

Державна служба України з надзвичайних ситуацій  
Львівський державний університет безпеки життєдіяльності  
Навчально-науковий інститут цивільного захисту  
Кафедра інформаційних технологій та систем електронних комунікацій

«Допущено до захисту»  
Начальник кафедри ІТ та СЕК  
підполковник служби цивільного  
захисту  
\_\_\_\_\_ Олександр ПРИДАТКО  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ року

## КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Система моніторингу та обслуговування MS SQL-сервера»

Виконала:  
здобувач VI курсу, групи КН-61мз  
спеціальності (освітньої програми)  
122 «Комп'ютерні науки»  
(Комп'ютерні науки)

\_\_\_\_\_  
(шифр і назва спеціальності (освітньої програми))

\_\_\_\_\_ Софія ГЛУШКО \_\_\_\_\_

Керівник \_\_\_\_\_ Ігор МАЛЕЦЬ \_\_\_\_\_

Рецензент

Державна служба України з надзвичайних ситуацій  
Львівський державний університет безпеки життєдіяльності  
Навчально-науковий інститут цивільного захисту

Кафедра інформаційних технологій та систем електронних комунікацій  
Освітній ступінь магістр  
Спеціальність 122 «Комп'ютерні науки»  
Освітня програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Начальник кафедри ІТ та СЕК  
підполковник служби цивільного  
захисту

\_\_\_\_\_ Олександр ПРИДАТКО  
“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ року

### ЗАВДАННЯ

на кваліфікаційну роботу

Здобувач \_\_\_\_\_ Софія Глушко \_\_\_\_\_  
(ім'я, прізвище)

1. Тема: Система моніторингу та обслуговування MS SQL-сервера

Керівник роботи \_\_\_\_\_ Ігор Малець, к.т.н., доцент \_\_\_\_\_  
(ім'я, прізвище, науковий ступінь, вчене звання)

затверджені наказом ЛДУ БЖД від “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ року № \_\_\_\_\_

2. Термін подання здобувачем роботи \_\_\_\_\_

3. Початкові дані до роботи:

1. SQL Server Architecture Explained: Named Pipes, Optimizer, Buffer Manager [З мережі] <https://www.guru99.com/sql-server-architecture.html#10>

2. SQL Server. Storage Engine. Страницы данных [З мережі] <http://msbi-dwh.blogspot.com/2015/02/sql-server-storage-engine.html>

3. What's new in .NET Core 3.0 (Preview 5) [З мережі] 2019  
<https://docs.microsoft.com/en-us/dotnet/core/whats-new/dotnet-core-3-0>

4. Методичні вказівки до виконання кваліфікаційної роботи магістра для здобувачів другого рівня вищої освіти спеціальності 122 «Комп'ютерні науки».

Укл. Ольга Смотри, Олександр Придатко, Назарій Бурак. – Львів: Вид-во ЛДУ БЖД, 2023. – 36 с.

4. Зміст кваліфікаційної роботи/проекту (перелік питань, які потрібно розробити)

Вступ

Розділ 1. Огляд MS SQL Server і існуючих аналогів

Розділ 2. Проектування системи

Розділ 3. Розробка системи

Розділ 4. Інструкція з експлуатації

Висновки

Список використаних джерел

Додатки

5. Консультанти розділів роботи

Розділ	Ім'я, Прізвище та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

6. Дата видачі завдання \_\_\_\_\_

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Вивчення джерел за тематикою роботи		
2.	Розроблення технічного завдання		
3.	Моделювання структури програмного забезпечення		
4.	Розробка програмного забезпечення		
5.	Оформлення документації кваліфікаційної роботи		

Здобувач

\_\_\_\_\_  
(підпис)

Софія ГЛУШКО

(ім'я та прізвище)

Керівник роботи

\_\_\_\_\_  
(підпис)

Ігор МАЛЕЦЬ

(ім'я та прізвище)

## Анотація

Софія Глушко «Система моніторингу та обслуговування MS SQL-сервера». Кваліфікаційна робота за спеціальністю 122 «Комп'ютерні науки» складається з текстової частини, що містить 4 розділи, 62 с., 19 рис., 11 табл., 10 джерел.

В магістерській кваліфікаційній роботі реалізовано систему моніторингу і обслуговування серверу баз даних MS SQL.

Програма дозволяє створювати користувацькі сценарії обслуговування серверу баз даних, призначати виконання процедур обслуговування на конкретний час, збирати статистику про стан серверу баз даних. Програмний продукт був створений з застосуванням мови програмування C# на платформі .Net Core. Для побудови базової архітектури було використано ASP.Net Core.

