Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр
(назва освітнього ступеня)

на тему: «Методи та засоби проведення аудиту системи управління базами даних»

Виконав(ла): студент(ка) VI курсу, групи СБм-61
спеціальності 125 «Кібербезпека»
(шифр і назва спеціальності)

(підпис) Дрозд В. М. (прізвище та ініціали)

Керівник (підпис) Карпінський М. П. (прізвище та ініціали)

Нормоконтроль (підпис) Кареліна О.В. (прізвище та ініціали)

Завідувач кафедри (підпис) Загородна Н.В. (прізвище та ініціали)

Рецензент (підпис) (прізвище та ініціали)

Тернопіль
2021

АНОТАЦІЯ

Методи та засоби проведення аудиту системи управління базами даних // Дипломна робота ОР «Магістр» // Дрозд Віталій Михайлович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2021 // С. , рис. – 4, табл. – 0, кресл. – 0, додат. – 3 .

Ключові слова: СИСТЕМА УПРАВЛІННЯ БАЗ ДАНИХ, АУДИТ СУБД, АУДИТ, БАЗИ ДАНИХ.

Дана магістерська кваліфікаційна робота присвячена дослідженню методів та засобів проведення аудиту систем управління баз даних. Проведено дослідження методів та засобів проведення аудиту системи управління баз даних.

Для проведення аудиту системи управління бази даних використано метод із використанням тригерів.

У першій главі проведено аналіз і класифікація підходів до аудиту баз даних.

У другій главі описані теоретичні відомості щодо проведення аудиту систем управління баз даних. Описано вбудовані засоби відстеження змін й оглянуто існуючі програми для аудиту.

У підрозділі «Охорона праці» розглянуто Закон про охорону праці й вимоги безпеки до робочих місць працівників з екранними пристроями. У підрозділі «Фактори що впливають на функціональний стан користувачів комп'ютерів» розглянуто різноманітні фактори які можуть впливати на функціональний стан користувачів комп'ютерів.

ANNOTATION

Methods and means of conducting audit of database management system // Thesis of OR "Master" // Drozd Vitalii Mikhailovich // Ternopil National Technical University named after Ivan Pulyuy, Faculty of Computer Information Systems and software engineering, Department of Cybersecurity, SBm-61 group // Ternopil, 2021 // P. , fig. - 4, table. - 0, chair. - 0, added. - 3.

Key words: DATABASE MANAGEMENT SYSTEM, AUDIT DBMS, AUDIT, DATABASES.

This master's thesis is devoted to the study of methods and tools for auditing database management systems. A study of methods and means of auditing the database management system.

The trigger method was used to audit the database management system.

The first chapter analyzes and classifies approaches to database auditing.

The second chapter describes the theoretical information on the audit of database management systems. Built-in change tracking tools are described and existing audit programs are reviewed.

In the subdivision "Labor protection" the Law on labor protection and safety requirements for workplaces of workers with screen devices is considered. The section "Factors affecting the functional status of computer users" discusses various factors that may affect the functional status of computer users.The first chapter presents existing approaches to text analysis.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

DML	Data Manipulation Language
DDL	Data Definition Language
СУБД	Система Управління Баз Даних
XML	Extensible Markup Language
TDE	Transparent Data Encryption

ВСТУП

Сьогодні існує дуже багато інформаційних систем, які містять бази даних (БД). Для розробників цих баз даних складним завданням є відстеження можливих змін у базі. Ще більш непростим завданням є створення такого рішення для відстеження цих змін, яке не сильно впливало б на продуктивність робочих навантажень і було б простим у проектуванні, реалізації та управлінні.

Аудит у контексті системи управління базою даних (СУБД) загалом може полягати у збереженні інформації про все, що відбувається у базі даних. Це можуть бути спроби авторизації користувачів, запити до даних, що призводять або не призводять до їхньої зміни, та багато іншого. Загалом аудит, журналізація чи контроль змін структури та даних БД зазвичай вирішують такі завдання:

- отримання інформації про те, хто, коли та звідки робив зміни структури БД та/або даних;
- відстеження історії зміни структури БД та/або даних;
- повідомлення про зміни структури БД та/або даних.

Система аудиту відіграє велику роль у відслідковуванні змін у разі відмови системи та необхідна для відновлення бази даних цієї системи із резервної копії. У разі відмови або сигналу відкату однієї з транзакцій журнал змін, що генерується системою аудиту, сканується, і всі записи транзакції, що скасовується, витягуються з журналу.

