічного програмного забезпечення (для організації СД, OLAP та BI-аналіза) та їх налаштування та інтеграції із наявними системами можуть собі дозволити лише крупні компанії. Організації з більш обмеженими можливостями, включаючи бюджетні (в сучасних економічних умовах), де серверне обладнання не оновлюється десятиріччями, і клієнтські платформи за своїм оснащенням інколи переважають серверні, потребують іншого підходу до реалізації аналітичної складової інформаційної системи.

Аналіз останніх досліджень і публікацій

Серед підходів вирішення проблем, пов'язаних із інтеграцією даних у єдиному СД, розглядаються підходи щодо використання віртуального сховища даних, вітрин даних або клієнтських псевдо-OLAP систем, що працюють без використання переваг попереднього обчислення агрегатів чи багатовимірного представлення даних. Наведемо короткий огляд кожного з підходів.

1. Вітрина даних (ВД) – це спрощений варіант СД, що містить лише тематично об'єднані дані. ВД максимально наближена до кінцевого користувача і містить дані, тематично орієнтовані на нього (наприклад, ВД для працівників відділу маркетингу може містити дані, необхідні для маркетингового аналізу). ВД істотно менше за об'ємом, чим СД, і для її реалізації не вимагається великих витрат. Вони можуть бути реалізовані як самостійно, так і разом з СД.

Самостійні ВД часто з'являються в крупних організаціях з великою кількістю незалежних підрозділів, що вирішують власні аналітичні завдання. Перевагами такого підходу є:

- проектування ВД для відповідей на певний круг питань;
- швидке впровадження автономних ВД і здобуття віддачі;
- спрощення процедур заповнення ВД і підвищення їх продуктивності за рахунок обліку потреб певного круга користувачів.

Недоліками автономних ВД є:

- багатократне зберігання даних у різних ВД, що призводить до збільшення витрат на їх зберігання і потенційних проблем, зв'язаних з необхідністю підтримки несуперечності даних;

- відсутність консолідованості даних на рівні предметної області, а отже – відсутність єдиної картини.

2. Надмірність інформації можна звести до нуля, використовуючи віртуальне СД. В даному випадку на відміну від класичного (фізичного) СД дані з ОДД не копіюються в єдине сховище. Вони витягуються, перетворюються і інтегруються безпосередньо при виконанні аналітичних запитів в оперативній пам'яті комп'ютера. Фактично такі запити безпосередньо адресуються до ОДД. Основними перевагами віртуального СД є:

- мінімізація об'єму пам'яті, займаної на носії інформації;
- робота з поточними, деталізованими даними.

Проте такий підхід має багато недоліків:

- час обробки запитів до віртуального СД значно перевищує відповідні показники для фізичного сховища.
- структури оперативних БД, розраховані на інтенсивне оновлення одиночних записів, у високій мірі нормалізовані.
- інтегрований погляд на віртуальне сховище можливий лише при виконанні умови постійної доступності всіх ОДД.
- виконання складних аналітичних запитів над ОДД займає великий обсяг ресурсів комп'ютерів, на яких вони працюють.
- неможливість отримання даних за довгий період часу.

3. Іншим підходом може бути використання клієнтських псевдо-OLAP або так званих настільних OLAP систем. Прикладами можуть бути реалізовані готові компоненти у візуальних мовах програмування, такі наприклад, як Decision Cube в Borland Delphi, кросстаб звіт у MS Access або звичайний Pivot-table у MS Excel. Серед переваг даного підходу виділемо:

- відсутність необхідності у багатовимірній моделі даних з боку сервера;
- зменшення навантаження на сервер або його відсутність як такого;
- максимальна простота реалізації на мінімальна кількість задіяних ресурсів.

Недоліками даного підходу є:

- повільність перебудови звіту при виконанні операцій консолідації, деталізації, обернення та зрізів через відсутність збереження розрахованих агрегатів;
- неможливість використання для аналізу великих обсягів інформації через високе навантаження на клієнтську частину.

Постановка завдання

Аналіз переваг та недоліків різних підходів представлення даних у підсистемі аналітичної обробки даних інформаційно-управляючої системи дозволяє зробити висновок, що використання концепції сховищ даних та OLAP із використанням специфічного програмного забезпечення для BI-аналізу є в більшості випадків неприйнятним для бюджетних або комерційних організацій з невеликим прибутком.

