УДК 004.056:004.057.3
DOI: 10.18372/2073-4751.76.18236

[1]Жигаревич О.К.,
orcid.org/0000-0002-7154-9733,
[2]Бердибаєв Р.Ш., PhD,
orcid.org/0000-0002-8341-9645,
[1]Сидоренко В.М., к.т.н.,
orcid.org/0000-0002-5910-0837,
[1]Положенцев А.А.,
orcid.org/0000-0003-0139-0752,
[3]Кримська А.О.,
orcid.org/0000-0001-6410-9476

МОДЕЛЬ ОНТОГОЛІКО-РЕЛЯЦІЙНОГО СХОВИЩА ДАНИХ ДЛЯ ФУНКЦІОНУВАННЯ ХМАРНОЇ SIEM-СИСТЕМИ

[1]Національний авіаційний університет
[2]Алматинський університет енергетики та зв'язку
[3]Чернівецький торговельно-економічний інститут
Державного торговельно-економічного університету

[1]v.sydorenko@ukr.net,
[2]r.berdybaev@aues.kz,
[3]ryhz8998@gmail.com

Вступ

Кількість кіберзагроз безперервно зростає і одним із ефективних засобів для їх виявлення та захисту інформації є використання *SIEM*-систем. В основі їх функціонування лежить використання баз даних (БД), тобто структурованих даних, що зберігаються у цифровому вигляді в комп'ютерній системі. Управління БД здійснюється за допомогою системи управління БД (СУБД). Дані разом із СУБД та відповідними додатками утворюють систему БД. Сучасні типи БД зазвичай зберігають дані у формі таблиць, де інформація представлена у вигляді рядків та стовпців. Такою інформацією легко управляти, додавати, редагувати, видаляти, оновлювати, контролювати тощо. Більшість сучасних БД використовують мову структурованих запитів (*SQL*) для внесення записів та отримання інформації.

Аналіз останніх досліджень та постановка завдання

Наразі існує широке різноманіття типів БД [1-3]. Вибір типу БД для певної *SIEM*-системи обумовлений особливостями використання даних в конкретному контексті. Під терміном «типи БД» розуміються шаблони та структури, які використовуються для організації інформації в системах управління БД (СУБД) [1-5].

Метою дослідження є розроблення моделі онтологіко-реляційного сховища даних для застосування у хмарній *SIEM*-системі.

Основна частина дослідження

Для досягнення мети необхідно провести аналіз сучасних типів БД, які використовуються в *SIEM*-системах, для виявлення їх сильних і слабких сторін.

1. Найпростіші типи БД

Спершу, розглянемо три типи БД, які все ще можна зустріти в спеціалізованих середовищах, але які вже значною мірою витіснені надійними та ефективними альтернативами.

1.1. Прості структури даних

Перший і найпростіший спосіб зберігання даних – це текстові файли. Цей метод використовується і сьогодні для роботи з невеликими обсягами інформації. Для розділення полів використовується спеціальний символ: кома або крапка з комою у *csv*-файлах, двокрапка або пробіл у **nix*-подібних системах:

root:*x*:0:0:*root*:/*root*:/*bin*/*bash*
bin:*x*:2:2:*bin*:/*bin*:/*usr*/*sbin*/*nologin*
sys:*x*:3:3:*sys*:/*dev*:/*usr*/*sbin*/*nologin*
sync:*x*:4:65534:*sync*:/*bin*:/*bin*/*sync*
games:*x*:5:60:*games*:/*usr*/*games*:/*usr*/*sbin*/*nologin*

1.2. Ієрархічні БД

На відміну від текстових таблиць, у наступному типі БД з'являються зв'язки між об'єктами. В ієрархічних БД кожен запис має одного попередника (предка). Це створює деревоподібну структуру, в якій записи класифікуються відповідно до їх відношення до більш низького рівня ланцюжка записів, структура ієрархічних БД показана на рис. 1.

1.3. Мережеві БД

Мережеві БД розширюють функціональність ієрархічних БД: записи можуть мати більше одного предка. Це означає, що можна моделювати складні взаємозв'язки, структура мережевих БД показана на рис. 2.

