УДК 004.65

КЕРУВАННЯ ПРАВАМИ ДОСТУПУ КОРИСТУВАЧІВ ДО БАЗИ ДАНИХ В HEIDISQL

Хоменко Анастасія

Науковий керівник: канд. екон. наук, доцент Фетісов В.С.

Ніжинський державний університет імені Миколи Гоголя, м. Ніжин, Україна

У статті розглядається функціонал клієнта HeidiSQL в аспекті адміністрування користувачів для захисту конфіденційної інформації у базах даних. Проблема, порушена у статті, має велике значення для різноманітних підприємств, які використовують бази даних і потребують надійного обмеження доступу до них з міркувань безпеки. Метою статті є ознайомлення користувачів з можливостями інструменту HeidiSQL та надання конкретних вказівок щодо ефективного керування доступом користувачів до баз даних за допомогою привілеїв. Такий підхід сприяє забезпеченню безпеки та оптимального використання інструменту для практичних потреб користувачів.

Ключові слова: бази даних, конфіденційна інформація, привілеї користувачів, обмеження доступу користувачів до бази даних.

Management of user access rights to the database in HeidiSQL

A. Khomenko

Scientific supervisor: Candidate of Economic Sciences, Associate Professor

Fetisov V.S.

Mykola Gogol Nizhyn State University, Nizhyn, Ukraine

The article examines the functionality of the HeidiSQL client in terms of user administration to safeguard confidential information in databases. The problem addressed in the article is crucial for various enterprises utilising databases, requiring secure access limitations. The article aims to familiarise users with the capabilities of the HeidiSQL tool and provide specific guidance on effectively

managing user access to databases through privileges. This approach ensures security and optimal utilization of the tool for practical user needs.

Keywords: databases, confidential information, user privileges, restriction of user access to the database.

Бази даних широко застосовуються в багатьох сферах нашого життя, оскільки різного роду підприємства мають потребу в зберіганні та обробці значних об’ємів інформації, а також в швидкому доступі до даних. Найбільш популярними є реляційні бази даних, які зберігають дані в структурованому шаблоні і можуть ідентифікувати взаємозв'язок між збереженими елементами цих даних. Створення, ведення та використання бази даних забезпечується системами управління базами даних. Існує досить багато різноманітних за функціональними можливостями систем управління базами даних – від простих однофайлових, орієнтованих на оброблення інформації відносно невеликого обсягу, до функціонально розвинених, призначених для розв’язання складних задач. Добре відомі такі системи управління базами даних, як MySQL, Oracle, SyBase, Informix, FoxPro, Paradox, Clipper, Access, Clarion та ін. [1].

Часто адміністрування баз даних за допомогою систем управління базами даних викликає труднощі, які пов’язані зі складністю процесів. Це породжує проблему в пошуку спеціального інструмента, призначеного для зручного та ефективного управління та роботи з різними базами даних. Такими інструментами є клієнти HeidiSQL, DBeaver, Navicat, SQL Server Management Studio та ін..

Існує значна кількість досліджень та наукових статей, присвячених розгляду та аналізу клієнтів для управління базами даних. Наприклад, Р. Батра у своїй книзі «SQL Практикум: Прискорений вступ до основ SQL» [2] присвячує цілий розділ огляду найкращих, на його думку, інструментів управління реляційними базами даних, при цьому надаючи перевагу безкоштовному програмному забезпеченню з відкритим кодом. А от А. Карнейро та інші співавтори описують в науковій роботі

«DBSitter: Розумний інструмент для адміністрування баз даних» [3] розробку новаторського підходу до адміністрування баз даних за допомогою прототипу DBSitter, який автоматизує моніторинг та виправлення помилок в реляційних системах управління базами даних, використовуючи кейс-підхід та взаємодію з адміністратором.

Як ми бачимо з наведених прикладів, науковці здебільшого порівнюють функціонал конкретних клієнтів або займаються розробкою нових технологій у цій області. Метою нашої статті є ознайомлення користувачів з можливостями клієнта HeidiSQL в аспекті адміністрування користувачів баз даних.

HeidiSQL – це вільний відкритий клієнт для управління базами даних, розроблений німецьким програмістом А. Бекером (A. Becker) та кількома іншими розробниками, написаний на Delphi. Він підтримує з'єднання та роботу з MySQL, MariaDB та Percona, а також Microsoft SQL Server, починаючи з версії 7.0 [4]. Варто зауважити, що HeidiSQL працює лише на платформі Windows.

Щоб управляти базою даних з HeidiSQL, користувач має увійти на локальний або віддалений сервер MySQL з прийнятним паролем, створивши сесію. В рамках цієї сесії користувач може управляти базами даних на сервері MySQL, і від'єднатися після закінчення роботи [4].

У HeidiSQL реалізовано багато можливостей, в тому числі і управління користувачами: додавання, видалення та редагування користувачів та їхніх паролів, а також управління привілеями користувачів. Оскільки бази даних можуть містити конфіденційну інформацію, то постає питання про обмеження доступу користувачів до баз даних в HeidiSQL.

Щоб бази даних, для яких ви не є власником, були доступні абсолютно кожному користувачеві на сервері, існує система користувачів цих баз даних. Доступ до будь-якої бази даних може бути призначений адміністратором (або уповноваженим користувачем) іншому користувачеві, причому він може бути

повним або обмеженим. Більш конкретно цей ступінь доступу виражається в привілеях.

Привілеї, надані обліковому запису MySQL, визначають, які операції обліковий запис може виконувати. Привілеї MySQL відрізняються в контексті, в якому вони застосовуються, і на різних рівнях роботи:

– Адміністративні привілеї дозволяють користувачам керувати роботою сервера MySQL;

– Привілеї баз даних застосовуються до баз даних і до всіх об’єктів у ній;

– Привілеї для об’єктів баз даних, таких як таблиці, індекси, подання та збережені підпрограми.