Проведення аналітичної обробки даних із використанням програмних компонент з боку клієнта не використовує переваг багатовимірної моделі, що полягає

в збереженні заделегідь розрахованих агрегатів. Це призводить до надмірних часових втрат при виконанні операцій консолідації, деталізації, обернення та зрізів, а одже фактично унеможлиблює онлайн аналіз даних.

Задача дослідження полягає у розробці підходу щодо реалізації системи онлайн аналітичної обробки даних, яка б використовувала переваги попереднього розрахунку агрегатів, але не була б прив'язана до дорогих систем BI-аналізу на базі сховищ даних.

Виклад основного матеріалу

Для реалізації OLTP-підсистеми вводу та зберігання інформації може бути використана будь-яка система керування базами даних, що підтримує стандарт ANSI SQL-92. Це дозволяє інтегрувати наступну підсистему аналізу майже в будь-якою клієнт-серверною ІС, що вже використовується на підприємстві.

Для тестування запропонованого підходу було розроблено дві ІС: «Облік та аналіз даних температурних датчиків» та «Облік результатів контролю знань студентів». Далі наведемо короткий огляд першої із них. Система дозволяє зберігати відомості по показанням температурних датчиків, розмішеним на теплових елементах системи опалення. Кожен датчик характеризується своїм положенням, яке виражається у прив'язці до контура опалення, корпусу, аудиторії, зовнішнього/або внутрішнього положення. Дані зберігаються із підтримкою хронології, тобто зберігається дата та час зняття показань датчика.

Виходячи із наведеного вище переліку даних, система дозволяє аналізувати температурні показники датчиків в аналітиці їх положення (Корпус, аудиторія), контуру системи опалення, типу датчика та моменту часу зняття показань (рік, місяць, день, година, хвилина). Зовнішній вигляд розробленого звіту, що дозволяє проводити оперативно-аналітичний аналіз, наведено на рис. 1.

Бази даних інформаційних систем, що використовувались для тестування, були реалізовані на платформі MS SQL Server 2008, але при вибірці даних та їх подальші обробці навмисно не використовувалися специфічні команди Transact-SQL, що зробило б неможливим перехід на іншу СКБД. Структура бази даних системи обліку та аналізу даних температурних датчиків наведено на рис.2. Таблицею фактів є DatchRows, в якій зберігаються дані, що потрапляють безпосередньо від датчиків. У таблиця Datchik зберігається перелік наявних датчиків, і вона, фактично, є зв'язкою між таблицею фактів та таблицями вимірів. Таблиці Korpus, Aud, Kontur та Tur є таблицями вимірів та зберігають відомості по корпусах, аудиторіях, контурах опалення та типах датчиків відповідно. Структура БД має тип «сніжинка», оскільки вимір «положення» представлений двома таблицями «Аудиторії» та «Корпуса».

Перший етап реалізації поставленої задачі полягає у динамічному формування SQL-запиту при онлайн перебудові аналітичного запиту. На першому етапі формується остовний запит «Main_Select_Stmt»:

```
Select T1.field as dim1, T2. field as dim2, ...,
TN.field as dimN,
```

```
T(N+1).field mes1, ..., T(N+K). field mesK
```

```
From T1 join T2 ... join TN ... join T(N+K) on <join
fields list>
```

де T1, ..., TN, ..., T(N+K) – таблиці підсистеми збереження даних.

У розділі select атрибути розділені на 2 категорії – виміри та міри. До кожного виміру підєднаний запит «Dim_Select», що визначає множину значень елементу виміру, а кожна міра пов'язана із функцією агрегації.

Запит «Main_Select_Stmt» та множина запитів «Dim_Select» задаються одноразово та є специфічними для кожної окремої ІС. Їх склад залежить від специфіки представлення даних у тій чи іншій БД, але після одноразової розробки є статичними.