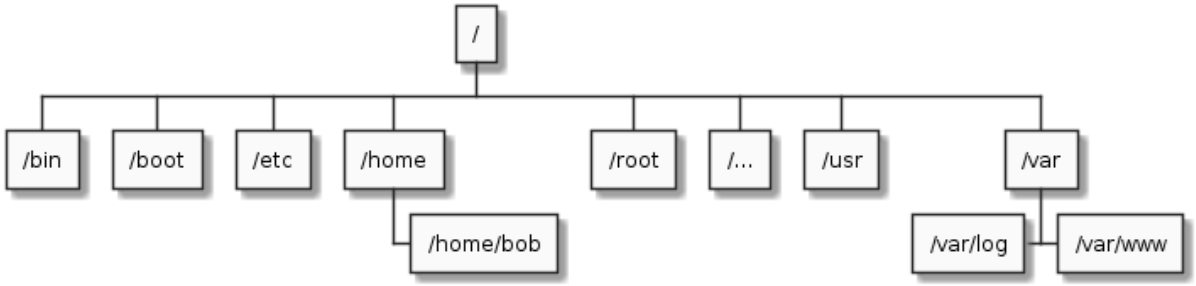

Рис. 1. Структура ієрархічних БД

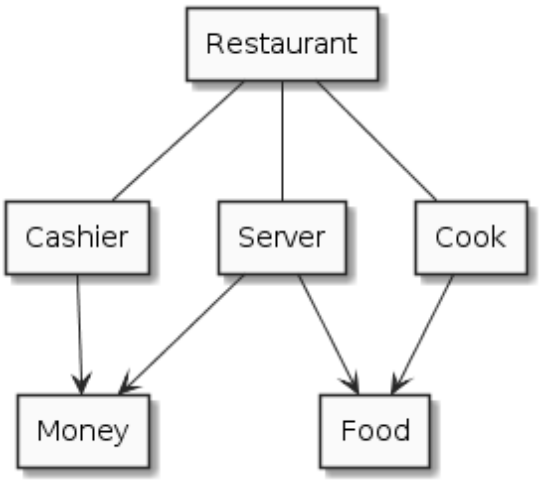

Рис. 2. Структура БД мережі

2. Реляційні БД

2.1. SQL

Реляційні БД є найстарішим і все ще широко використовується типом загального призначення. Дані у реляційних БД структуровані у вигляді таблиць, які складаються зі стовпців та рядків.

Кожен стовпець у таблиці має свою назву та тип даних. Кожен рядок представляє собою окремий запис або елемент інформації в таблиці, який містить значення для кожного зі стовпців. Структуру реляційних БД можна побачити на рис. 3.

2.2. OLTP

OLTP призначена для виконання бізнес-транзакцій, що виконуються декількома користувачами, структура БД *OLTP* показана на рис. 4.

Реляційні БД використовують такі SIEM-системи: *IBM QRadar*, *AlienVault USM*, *LOGRHYTHM*, *AlienVault OSSIM*, *Splunk*, *FortiSIEM*, *Wazuh*, *SolarWinds*, *ManageEngine*, *RuSIEM*, *Prelude OSS*, *Prelude SIEM*, *Sagan*, *Maxpatrol*, *EventTracker*, *Trustwave SIEM Enterprise*, *McAfee* (*ESM*) [6-8].