Для створення користувацького інтерфейсу була використана технологія Angular 7.

Метою даного проекту є розробка програмного забезпечення для обслуговування і моніторингу серверу баз даних MS SQL Server.

Джерелом розробки є науково-технічна література, публікації в виданнях, довідники, публікації в Інтернеті по опису архітектури і принципу роботи MS SQL Server та досвіду їх обслуговування, розробка веб додатків.

БАЗА ДАНИХ, РЕЛЯЦІЙНА МОДЕЛЬ ДАНИХ, СЕРВЕР, T-SQL СКРИПТИ, СТРУКТУРА СИСТЕМИ.

## **Annotation**

Sofiia Hlushko "The system for monitoring and maintaining MS SQL server". Graduation work on the specialty 122 "Computer Science" consists of a text part containing 4 sections, 69 pages, 19 figures, 11 table, 10 sources.

In this work for a Master Degree, the system for monitoring and maintaining the server MS SQL database was implemented.

The program allows you to create custom database server maintenance scripts. Assign the execution of maintenance procedures for a specific time. Collect statistics about the status of the database server. The software product was created using the C # programming language on the .Net Core platform. ASP.Net Core was used to build the basic architecture.

Angular 7 was used to create a user interface.

The purpose of this project is the development of software for maintenance and monitoring of the MS SQL Server database server.

The source of the development is scientific and technical literature, publications in publications, reference books, publications on the Internet describing the architecture and principle of operation of MS SQL Server and the experience of their maintenance, development of web applications.

**DATABASE, RELATIONAL DATA MODEL, SERVER, T-SQL SKRIPTS, SYSTEM STRUCTURE.**

## ЗМІСТ

Вступ.....	8
<b>Розділ 1. Огляд MS SQL Server і існуючих аналогів.....</b>	<b>10</b>
1.1.Визначення бази даних.....	10
1.2.Microsoft SQL Server.....	10
1.2.1. Рівень протоколу (Protocol Layer).....	11
1.2.2. Обробник запитів (Query Processor).....	12
1.2.3. Механізм сховища (Storage Engine).....	12
1.3.Зберігання даних в MS SQL Server Зберігання даних в MS SQL Server.....	13
1.4.Статистика.....	15
1.5.Необхідність в обслуговуванні.....	15
1.5.1. Фрагментація.....	15
1.5.2. Застаріла статистика.....	16
1.6.Існуючі аналоги.....	17
<b>Розділ 2. Проектування системи.....</b>	<b>19</b>
2.1.Функціонал.....	19
2.1.1. Керування T-SQL скриптами.....	20
2.1.2. Керування параметрами для підключення до серверу бази даних.....	21
2.1.3. Виконання T-SQL скриптів.....	23
2.1.4. Планування виконання скриптів.....	24
2.1.5. Моніторинг.....	24
2.2.Модель системи.....	25
<b>Розділ 3. Розробка системи.....</b>	<b>28</b>
3.1.Сервер інтерфейсу користувача.....	28
3.1.1. Сервіси.....	28

3.1.2. Компоненти .....	30
3.2.Сервер системи обслуговування .....	31
3.2.1. Структура застосунку .....	31
3.2.2. Виконання скриптів .....	35
3.2.3. Планування задач виконання скриптів .....	36
3.2.4. Доступ до даних .....	37
3.2.5. API системи .....	38
<b>Розділ 4. Інструкція з експлуатації.....</b>	<b>39</b>
4.1.Встановлення системи .....	39
4.1.1. Розгортання серверу бази даних.....	39
4.1.2. Розгортання серверу обслуговування .....	39
4.1.3. Розгортання серверу інтерфейсу користувача .....	40
4.2.Інструкція користувача.....	41
4.3.Недоліки та удосконалення.....	45
Висновки.....	46
Список використаної літератури.....	47
Додатки.....	48

## ВСТУП

У сучасному світі велику роль для більшості підприємств відіграють складні системи для автоматизації виконання певних бізнес задач, обслуговування клієнтів, аналізів ринку, маркетингу і так далі. В загальному, всі ці системи призначені для обробки даних, об'єми яких часто бувають дуже великими. Тому задачі збереження і обробки даних потрібно приділяти дуже багато уваги. Для цього існує багато систем керування базами даних. Одними з найпоширеніших є реляційні системи керування базами даних (РСКБД).