Більшість СУБД мають низку вбудованих функцій для відстеження змін даних та структури бази даних. Тим не менш, кожна база даних унікальна, у кожній є необхідність відстеження певних змін і до кожної з них потрібен свій підхід до аудиту, тому в рамках даної роботи існує потреба в аналізі підходів до аудиту та вибору найбільш оптимального з них, а також у визначенні найбільш відповідної структури журналу аудиту для подальшого використання в розробці підсистеми аудиту БД конкретної інформаційної системи.

Метою даної роботи є вивчення методів та засобів проведення аудиту системи управління базами даних.

Для реалізації мети, викладеної вище, необхідно вирішити дві ключові задачі: теоретичну та практичну. Теоретична задача передбачає аналіз методів організації і проведення аудиту баз даних та журналів аудиту. На основі цього аналізу будуть обрані методи для вирішення практичного завдання. Для виконання практичної частини роботи необхідно також обґрунтувати вибір СУБД та середовища розробки програми (інтерфейсу).

1 АНАЛІТИЧНА ЧАСТИНА

1.1 Класифікація підходів до аудиту баз даних

У корпоративних інформаційних системах, особливо у яких регулярно відбувається оновлення даних, необхідно частково чи повністю вирішувати завдання, пов'язані з відстеженням змін даних у базі даних та/або структури бази даних.

В даний час існує кілька підходів до аудиту баз даних, але будь-якої єдиної класифікації цих підходів немає. У роботі буде розглянуто кілька найчастіше використовуваних підходів до аудиту баз даних, які можна розділити за такими ознаками.

За компонентами:

- на рівні сервера;
- на рівні бази даних.

За способом знаходження змін:

- за допомогою тригерів;
- за журналом транзакцій.

1.2 Аудит на рівні сервера

В основі більшості сучасних інформаційних систем лежить триланкова архітектура, що ґрунтується на схемі «Клієнт – Сервер додатка – Сервер даних», причому, як правило, завдання моніторингу операцій, які виконує користувач, вирішується на рівні сервера додатків. У цьому випадку під аудитом маються на увазі такі дії, як відстеження різних спроб входу в систему, реєстрація відновлення або створення резервної копії бази даних, реєстрація часу створення або видалення бази даних тощо.

Перевага **аудиту на рівні сервера** полягає в тому, що він відіграє певну роль у забезпеченні безпеки системи, оскільки, використовуючи цей підхід,

можна запобігти діям зловмисника, відстежуючи невдалі чи несанкціоновані спроби входу до системи [1]. Але, попри важливу роль моніторингу даних лише на рівні сервера, є необхідність контролю змін даних лише на рівні операцій, вироблених у самій базі даних у вигляді різних SQL-команд.

1.3 Аудит на рівні бази даних

Зазвичай кожна інформаційна система має адміністратора, який супроводжує її за допомогою SQL-інтерфейсу, і його дії не фіксуються в журналі операцій, що проводяться на рівні сервера. Таким чином, складно відстежити можливі несанкціоновані дії з боку адміністратора. Також сама система не може бути повністю захищена від вразливостей, що дозволяють її зламати, що може призвести до небажаних змін у базі даних, які не відображатимуться в журналі подій [2].

Крім різноманітних несанкціонованих дій, які можуть бути застосовані до бази даних, дуже часто виникає необхідність зробити прямі зміни в базі даних або її структурі за допомогою SQL-команд, наприклад, таких як додавання, зміна або видалення таблиць і даних у них. Очевидним є те, що такі операції необхідно контролювати, тому що є ймовірність того, що вони виявляться некоректними, наприклад, буде видалена не та таблиця або неправильно змінені якісь дані.

Всі перелічені випадки свідчать, що аудиту подій лише на рівні сервера недостатньо, і також необхідний аудит операцій лише на рівні бази даних. Він може стати корисним інструментом для підвищення рівня безпеки системи та зменшення помилок при її експлуатації.

1.4 Аудит за допомогою тригерів

При виборі механізму, що забезпечує відстеження подій на рівні даних, варто враховувати, що одним із головних критеріїв вибору є мінімальний вплив цього механізму на продуктивність системи.

Найбільш відомим і часто використовуваним методом відстеження змін, пов'язаних з базами даних є аудит за допомогою тригерів. Тригер являє собою особливий тип процедури, що зберігається, яка виконується автоматично при виникненні дій з модифікації даних у базі даних, тобто. тригер автоматично запускається сервером під час спроби змінити дані.

За допомогою тригерів можна реалізувати відстеження та протоколювання операцій, пов'язаних як із зміною даних у базі, так і зі зміною структури бази даних. Події, пов'язані зі зміною даних у таблицях бази даних або уявленнях, описуються такими операціями DML, як INSERT, DELETE або UPDATE. Тригери, що спрацьовують ці події мови обробки даних DML, називаються тригерами DML. Тригери, що активуються у відповідь на події, пов'язані зі змінами в структурі бази даних, наприклад, CREATE, DROP, ALTER і т.д., називаються тригерами DDL.