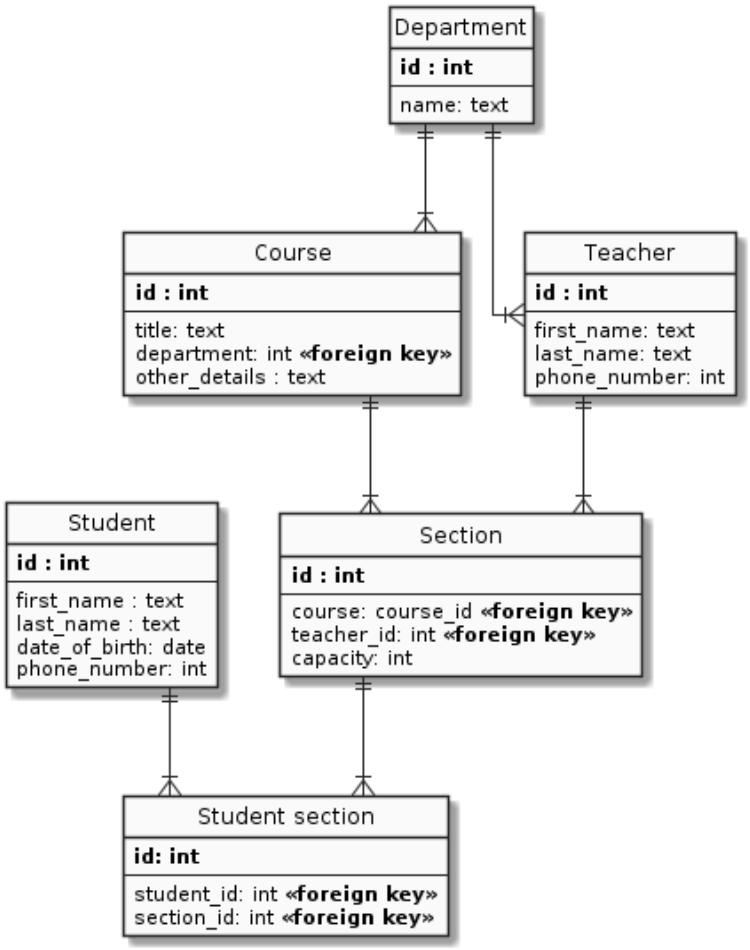

Рис. 3. Структура реляційних БД

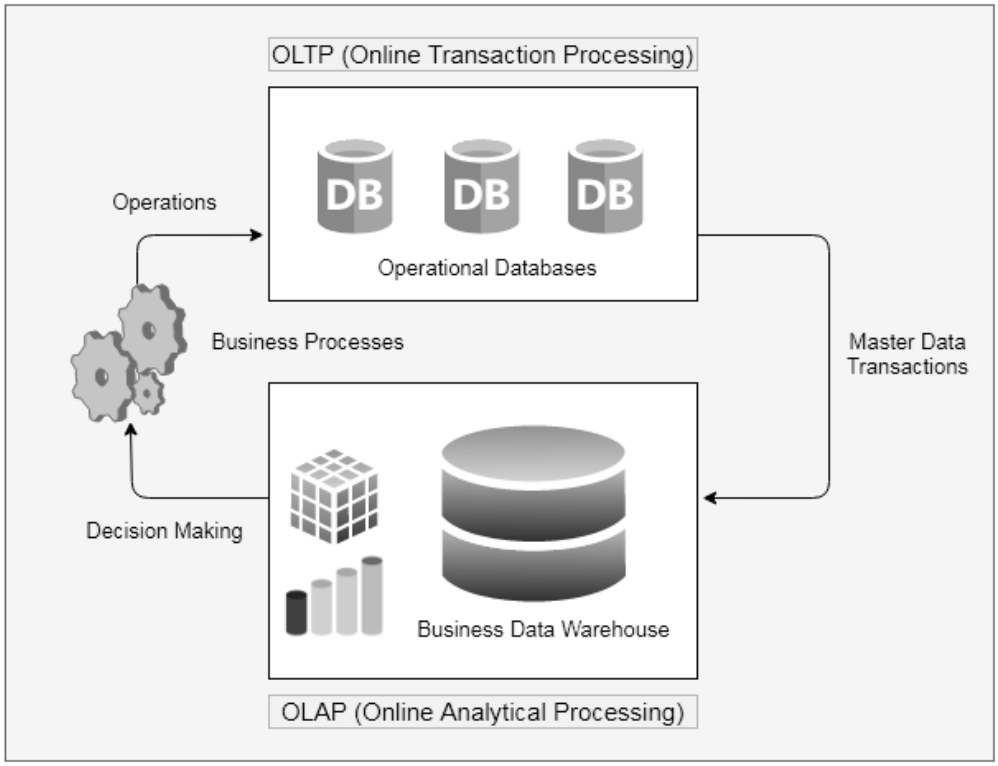

Рис. 4. Структура БД *OLTP*

3. БД NoSQL

NoSQL – це група типів БД, які пропонують підходи, відмінні від стандартної реляційної моделі. *NoSQL* означає «не-*SQL*» або «не тільки *SQL*», щоб пояснити, що іноді дозволяється використовувати *SQL*-подібні запити.

Нереляційна БД *NoSQL* надає змогу зберігати і обробляти неструктуровані або напівструктуровані дані (на відміну від реляційної БД, яка визначає структуру даних, що містяться в ній). Популярність БД *NoSQL* зростає в міру того, як веб-додатки поширюються і стають більш складними.

3.1. БД типу «ключ-значення»

У БД типу «ключ-значення» для зберігання інформації необхідно вказати ключ і об'єкт даних, який необхідно зберегти. Наприклад, *JSON*-об'єкт, зображення або текст. Для запиту даних відправляється ключ і отримується *blob*, структура *NoSQL* показана на рис. 5.

3.2. Документо-орієнтовані БД

Документо-орієнтовані БД (бази документів або сховища документів) поділяють базову семантику доступу та пошуку сховищ ключів і значень. Такі БД також використовують ключ для унікальної ідентифікації даних. Різниця між сховищами ключів-значень і БД документів полягає в тому, що замість зберігання блоків, БД документів зберігають дані в структурованих форматах – *JSON*, *BSON* або *XML*, структура БД документів показана на рис. 6 [9-10].

3.3. Графові БД

Замість того, щоб відображати зв'язки за допомогою таблиць і зовнішніх ключів, графові БД встановлюють зв'язки за допомогою вузлів, ребер та властивостей. Структура графових БД представлена на рис. 7. Графові БД представляють дані у вигляді окремих вузлів, які можуть мати будь-яку кількість пов'язаних з ними властивостей. Графові БД зберігають дані в контексті сутностей і зв’язків між сутностями.

3.4. Стовпчикові БД

Стовпчикові (колонкові) БД (рис. 8) також відомі як нереляційні сховища стовпців або БД з широкими стовпцями, відносяться до категорії *NoSQL*-систем, проте зовні вони схожі на реляційні БД.

Подібно до реляційних, стовпчикові БД зберігають дані у вигляді рядків та стовпців, але мають відмінну структуру співвідношення між елементами.

У реляційних БД всі рядки повинні відповідати фіксованій схемі. Схема визначає, які стовпці будуть у таблиці, їх типи даних та інші характеристики. У стовпчастих БД, натомість, існують структури, відомі як «сім'ї стовпців», замість таблиць. Сім'ї стовпців містять рядки, кожен із яких може мати свій власний формат. Кожен рядок складається з унікального ідентифікатора, що використовується для пошуку, за яким слідують набори імен стовпців та їх значень.

3.5. БД часових рядів

Такі БД призначені для збору та управління елементами, які змінюються з часом. Більшість таких БД організовано у структури, які записують значення для одного елемента. Наприклад, можна створити таблицю для відстеження температури процесора. Усередині кожне значення буде складатися з позначки часу і значення температури. У таблиці може бути кілька метрик; структура БД часових рядів показана на рис. 9.

NoSQL використовують такі SIEM-системи: *AlienVault USM, AlienVault OSSIM, MozDef, Maxpatrol, SearchInform SIEM*.