Користувач test

Покази датчиків

[Гол. меню](#)

Аудиторія	Корпус	Контур	Тип	Рік	Місяць	День	Година	Хвилина
1-101 ▲ 2-201 ▼	Головний корпус 2 ▲ Дит.садок Кут.акіма Філологія Головний корпус 1 ▼	Перший контур опалення ▲ Другий контур опалення ▼	Внутренний ▲ Внешний ▼	2013 ▲ 2014 ▼ 2015 ▼	січень ▲ лютий ▼ березень ▼ квітень ▼ травень ▼	01 ▲ 02 ▼ 03 ▼ 04 ▼ 05 ▼	05 ▲ 06 ▼ 07 ▼ 08 ▼ 09 ▼	00 ▲ 01 ▼ 02 ▼ 03 ▼ 04 ▼

У рядки У колонки У комірки

Аудиторія / Корпус	Головний корпус 1	Головний корпус 2	Всього
1-101	15.50		15.50
2-201		16.70	16.70
Всього	15.50	16.70	16.10

Рис. 1. Зовнішній вигляд звіту «покази датчиків» ІС «Облік та аналіз даних температурних датчиків»

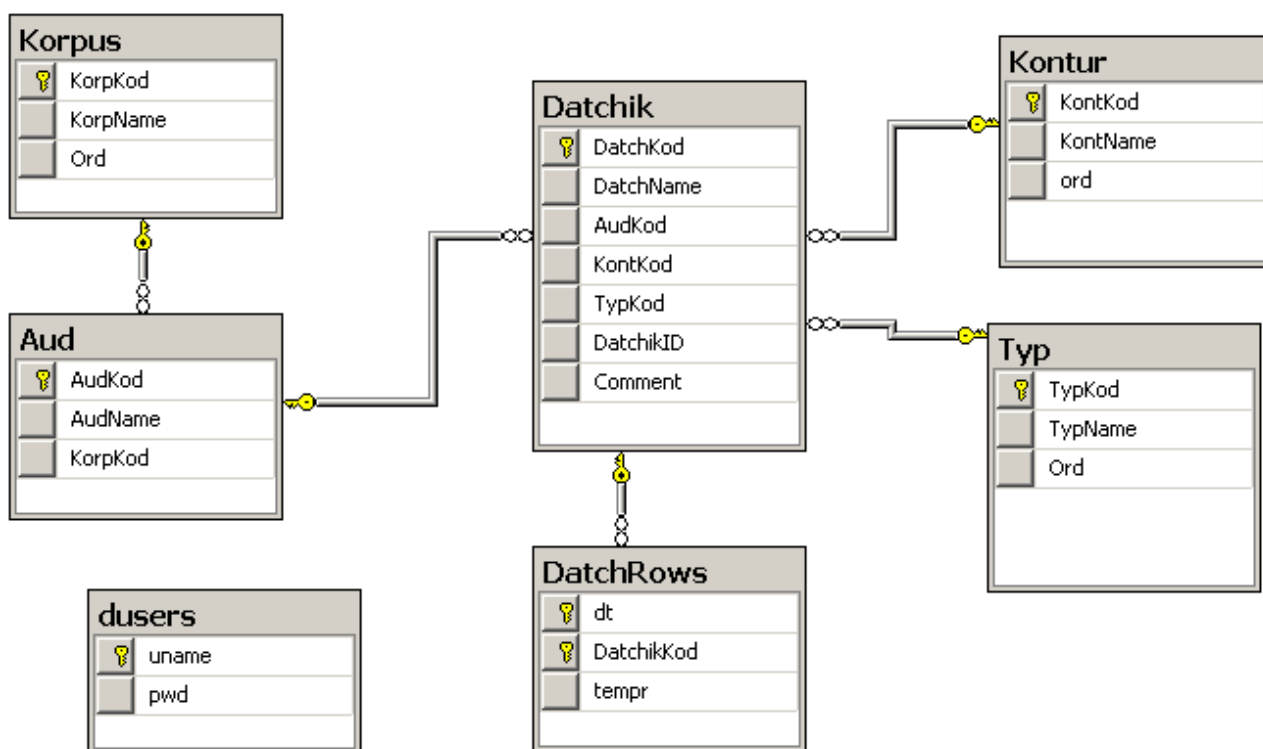


Рис. 2. Логічна архітектура БД показань датчиків температури елементів опалення

При формуванні кінцевої кросс-таблиці дані визначаються наступними критеріями: вимір рядків; вимір колонок; міра комірок; перелік фільтрів по вимірах. У результуючому запиті динамічно формуються розділи select, where, group by та having. Розділ from фактично є статичним та базується на «Main_Select_Stmt». Розділ select визначається виміром рядків, виміром колонок та мірою комірок. Розділ group by – виміром рядків та виміром колонок. У розділ where включаються виміри, що не вийшли до колонок чи рядків, в розділ having – ті, що вийшли. Результуючий запит виглядає таким чином:

```
Select ВимірКолонок, ВимірРядків, ФункціяАгрегаціїМіриКомірок(МіраКомірок)
```

```
From (Main_Select_Stmt )
```

```
Where ФільтрПоВимірах1
```

```
Group by ВимірКолонок, ВимірРядків
```

```
Having ФільтрПоВимірах2
```

де ФільтрПоВимірах: Вимір {=<|>|<>|in} {значення|масив значень}{,... N]

ФільтрПоВимірах2 включає ВимірКолонок та ВимірРядків

ФільтрПоВимірах1 включає всі інші виміри

На наступному етапі отримані дані мають бути перетворені у кросс-таблицю із обчисленням проміжних агрегатів. Це може бути досягнуте шляхом заповнення у циклі асоціативного масиву по результату роботи динамічно сформованого масиву з боку клієнтського додатку. Але основним суттєвим недоліком даного підходу є майже стовідсоткове збільшення обчислювального навантаження на клієнтську платформу.

Альтернативою цьому підходу може бути застосування операторів стандарту SQL/OLAP. Проміжні агрегати можуть бути розраховані за допомогою розширення речення group by операторами cube, rollup та