Окрім тригерів, що відстежують зміну структури бази даних та даних у ній, існують тригери входу, які спрацьовують у відповідь на подію LOGON. Ця подія виникає при встановленні сеансів користувача. Тригери входу можна використовувати не тільки для автентифікації під час входу та керування сеансами сервера, але й для відстеження різних спроб входу в систему.

Метод моніторингу за допомогою тригерів також використовується в деяких вбудованих засобах аудиту СУБД, в яких система створює набір тригерів за певними параметрами в автоматичному режимі. Прикладами таких рішень для відстеження змін структури бази даних або даних у ній є вбудований в СУБД Oracle засіб AUDIT або Change Tracking, вбудований в SQL Server.

Основним недоліком застосування тригерів та, відповідно, вбудованих засобів аудиту для відстеження змін є значне зниження продуктивності системи. Це відбувається, оскільки в даному випадку одночасно з кожною операцією транзакції здійснюється додавання запису в таблицю аудиту. При використанні аудиту за допомогою тригерів або вбудованих засобів аудиту доцільно заздалегідь встановити умови аудиту, за яких моніторинг зачіпатиме не всі можливі зміни в базі даних та всіх її таблицях, а лише найважливіші з них.

Інакше, якщо до аудиту буде підключено більшу частину операцій над базою даних, буде вплинути на продуктивність системи у бік її зменшення.

1.5 Аудит за журналом транзакцій

У зв'язку з деякими недоліками аудиту за допомогою тригерів поширення також отримав підхід до моніторингу змін у базі даних, заснований на роботі з журналом транзакцій, з якого можна отримати інформацію про операції DDL і DML. Для деяких СУБД існують системи аудиту журналу транзакцій, створені у вигляді окремих продуктів, наприклад, Change Data Capture або Oracle Audit Vault. Ці системи мають можливість відстежувати зміни, зафіксовані в журналі транзакцій, застосовувати різні фільтри для вибірки записів, що цікавлять користувача, формувати зручні звіти з цими змінами в різних форматах і т.д.

Порівняно з аудитом за допомогою тригерів та вбудованими засобами аудиту моніторинг за допомогою журналу транзакцій не так значно негативно впливає на продуктивність системи, хоча теж є досить складним процесом, пов'язаним із трансформацією даних та парсингом.

Однією з важливих переваг аудиту з журналу транзакцій є те, що в цьому журналі зберігається інформація про всі версії вже давно змінених даних і проведені операції. Це означає, що користувач може запросити історію будь-яких модифікацій за період з журналу транзакцій. Очевидно, що це корисно і затребувано, але підтримка цієї функції на сервері системи може позначитися на її продуктивності, тому що виконання таких запитів передбачає сканування великої кількості архівних файлів журналу транзакцій.

2 ТЕОРЕТИЧНА ЧАСТИНА

2.1 Системи управління базами даних

Під системою СУБД розуміють програмне забезпечення, за допомогою якого можна керувати створенням та використанням баз даних. Як правило, СУБД володіє такими основними функціями:

- визначення даних (створення бази даних, вказівки типів та структури даних тощо), яке зазвичай реалізується за допомогою мови визначення даних DDL;
- обробка даних (вставка, видалення, зміна даних у таблицях бази даних, організація запитів тощо), що зазвичай реалізується за допомогою мови маніпулювання даними DML;
- підтримка безпеки та цілісності даних (контроль запитів користувача, забезпечення несуперечливого стану даних, що зберігаються в базі даних);
- резервне копіювання та відновлення даних у разі різних збоїв.

На даний момент існує кілька класифікацій СУБД: за моделлю даних (реляційні, ієрархічні, мережеві, об'єктно орієнтовані, об'єктно реляційні), за ступенем розподіленості (локальні та розподілені) та за способом доступу до бази даних (файл серверні та клієнт серверні).

Що ж до класифікації за першою ознакою, очевидно, більшість розробників орієнтуються на реляційні СУБД.

Між локальними та розподіленими СУБД перевага найчастіше надається розподіленим. Особливістю локальних СУБД і те, що її частини перебувають у одному комп'ютері, тобто. Щоб декільком користувачам одночасно працювати з однією базою даних, на кожному комп'ютері повинна бути копія цієї локальної бази даних. Істотним недоліком таких СУБД є проблема синхронізації копій даних, тому для спільної роботи кількох користувачів різних комп'ютерах локальні СУБД не використовуються на відміну від розподілених СУБД, частини яких можуть розміщуватися на декількох комп'ютерах.