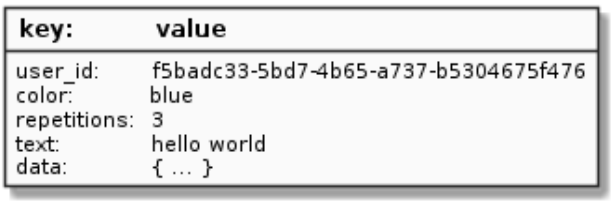

Рис. 5. Структура БД типу «ключ-значення»

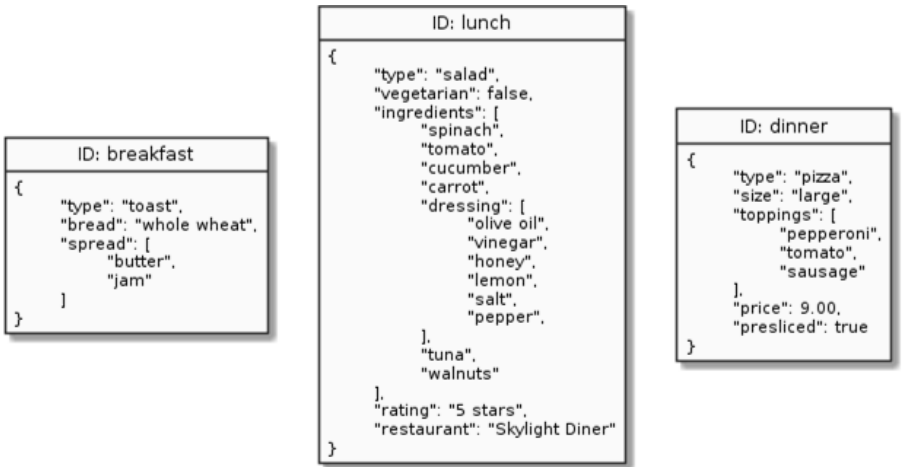

Рис. 6. Структура документо-орієнтованих БД

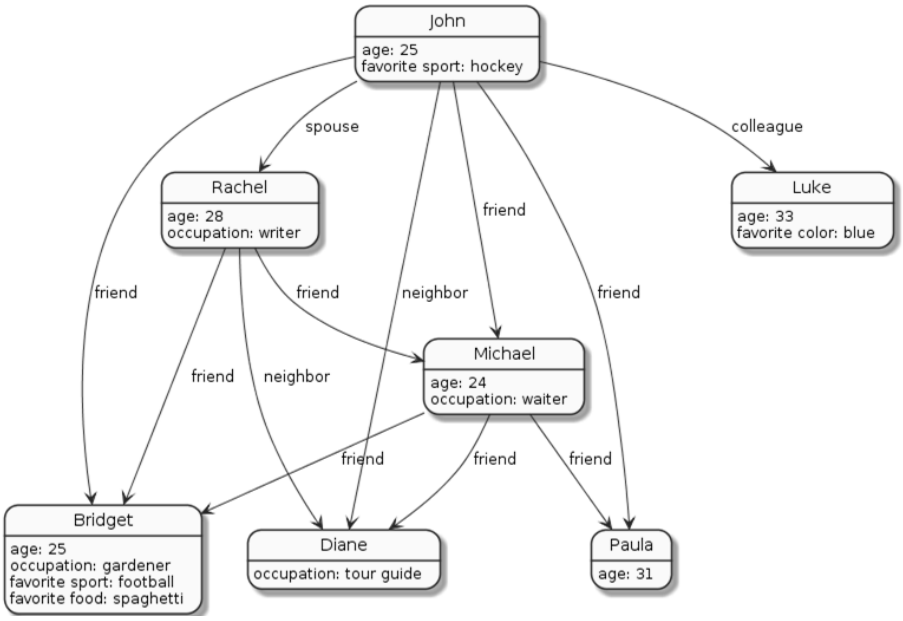

Рис. 7. Структура графових БД

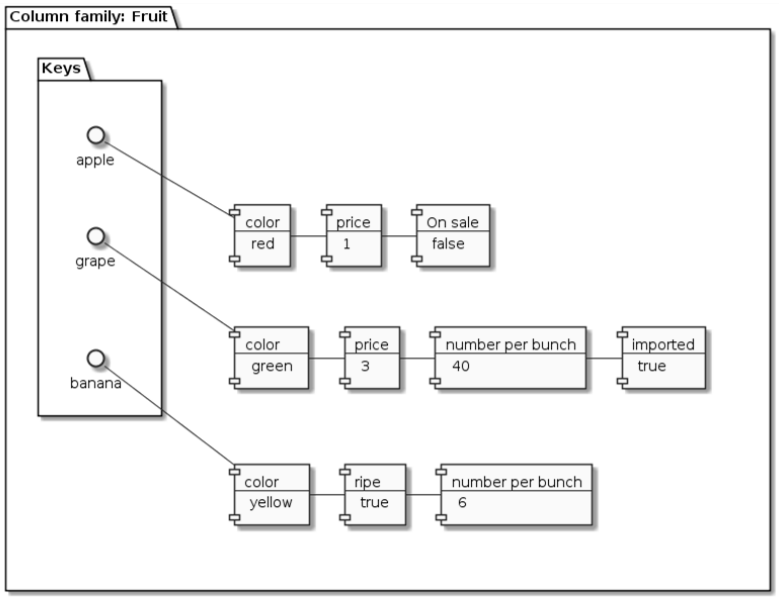

Рис. 8. Структура стовпчикових БД

Time	CPU Temp	System Load	Memory Usage %
2019-10-31T03:48:05+00:00	37	0.85	92
2019-10-31T03:48:10+00:00	42	0.87	90
2019-10-31T03:48:15+00:00	33	0.74	87
2019-10-31T03:48:20+00:00	34	0.72	77
2019-10-31T03:48:25+00:00	40	0.88	81
2019-10-31T03:48:30+00:00	42	0.89	82
2019-10-31T03:48:35+00:00	41	0.88	82

Рис. 9. Структура БД часових рядів

4. Комбіновані БД

NewSQL та багатомодельні БД є різними видами БД, проте вони націлені на розв'язання спільної групи завдань, що виникають внаслідок використання протилежних підходів *SQL* або стратегій *NoSQL*.