grouping sets. Оператор cube дає найбільш повний результат, але є найбільш ресурсоємким, grouping sets може бути налаштований під конкретну задачу, але підтримується не в усіх діалектах. З цих причин використовується саме rollup (хоча він не завжди є найоптимальнішим).

Оператор PIVOT може бути застосований для представлення результату кінцевого запиту у вигляді сводної таблиці. Суттєвим недоліком виступає той факт, що при застосуванні цього оператора зєдегідь має бути відомий перелік значень виміру колонок. Але цей недолік може бути подоланий шляхом знову ж таки динамічного формивання запиту через попередню його обробку в асоціативному масиві.

Висновки

Запропонований до розгляду підхід реалізації підсистеми онлайн аналітичного обробки даних має на меті використання переваг попереднього розрахунку агрегатних даних із мінімальними часовими та фінансовими затратами без використання концепції сховищ даних та багатовимірних БД. Час очікування при первинному формуванні аналітичного звіту перевищує показники систем ВІ-аналізу із використанням СД, але показники при використанні подальших операцій консолідації, деталізації, обернення та зрізів є співставними за рахунок мінімізації наступних звернень до реляційної БД.

При виконанні наступних операцій аналітичної обробки на клієнтах із невисокою подужністю обчислювальних ресурсів підвищення продуктивності може бути досягнуте шляхом використання трьохрівневої архітектури із сервером додатків та тонкого клієнта, наприклад Web-Browser (Тонкий клієнт) – Apache+PHP (сервер додатків) – MS SQL Server (сервер баз даних).

ЛІТЕРАТУРА

1. Петкович Д. Microsoft SQL Server 2008. Руководство для начинающих / Д. Петкович. – БВХ-Петербург, 2009. – 752 с.
2. Барсегян А. А. Методы и модели анализа данных: OLAP и Data Mining / А. А. Барсегян, М. С. Куприянов, В. В. Степаненко, И. И. Холод. – Петербург, 2004. – 336 с.
3. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Томас Коннолли, Карелии Бегг ; 3-е издание. : [пер. с англ.]. – М. : Издательский дом «Вильямс», 2003. – 1440 с.
4. Microsoft SQL Server 2005 Analysis Services. OLAP и многомерный анализ данных. – «БХВ-Петербург», 2007. – 928 с.
5. Байдачный С. SQL Server 2005. Новые возможности для разработчиков / С. Байдачный, Ю. Лозинский, Д. Маленко. – «Солон», 2006. – 208 с.
6. Татарчук М. І. Корпоративні інформаційні системи : [навч. посібник] / М. І. Татарчук. – К. : КНЕУ, 2005. – 291 с.
7. Нестеров С. А. Базы данных. Интеллектуальный анализ данных : [учеб. пособие] / С. А. Нестеров. – СПб. : Изд-во Политехн. ун-та, 2011. – 272 с.