В офіційній документації MySQL [5] рекомендують проявляти особливу обережність при наданні привілею «FILE» та адміністративних привілеїв:

Привілей «FILE» надає доступ на читання будь-якого файлу на сервері, до якого є доступ у самої системи MySQL і доступ на створення файлу в директоріях, на які MySQL має права запису. Отримати доступ до таблиці можна за допомогою привілею «SELECT».

«GRANT OPTION» дозволяє як призначити конкретні права певному користувачеві, так і відібрати. А два користувачі, які мають різні привілеї та мають «GRANT OPTION» привілей, можуть об’єднувати привілеї.

«ALTER» дозволяє змінювати структуру таблиць.

«SHUTDOWN» дозволяє вимикати MySQL-сервер.

«PROCESS» можна використовувати для перегляду інформації про потоки (процеси), які виконуються на сервері.

«SUPER» дозволяє завершити процеси, що належать іншим користувачам; змінити глобальні системні змінні; проводити оновлення для системних змінних навіть за наявності лише прав на читання та ін.

Привілеї, надані системній базі даних MySQL, можна використовувати для зміни паролів та іншої інформації про привілеї доступу. Паролі зберігаються в зашифрованому вигляді, тому зловмисник не зможе просто прочитати їх, щоб дізнатися простий текстовий пароль. Однак користувач із доступом на запис до стовпця «mysql.user» системної таблиці «authentication_string» може змінити пароль облікового запису, а потім підключитися до сервера MySQL за допомогою цього облікового запису.

«INSERT» або «UPDATE» надані для MySQL системної бази даних дозволяють користувачеві додавати привілеї або змінювати існуючі привілеї відповідно.

«DROP» для MySQL системної бази даних дозволяє користувачеві віддалено використовувати таблиці привілеїв або навіть саму базу даних.

На Рис. 1 зображено менеджер користувачів в HeidiSQL – інструмент, який дозволяє адміністраторам баз даних керувати доступом користувачів до бази даних. Він надає можливість створювати, редагувати та видаляти користувачів, надавати їм права доступу та встановлювати інші параметри безпеки.

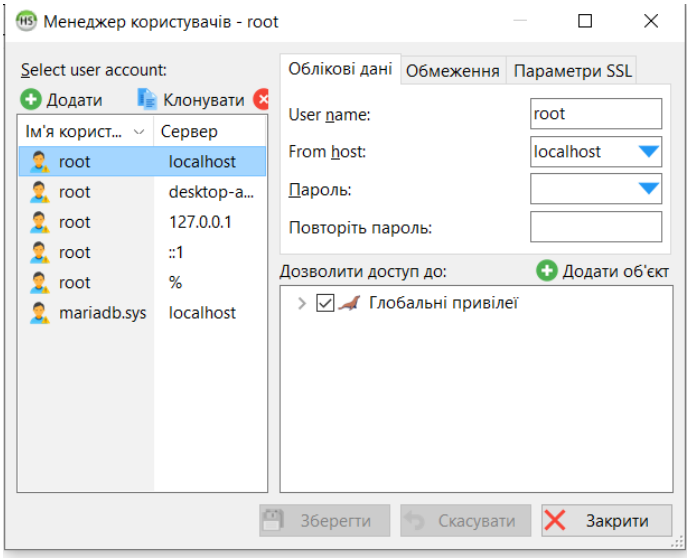

Рис. 1. Менеджер користувачів в HeidiSQL

Безперечно, обмеження доступу до баз даних є важливим аспектом забезпечення безпеки та ефективності роботи. Така функція гарантує, що лише авторизовані користувачі мають право звертатися до баз даних. А це, в свою чергу, запобігає несанкціонованому доступу, який може призвести до витоку конфіденційної інформації або навіть видалення даних. Якщо кожен користувач має доступ лише до тих даних, які йому потрібні для виконання конкретних завдань, це допомагає уникнути конфліктів при доступі та зменшує навантаження на систему. Не менш важливим є те, що обмеження доступу допомагає підприємствам дотримуватися вимог щодо захисту конфіденційної інформації, наприклад, таких як особисті дані чи медична інформація, і уникати можливих штрафів або інших правових наслідків.

Отже, з усього вищезазначеного можна зробити висновок, що клієнт HeidiSQL є досить зручним інструментом для користувачів, які працюють з базами даних і мають потребу в захисті конфіденційної інформації. Розглянутий інструмент володіє значним набором можливостей у сфері адміністрування користувачів, що робить його особливо цікавим для досліджень. Подальші перспективи в цьому напрямку передбачають систематичне дослідження нових технологій пов’язаних з розвитком засобів безпеки та оптимізацією функціоналу клієнта HeidiSQL.

Список джерел

1. Фетісов В. С. Робота із СУБД Access: [навчально-методичний посібник] / Фетісов В. С. 2-ге вид., перероб. і доп. – Ніжин: НДУ ім. М. Гоголя, 2011. – с. 103
2. Batra R. SQL Primer: An Accelerated Introduction to SQL / Batra R. Berkeley, CA: Apress, 2018. – с. 194
3. A. Carneiro, et al., DBSitter: An Intelligent Tool for Database Administration: materials of the 15th International Conference on Database and Expert Systems Applications (DEXA 2004) / Galindo, F., Takizawa, M., Traunmüller, Berlin, Heidelberg: Springer, 2004. – с. 171-180
4. https://uk.wikipedia.org/wiki/HeidiSQL

5. https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.h