4.1. БД NewSQL

БД *NewSQL* успадковують реляційну структуру і семантику, але побудовані з використанням більш сучасних, масштабованих конструкцій. Метою є забезпечення більшої масштабованості, ніж у реляційних БД, і вищих гарантій узгодженості, ніж у *NoSQL*.

Компроміс між узгодженістю та доступністю є фундаментальною проблемою розподілених БД, яка описується теоремою *CAP* [11-12].

4.2. Багатомодельні БД

Такі БД поєднують в собі функціональність декількох типів баз. Переваги такого підходу очевидні – одна і та ж система може використовувати різні представлення для різних типів даних.

Спільне розташування даних з декількох типів БД в одній системі дозволяє виконувати нові операції, які в іншому випадку були б складними або неможливими. Наприклад, мультимодельні БД можуть дозволити користувачам отримувати доступ і керувати даними, що зберігаються в різних типах БД, в рамках одного запиту, а також підтримувати узгодженість даних при виконанні операцій, що змінюють інформацію в декількох системах одночасно [13].

5. Об'єктно-орієнтовані БД

Інформація в об'єктно-орієнтованій БД (ООБД) представлена у вигляді об'єкта, як і в об'єктно-орієнтованому програмуванні, структура ООБД показана на рис. 10.

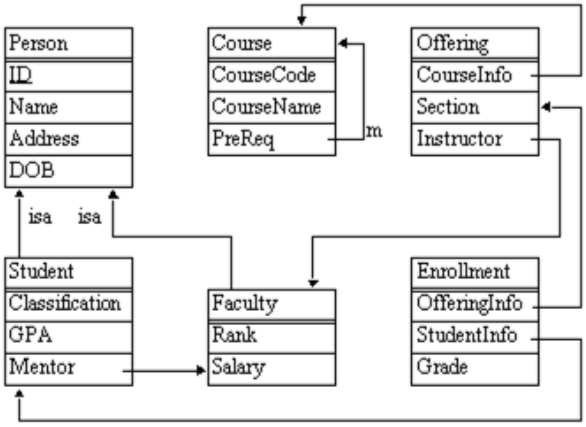

Рис. 10. Структура ООБД

6. Хмарні БД

Хмарна БД є набором структурованих або неструктурованих даних, розміщених на приватній, публічній або гібридній платформі хмарних обчислень [7,14-16]. Існує два типи моделей хмарних БД: традиційна база і БД як послуга (*DBaaS*). У моделі *DBaaS* адміністративні завдання та обслуговування виконує хмарний провайдер, структуру хмарних БД [17-18] показано на рис. 11.

Хмарні типи БД використовують такі SIEM-системи: *HPE ArcSight Splunk Ixia ThreatARMOR, Micro Focus ArcSight, Trustwave SIEM Enterprise*.

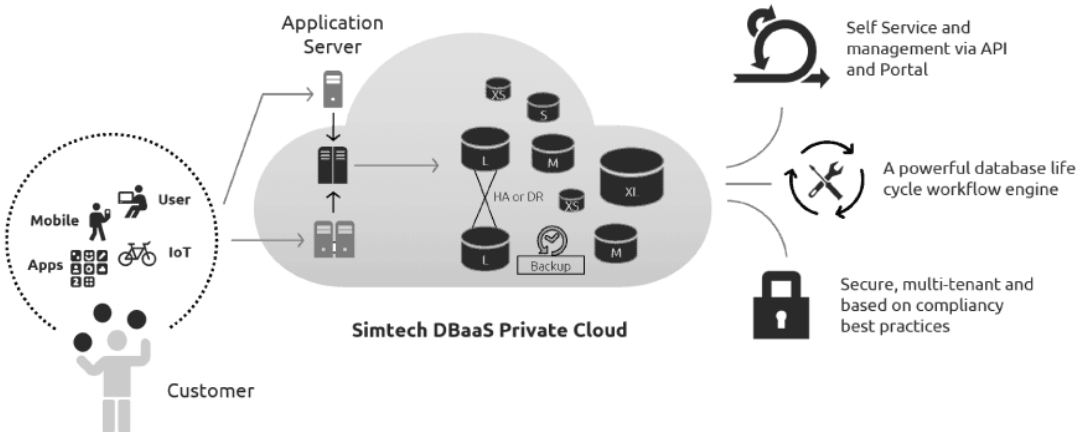

Рис. 11. Структура хмарних БД

Результати аналізу СУБД, які використовуються в різних *SIEM*-системах наведено в табл. 1.

Після проведеного у табл. 1. аналізу слід зазначити, що кожен з цих видів СУБД залишається актуальним у власній сфері, де взаємозв'язки між даними обумовлені конкретною структурою БД.

Крім того, слід розглянути можливість використання гібридних БД, які поєднують у собі різні типи, такі як *SQL* та *NoSQL* – це дозволить зберегти зручність у зберіганні та класифікації даних, а також забезпечити високу швидкість отримання великих обсягів інформації завдяки попередній індексації.