РСКБД мають складну структуру. Вони, в першу чергу, повинні реалізовувати реляційну модель, яка вимагає широкого функціоналу, такого як: структурування всіх даних в таблиці, підтримання встановлених обмежень колонок, первісних і зовнішніх ключів, реалізація транзакцій і властивостей ACID. Оскільки дані повинні зберігатись на фізичному носії, який має свої обмеження по швидкодії і об'єму, потрібно оптимально структурувати дані і реалізувати доступ до них. Найпоширенішими структурами являються набори і індекси. Набір являє собою невпорядкований набір даних, а індекс – збалансоване дерево. Індекс дозволяє здійснювати швидкий пошук даних за рахунок зменшення операцій читання. Але також індекс має і негативні сторони, такі як збільшення загального об'єму бази даних і необхідність підтримувати сам індекс. Це зобов'язує розробника відповідально відноситись до створення індексів і певним чином обслуговувати вже існуючі. Також база даних може мати в собі певні службові механізми, наприклад статистика, яка також зберігається в базі даних і потребує оновлення.

Загалом, сам сервер може підтримувати працездатність своїх компонентів, наприклад, шляхом оновлення статистики, підтримки логічної структури індексів, оптимізації використання ресурсів і так далі. Але для збереження швидкодії і безвідмовної роботи безумовно необхідне обслуговування серверу.



І це є доволі складною і об'ємною роботою, для якої необхідні відповідні знання і навички. Тому, навіть професіоналу потрібні певні інструменти.

## ВИСНОВКИ

Темою кваліфікаційної роботи є розробка системи моніторингу і обслуговування серверу баз даних MS SQL Server. Дана система призначена для зменшення затрат і підвищення якості обслуговування серверів баз даних.

Мною була розглянута архітектура і основні механізми роботи MS SQL Server. Були визначені основні об'єкти серверу, які потребують моніторингу і обслуговування. Основними такими об'єктами виявились індекси і статистика. Також даний список може розширюватись в залежності від потреб користувача. Було розглянуто подібні системи і визначено їхні основні переваги та недоліки.

Я визначила необхідний функціонал та вимоги до системи. Описала основні прецеденти і на основі них виокремила модель даних системи та її складові.

Визначила основні технології для реалізації. Для серверу бази даних було обрано MongoDB. Для застосунку інтерфейсу користувача – Angular 7. Платформою для застосунку системи слугував .Net Core 3.0 (Preview 3). Бібліотека StackExchange Dapper ORM слугує для виконання скриптів, планувальник Quartz.Net для виконання в заданий час скриптів.

Даний застосунок може бути використаний на підприємствах з декількома навантаженими базами даних для їх обслуговування та моніторингу. Також він може бути корисний для невеликих компаній, в яких немає адміністраторів баз даних. За допомогою цієї системи обслуговування може проводитися розробником або системним адміністратором.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Реляційна модель даних [З мережі] 2017.  
[https://ru.bmstu.wiki/Реляційна\\_модель\\_даних](https://ru.bmstu.wiki/Реляційна_модель_даних)
2. SQL Server Architecture Explained: Named Pipes, Optimizer, Buffer Manager [З мережі] <https://www.guru99.com/sql-server-architecture.html#10>
3. Virendra Yaduvanshi, SQL Server Network Interface/Protocol [З мережі] 2012 <https://wikidba.net/2012/08/31/sql-server-network-interfaceprotocol/>
4. Zameer Ansari, How does the SQL Server Query Optimizer Works [З мережі] 2017 <https://hackernoon.com/how-does-the-sql-server-query-optimizer-works-47541e2205aa>
5. SQL Server. Storage Engine. Страницы данных [З мережі] <http://msbi-dwh.blogspot.com/2015/02/sql-server-storage-engine.html>
6. Robert Sheldon, SQL Server Statistics Basics [З мережі] 2016 <https://www.red-gate.com/simple-talk/sql/performance/sql-server-statistics-basics/>
7. Chad Boyd, SQL Server Fragmentation What causes it (Part 3 of 9) [З мережі] 2008 <https://www.mssqltips.com/sqlservertip/2263/sql-server-fragmentation-what-causes-it-part-3-of-9/>
8. Alexandre Hamel, SQL Server Statistics: Maintenance And Best Practices [З мережі] 2014 <https://blog.pythian.com/sql-server-statistics-maintenance-and-best-practices/>
9. What's new in .NET Core 3.0 (Preview 5) [З мережі] 2019 <https://docs.microsoft.com/en-us/dotnet/core/whats-new/dotnet-core-3-0>
10. Common web application architectures [З мережі] 2019 <https://docs.microsoft.com/en-us/dotnet/standard/modern-web-apps-azure-architecture/common-web-application-architectur>