Дворецкий М. Л., Черноморский государственный университет имени Петра Могилы, г. Николаев, Украина

ДИНАМИЧЕСКОЕ ФОРМИРОВАНИЕ ЗАПРОСА С ИСПОЛЬЗОВАНИЕМ SQL/OLAP В T-SQL ПРИ АВТОМАТИЗАЦИИ КРОСС-ТАБЛИЦ НА БАЗЕ РЕЛЯЦИОННЫХ ИСТОЧНИКОВ ДАННЫХ

В статье в обобщенном виде рассматривается архитектура современной информационно-аналитической системы. Отдельное внимание уделяется сложностям, связанным с разработкой, внедрением и поддержкой ИУС на базе ХД. Во время анализа подходов относительно реализации аналитической обработки данных без физического хранилища данных рассмотрены такие подходы, как витрины данных, виртуальное хранилище данных и подходы, связанные с использованием настольных OLAP-систем, приведен перечень преимуществ и недостатков каждого из них.

На основании выполненного анализа относительно реализации аналитической обработки данных без физического хранилища с приведением преимуществ и недостатков каждого из них, автором выдвигается гипотеза, что использование концепции хранилищ данных и OLAP является в большинстве случаев неприемлемым для организаций с небольшой прибылью. Задача исследования сформулирована в разработке подхода относительно реализации системы онлайн аналитической обработки данных, которая бы использовала преимущества предыдущего расчета агрегатов, но не была бы привязана к хранилищам данным.

Первый этап реализации поставленной задачи заключается в динамическом формировании SQL-запроса при онлайн перестройке аналитического отчета. На следующем этапе полученные данные должны быть преобразованы в кросс-таблицу с вычислением промежуточных агрегатов. Предложенный к рассмотрению подход реализации подсистемы онлайн аналитической обработки данных имеет целью использование преимуществ предыдущего расчета агрегатных данных с минимальными временными и финансовыми затратами без использования концепции хранилищ данных и многомерной БД.

Ключевые слова: *хранилище данных; многомерная модель; оперативно-аналитический анализ; сводная таблица; SQL-запрос, рассчитанные агрегаты; реляционная модель.*

Dvoretzkiy M., Petro Mohyla Black Sea State University, Mikolaiv, Ukraine

DYNAMIC CREATION OF SQL QUERY WITH USING OF SQL/OLAP IN T-SQL ATTACHED TO CROSS TABLES BASED ON RELATIONAL DATA SOURCES

In the article in the generalized kind is examined the architecture of modern informational (and research) system. For realization of that in a complete measure it is necessary to use conception of data warehouse and multidimensional model of data. Separate attention is spared to the complications, related to development, introduction and support of decision support system based on Data Warehouses. It is bond with time and financial losses. Author makes on assume that organizations with limit possibilities, including budgetary, need other realization of informational system analytical tools.

During the analysis of analytical processing of data realization approaches without the physical data warehouses, are considered data marts, virtual data warehouses and approaches based on desktop-OLAP systems. List of advantages and defects each of them is brought.

Based on researches, author pulls out a hypothesis that the use of data warehouse conception, OLAP and specific applications for BI-analysis is in most cases unacceptable for budgetary (or commercial) organizations with a small profit. A research task is set forth as development approaches of the system on-line analytical processing of data realization, that would take previous calculation of aggregates advantage, but would not be tied to the expensive BI-analysis systems based on data warehouses.

The first step of the task solution is forming of SQL-query linked to on-line building of analyst report. The data obtained on the next step must be presented in a cross-table with the intermediate aggregates calculation. Approach of on-line analytical processing of data subsystem realization, with is suggested, uses advantage of previous data aggregate calculation. Besides, it has minimum time and financial expenses because of non using data warehouses and multidimensional databases.

Key words: *data warehouse; multidimensional model; on-line analytical processing; cross-table; SQL-query; calculated aggregates; relational model.*

© Дворецький М. Л., 2014

Дата надходження статті до редколегії 16.12.2014 р.