Таблиця 1. Порівняльний аналіз *ESB*

SIEM	**СУБД**
IBM QRadar	*Ariel database, PostgreSQL, SQLite*
LogRhythm	*Oracle, SQL Server, MySQL*
Splunk	*DB2 / Linux, Informix, MemSQL, MySQL, AWS Aurora, Microsoft SQL Server, Oracle, PostgreSQL, AWS RedShift, SAP SQL Anywhere, Sybase ASE, Sybase IQ, and Teradata*
McAfee (ESM)	*MSSQL, Oracle, MySQL, Data Access Server (DAS), DB2 / UDB*
AlienVault USM	*RedisDB, MySQL*
AlienVault OSSIM	*RedisDB, MySQL*
FortiSIEM	*PostgreSQL*
Ixia ThreatARMOR	*Rap Sheet*
MozDef	*RabbitMQ, MongoDB, Elasticsearch, Kibana*
Wazuh	*MySQL, PostgreSQL*
Prelude OSS	*MySQL, PostgreSQL*
Prelude SIEM	*MySQL, PostgreSQL*
Sagan	*MySQL, PostgreSQL*
Maxpatrol	*ElasticSearch, MongoDB, MS SQL Express*
SolarWinds	*MSSQL, Oracle, MySQL, MariaDB.*
ManageEngine	*Oracle, SQL, DB2, MySQL*
EventTracker	*Microsoft SQL Server*
Micro Focus ArcSight	*Own development CORR-E*
Trustwave SIEM Enterprise	*Microsoft SQL Server, Microsoft SQL Azure, ORACLE, SYBASE, MySQL, IBM, DB2, Hadoop*
BlackStratus SIEMStorm	Власна розробка

Для сталого функціонування сучасна *SIEM*-система має оперативно вирішувати кілька облікових завдань. З одного боку, система повинна з максимально можливою швидкістю зберігати, обробляти та шукати події в журналах. З іншого боку, надійно та структуровано зберігати службові дані за користувачами, метадані, налаштування конфігурації, лічильники гешованих потоків та архів попереджень.

Використання однієї БД для всіх цих задач не відповідає вимогам архітектури

та безпеки. Тому, для вирішення цього завдання було необхідно розробити модель гібридного онтологіко-реляційного сховища даних, що базується на спільній роботі двох різних БД з відповідними характеристиками.

Модель онтологіко-реляційного сховища даних (рис. 12) складається з 2 типів БД:

1) Тип БД 1

Призначення: швидке опрацювання журналів.

Для вирішення цього завдання було обрано відкриту технологію *Elasticsearch*. *Elasticsearch* чудово орієнтована на роботу з журналами. Після індексації можливо шукати, сортувати, фільтрувати дані, а не рядки даних у стовпцях. Що, у свою чергу, демонструє інший підхід до пошуку даних, і вказує на те, що *Elasticsearch* може виконувати складний повнотекстовий пошук.

Документи представлені як об'єкти *JSON*. При цьому серіалізація (процес перекладу будь-якої структури даних у послідовність біт) *JSON* підтримується більшістю мов програмування і є стандартним форматом для *NoSQL*.

Elasticsearch – це повнотекстова пошукова платформа з відкритим вихідним кодом, що використовує бібліотеку *Lucene* і написана на *Java*. Вона призначений для складних пошуків на базі документів/файлів. У БД *Elasticsearch* таблиці називаються індексами, а процес завантаження документів – індексування. Можна її вважати і не реляційним сховищем документів у форматі *JSON*, і пошуковою системою на базі повнотекстового пошуку *Lucene*. Офіційні клієнти доступні *Java, NET* (*C#*), *Python, Groovy, JavaScript, PHP, Perl, Ruby*. *Elasticsearch* розробляється компанією *Elastic* і розповсюджується за відкритою ліцензією. Для діючої моделі було доопрацьовано програмний код *Java*.

2) Тип БД 2

Призначення: надійне зберігання службової інформації.

Для виконання цього завдання було обрано відкриту технологію *MongoDB*. *MongoDB* – це документоорієнтована СУБД, яка не вимагає опису схеми таблиць. Вважається одним із класичних прикладів *NoSQL*-систем, що використовує *JSON*-подібні документи та схему БД. Написана мовою *C++*, застосовується у веброзробці, зокрема, у рамках *JavaScript*-орієнтованого стека *MEAN*.

Система може працювати з набором реплік, тобто містити дві або більше копій даних на різних вузлах. Кожен екземпляр набору реплік може будь-якої миті виступати в ролі основної або допоміжної репліки. Усі операції запису та читання за замовчуванням здійснюються з основною реплікою. Допоміжні репліки підтримують копію даних у актуальному стані. У разі коли основна репліка дає збій, набір реплік проводить вибір, яка з реплік має стати основною. Другорядні репліки можуть додатково бути джерелом для операцій читання.

Система масштабується горизонтально, використовуючи техніку сегментування об'єктів БД – розподіл їх частин різним вузлам кластера. Адміністратор вибирає ключ сегментування, який визначає, за яким критерієм дані будуть рознесені вузлам (залежно від значень геша ключа сегментування). Завдяки тому, що кожен вузол кластера може приймати запити забезпечується балансування навантаження. Система може бути використана як файлове сховище з балансуванням навантаження та реплікацією даних (функція *Grid File System*). Крім того, надаються програмні засоби для роботи з файлами та їх вмістом. *GridFS* використовується в плагінах для *Nginx* та *lighttpd*. *GridFS* поділяє файл на частини та зберігає кожну частину як окремий документ. Розповсюджується за відкритою ліцензією *AGPL*.

