УДК 004.6

Гуцол Н. А., *здобувач вищої освіти 2 курсу спеціальності 122 Комп'ютерні науки,*
Потапова Н. А., *канд. екон. наук., доцент, доцент кафедри інформаційних технологій*

АВТОМАТИЗОВАНЕ МАСКУВАННЯ ТА АНОНІМНІСТЬ ДАНИХ НА РІВНІ SQL

Донецький національний університет імені Василя Стуса, м. Вінниця

У час швидкого розвитку цифрових технологій та збільшення кількості особистих даних, що зберігаються в базах даних, одним із головних завдань сучасних інформаційних систем стає захист конфіденційності даних зі збереженням їх користі для аналізу. Закони, як-от Загальний регламент про захист даних у Європейському Союзі та подібні правила в інших країнах, підкреслюють важливість впровадження технологій захисту особистої інформації.

Один із дієвих способів захисту конфіденційності даних – використання технологій маскування прямо в SQL-запитах [3]. Цей підхід дає змогу зберігати початкові дані в базі даних, але надавати користувачам та програмам доступ лише до захищених версій цих даних, зменшуючи ризики неправомірного доступу до чутливої інформації. Аналіз сучасних досліджень показує великий інтерес до питань захисту приватності даних. Але практичні аспекти впровадження таких підходів безпосередньо на рівні SQL-запитів залишаються недостатньо висвітленими в наукових публікаціях.

Метою нашого дослідження є розробка методів та практичних порад щодо впровадження автоматичного маскування даних на рівні SQL, що забезпечує захист чутливої інформації без зниження швидкодії систем та без потреби зміни наявної структури баз даних. Існують основні підходи до маскування різних типів даних, що передбачають використання вбудованих функцій SQL для зміни даних під час їх вибірки [1]:

1. Часткове маскування – заміна частини даних символами-замінниками зі збереженням певної частини початкової інформації, як-от перші символи імені чи домен електронної пошти.

2. Узагальнення – зниження рівня деталізації даних (наприклад, показ лише року народження замість повної дати).

3. Заміна даних на псевдоніми – заміна ідентифікаторів на випадкові значення з можливістю зворотної зміни за наявності відповідного ключа.

Розглянемо підхід до побудови системи, що полягає у створенні проміжного шару між базою даних та програмами через використання представлень і функцій, що реалізують логіку маскування [4]. Такий підхід дає змогу:

1. Зберігати початкові дані для адміністративних, перевірочних та статистичних цілей.

2. Надавати різним категоріям користувачів різні рівні доступу до даних із відповідним рівнем маскування.

3. Централізовано керувати правилами маскування та забезпечувати їх однакове застосування.

4. Впроваджувати маскування в наявні системи без потреби зміни структури бази даних.

Експериментальне дослідження ефективності запропонованого підходу було проведене на тестовій базі даних, що містить інформацію користувачів, включно з іменами, контактними даними, датами народження. Результати показали, що:

1. Впровадження маскування на рівні SQL призводить до незначного зниження швидкодії системи – збільшення часу виконання запитів становить у середньому 7–10 %.

2. Використання відповідних індексів дає змогу мінімізувати вплив на швидкодію під час роботи з великими обсягами даних.

3. Застосування розроблених шаблонів маскування дає змогу зберегти аналітичну цінність даних за умови ефективного захисту конфіденційної інформації.

Особливу увагу приділено розробці методів динамічного маскування, що враховують контекст запиту та рівень доступу користувача [4]. Цей підхід ґрунтується на використанні контекстно-залежних функцій, що дає змогу динамічно налаштовувати рівень маскування даних залежно від ролі користувача та мети запиту. Для оцінки ефективності захисту даних запропоновано спосіб кількісного аналізу ризику розпізнавання особи на основі замаскованих даних [2]. Спосіб ґрунтується на обчисленні інформаційної різноманітності даних до та після маскування й оцінці ймовірності успішного витоку особистих даних.

Порівняно з наявними підходами до захисту даних, як-от шифрування на рівні бази даних та фізичне розділення баз даних для різних рівнів доступу, запропонований підхід показує більшу гнучкість, простішу інтеграцію з наявними системами та менші витрати на адміністрування [5].

Подальші дослідження планується спрямувати на розробку автоматизованих інструментів для виявлення чутливих даних у базах даних та створення відповідних правил маскування.

Список використаних джерел

1. Data Masking and Virtualization for Development and Test Environments. *RDTEX – Raw Data Technologies.* URL: https://rdtex.ua/eng/project-solutions-2/data-masking-and-virtualization/ (дата звернення 23.03.2025).

2. Інструменти безпеки в системі керування базами даних (СКБД) MySQL Enterprise Edition. 25.04.2024. URL: https://erc.ua/news-reviews/24517/instrumenti-bezpeki-v-sistemi-keruvannia-bazami-danikh-skbd-mysql-enterprise-edition (дата звернення 13.03.2025).

3. Understanding SQL Server Dynamic Data Masking. DataSunrise Knowledge Center. URL: https://www.datasunrise.com/knowledge-center/sql-server-dynamic-data-masking/ (дата звернення 13.03.2025).

4. Dynamic data masking. *Microsoft Learn.* URL: https://learn.microsoft.com/en-us/sql/relational-databases/security/dynamic-data-masking?view=sql-server-ver16 (дата звернення 23.03.2025).

5. A Basic Guide to SQL Server Security Fundamentals. 10.04.2025. URL: https://corewin.ua/en/blog-en/sql-server-security-basics/ (дата звернення 20.03.2025).