Реалізація моделі онтологіко-реляційного сховища даних

Запропонована у роботі модель (рис. 12) може бути реалізована у складі системи корелювання подій та управління інцидентами кібербезпеки [20]. При масштабуванні ресурсів у розробленій *SIEM* існує кілька практичних правил:

- вузли *SIEM* орієнтовані на потужність процесора. Вони також служать інтерфейсом користувача для браузера;
- вузли *Elasticsearch* повинні мати якомога більше оперативної пам'яті та найшвидші диски, які можливо використати. Тут все залежить від швидкості введення-виведення;
- *MongoDB* зберігає метаінформацію та дані конфігурації і не потребує багато ресурсів;
- отримані повідомлення зберігаються лише в *Elasticsearch*;

Основним завданням онтологіко-реляційного сховища даних для роботи *SIEM* є суміщення роботи двох типів баз даних та одночасне збереження можливості кластеризації БД обох типів.

Запропонований підхід організації функціонування моделі онтологіко-реляційного сховища даних для *SIEM*-системи дозволяє сервісу індексації отримувати доступ до зовнішніх сховищ даних (при цьому дані коректно індексуються та коректно виводяться при пошуку), проводити масштабування (кластеризацію) при зростанні обсягу даних, підтримувати роботу з різними запитами (прості, складні, структуровані) та з різними типами даних, дозволяє робити агрегацію, проводити аналіз, збирати сутності, закономірності, спрощувати пошук та забезпечувати високу швидкість пошуку.

Крім того, *SIEM* на базі цієї моделі може працювати з набором реплік (тобто містити 2 або більше копії даних на різних вузлах), масштабується горизонтально, використовуючи техніку сегментування об'єктів БД і може бути використана як файлове сховище з балансуванням навантаження і реплікацією даних (функція *Grid File System*).

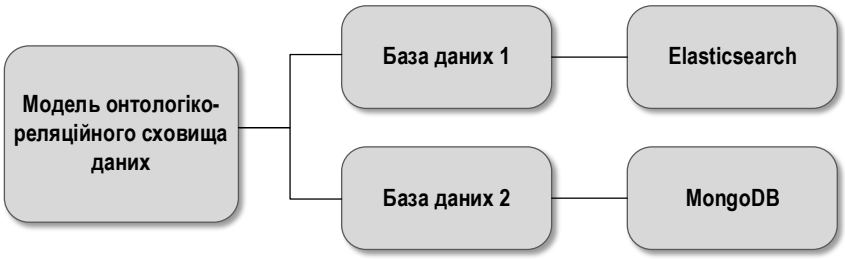

Рис. 12. Схема реалізації запропонованої моделі

Висновки

У роботі проведено аналіз сучасних типів БД, що використовуються в *SIEM*-системах, який показав, що кожен з видів БД залишається актуальним у власній сфері, де взаємозв'язки між даними обумовлені конкретною структурою СУБД. При виборі бази даних для створення *SIEM*-системи важливо враховувати фактори зручності зберігання даних, швидкості їхнього витягнення і використання. Також варто передбачити можливість інтеграції з іншими модулями системи та зовнішніми *API*, щоб підтримувати різноманітні БД для більшості систем *DPI* (комплексних аналізаторів глибокого інспекційного вмісту), як програмних так і апаратних. Крім того, слід розглядати можливість використання гібридних баз даних, які поєднують у собі різні типи, такі як *SQL* та *NoSQL*.

Розроблено модель онтологіко-реляційного сховища даних, яка за рахунок використання двох різних баз даних *Elasticsearch* та *MongoDB* з відповідними характеристиками, дозволяє покращити зручність у зберіганні та класифікації даних, а також забезпечити високу швидкість отримання великих обсягів інформації завдяки попередній індексації, масштабування горизонтально за рахунок сегментування об'єктів БД, а також балансування навантаження і реплікацію даних.

Література

1. Vielberth M., Pernul G. A Security Information and Event Management Pattern. *12th Latin American Conference on Pattern Languages of Programs (SugarLoafPLoP)* / Valparaíso, Chile, 2018. 12 p.

2. Agrawal K., Makwana H. A Study on Critical Capabilities for Security

Information and Event Management. *International Journal of Science and Research (IJSR)*. V. 4. Iss. 7. P. 1893–1896.

3. Henrik Karlzén. An Analysis of Security Information and Event Management Systems. Department of Computer Science and Engineering Chalmers University of Technology University of Gothenburg, Göteborg, Sweden, January 2009. URL: http://publications.lib.chalmers.se/records/fulltext/89572.pdf.

4. Ribolovlev D., Karasov S., Polyakov S. Classification of emergency management systems for incidents without baking. *Food of cyber security*. 2018. No. 3(27). P. 47–53.

5. Ariel Query Language Guide. IBM QRadar 7.3.3 (2013 and 2019). URL: https://www.ibm.com/docs/en/SS42VS_7.3.3/com.ibm.qradar.doc/b_qradar_aql.pdf

6. SIEM Analytcis. URL: http://www.siem.su/compare_SIEM_systems.php

7. Lee J., Kim Y., Kim J., Kim I. Toward the SIEM architecture for cloud-based security services. *2017 IEEE Conference on Communications and Network Security (CNS)* / Las Vegas, USA, 2017. P. 398–399.

8. Bachane I., Adsi Y.I.K., Adsi H.C. Real time monitoring of security events for forensic purposes in Cloud environments using SIEM. *2016 Third International Conference on Systems of Collaboration (SysCo)* / Casablanca, Morocco, 2016. P. 1–3.

9. AlSabbagh B., Kowalski S. A Framework and Prototype for A Socio-Technical Security Information and Event Management System (ST-SIEM). *2016 European Intelligence and Security Informatics Conference (EISIC)* / Uppsala, Sweden, 2016. P. 192–195.

10. Serckumecka A., Medeiros I., Bessani A. Low-Cost Serverless SIEM in the Cloud. *2019 38th Symposium on Reliable Distributed Systems (SRDS)* / Lyon, France, 2019. P. 381–3811.

11. Nabil M., Soukainat S., Lakbabi A., Ghizlane O. SIEM selection criteria for an efficient contextual security. *2017 International Symposium on Networks, Computers and Communications (ISNCC)* / Marrakech, Morocco, 2017. P. 1–6.

12. Mahmoud R.-V., Kidmose E., Turkmen A., Pilawka O., Pedersen J.M. DefAtt - Architecture of Virtual Cyber Labs for Research and Education. *2021 International Conference on Cyber Situational Awareness Data Analytics and Assessment (CyberSA)* / Dublin, Ireland, 2021. P. 1–7.

13. Danik Yu., Hryschuk R., Gnatyuk S. Synergistic effects of information and cybernetic interaction in civil aviation. *Aviation*. 2016. V. 20. No. 3. P. 137–144.

14. Berdibayev R., Gnatyuk S., Yevchenko Yu., Kishchenko V. A concept of the architecture and creation for SIEM system in critical infrastructure. *Studies in Systems, Decision and Control*. 2021. V. 346. P. 221–242.

15. Oksiiuk O., Chaikovska V., Fesenko A. Security Technique for Authentication Process in the Cloud Environment. *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)* / Kyiv, Ukraine, 2019. P. 379–382.

16. Gnatyuk S., Berdibayev R., Avkurova Z., Verkhovets O., Bauyrzhan M. Studies on cloud-based cyber incidents detection and identification in critical infrastructure. *CEUR Workshop Proceedings*. 2021. V. 2923. P. 68–80.

17. Lukova-Chuiko N., Fesenko A., Papirna H., Gnatyuk S. Threat hunting as a method of protection against cyber threats. *CEUR Workshop Proceedings*. 2021. V. 2833. P. 103–113.

18. Astapenya V., Buriachok V., Sokolov V., Skladannyi P., Ageyev D. Last mile technique for wireless delivery system using an accelerating lens. *2020 IEEE International Conference on Problems of Infocommunications Science and Technology (PIC S&T)* / Kharkiv, Ukraine, 2020. P. 811–814.

19. Гнатюк С.О., Сидоренко В.М., Жигаревич О.К., та ін. Система корелювання подій та управління інцидентами кібербезпеки на об’єктах критичної інфраструктури. *Кібербезпека: освіта, наука, техніка*. 2023. Т. 3. № 19. С. 176–196.

Жигаревич О.К., Бердибаєв Р.Ш., Сидоренко В.М., Положенцев А.А., Кримська А.О.

МОДЕЛЬ ОНТОГОЛІКО-РЕЛЯЦІЙНОГО СХОВИЩА ДАНИХ ДЛЯ ФУНКЦІОНУВАННЯ ХМАРНОЇ SIEM-СИСТЕМИ

Системи управління подіями інформаційної безпеки (SIEM) широко використовуються для запобігання втрат інформації у комп'ютерних системах і мережах. Наразі існують різні підходи до створення сховищ (баз) даних для SIEM-систем. В результаті проведеного аналізу не було виявлено універсального типу баз даних, а кожна з них має власні переваги і недоліки. Ця стаття націлена на дослідження відомих типів баз даних і систем управління ними, які можуть бути корисними для реалізації власної моделі сховища даних в сучасних SIEM. У роботі пропонується порівняльна характеристика їхніх можливостей, а також переваг і недоліків. Крім того, в роботі розроблено модель онтологіко-реляційного сховища даних, яка за рахунок використання двох різних баз даних з відповідними характеристиками, дозволяє покращити зручність у зберіганні та класифікації даних, забезпечити високу швидкість отримання великих обсягів інформації завдяки попередній індексації, а також балансування навантаження і реплікацію даних.

Ключові слова: *база даних, система управління базами даних, SIEM, онтологіко-реляційна модель, SQL, NoSQL, NewSQL, балансування навантаження, реплікація даних.*

Zhyharevych O.K., Berdibayev R.Sh., Sydorenko V.M., Polozhentsev A.A., Krymska A.O.

MODEL OF ONTOLOGICAL-RELATIONAL DATA STORAGE FOR THE FUNCTIONING OF A CLOUD SIEM SYSTEM

Information security event management systems (SIEM) are widely used to prevent information loss in computer systems and networks. Currently, there are different approaches to creating data storages (databases) for SIEM systems. The analysis has not revealed a universal type of database, and each of them has its own advantages and disadvantages. This paper is aimed at studying the known types of databases and their management systems that can be useful for implementing your own data storage model in modern SIEM. The paper offers a comparative characterization of their capabilities, as well as advantages and disadvantages. In addition, the paper develops a model of an ontological-relational data warehouse, which, by using two different databases with appropriate characteristics, allows to improve the convenience of storing and classifying data, to ensure high speed of obtaining large amounts of information due to preliminary indexing, as well as load balancing and data replication.

Keywords: *database, database management system, SIEM, ontology-relational model, SQL, NoSQL, NewSQL, load balancing, data replication.*