За методом доступу до бази даних СУБД поділяються на файл серверні та серверні. У СУБД кожного з цих двох типів є свої переваги та недоліки і, відповідно, своя цільова аудиторія.

У файл серверних СУБД дані зазвичай розміщуються у каталогах одного комп'ютера, який постійно підключений до мережі. СУБД розташовується кожної робочої станції, і доступ СУБД до даних реалізується через локальну мережу. Перевагою таких СУБД є легкість їх встановлення, простота використання, відсутність потреби у додатковому програмному забезпеченні та невелике навантаження процесор файлового сервера. Але у зв'язку з тим, що файл-серверні СУБД мають спрощену архітектуру, вони не мають можливості підтримувати всі функції СУБД, наприклад, у них немає функції автоматичного відновлення даних після можливих збоїв, не ведеться журнал транзакцій і т. п. До недоліків таких СУБД можна також віднести високу завантаженість локальної мережі та складність забезпечення високої надійності та безпеки даних. Отже, файл-серверні СУБД використовуються для вирішення щодо нескладних завдань, наприклад, що передбачають невелику кількість користувачів або невелику кількість даних, що оброблюються. Проте такі СУБД мають досить широку сферу застосування та використовуються для ведення персональних баз даних та в невеликих організаціях, розташованих найчастіше в одному будинку, з невеликою кількістю користувачів та невисокою частотою оновлення даних. У таких умовах використання файл-серверних СУБД є цілком виправданим. На сьогоднішній день існує досить багато файл-серверних СУБД, але найбільш поширеними є Microsoft Access, Borland dBase, Corel Paradox, Microsoft FoxPro та ін.

За описаними вище особливостями і недоліками файл серверних СУБД великих організацій їх використання є неприйнятним. І тут доцільно використовувати клієнт серверні СУБД. Вони дозволяють клієнту і серверу обмінюватися інформацією так: клієнт посилає запит на сервер бази даних, який розташовується на машині з даними, сервер бази даних своєю чергою приймає цей запит, шукає у даних необхідну інформацію та передає її клієнту. У цьому

більшість обчислювального навантаження лягає на сервер, отже, недоліком клієнт серверних СУБД є підвищені вимоги до серверу. Також до мінусів таких СУБД можна віднести складність їх встановлення та супроводу, але для великих підприємств, на яких і використовуються клієнт серверні СУБД, ці недоліки не мають особливої ролі. До того ж вони мають низку значних переваг. Одним з них є те, що такі СУБД набагато менш вимогливі до пропускної спроможності комп'ютерної мережі і мають більшу продуктивність, ніж файл серверні СУБД, так як сервер проводить пошук даних за параметрами запитів, які передає йому клієнт, а результат виконання цих запитів зазвичай набагато менше за обсягом ніж фрагменти файлів. Також до переваг клієнт серверних СУБД можна віднести зручність управління та можливість забезпечення таких важливих для великих організацій характеристик, як висока безпека та надійність. Прикладами клієнтів серверних СУБД є Oracle Database, MySQL, Interbase, Microsoft SQL Server і т.д.

В рамках даної роботи необхідно розробити підсистему аудиту бази даних для інформаційної системи великого підприємства з великою кількістю користувачів. База даних цієї інформаційної системи містить досить велику кількість даних, що часто оновлюються. Тож розробки підсистеми аудиту цієї бази даних буде доцільно використовувати реляційну розподілену СУБД, використовує клієнт серверні технології.

В даний час абсолютними лідерами на ринку СУБД, що відповідають заданим параметрам, є компанії Oracle, Microsoft та IBM. Їхня загальна частка на ринку становить близько 90%, а найчастіше використовуваними СУБД є Oracle Database, IBM DB2 та Microsoft SQL Server [9].

Корпорація Oracle від початку її заснування спеціалізувалася на створенні реляційних СУБД. На даний момент Oracle займає одну з лідируючих позицій на ринку СУБД і лідирує на платформах UNIX та Windows. Причиною популярності продуктів Oracle, у тому числі СУБД Oracle Database, є високі експлуатаційні характеристики СУБД. До них можна віднести підтримку великої кількості платформ, дуже високу надійність, високу швидкість обробки даних, наявність великого спектру засобів розробки, адміністрування та моніторингу,

орієнтацію на Інтернет технології та багато іншого. Однак мінусами СУБД Oracle є висока вартість самої СУБД та її супроводу, потреба у висококваліфікованому персоналі для підтримки бази даних, складність адміністрування та використання. Проте всі витрати на впровадження та освоєння цієї СУБД згодом окупаються надійною та ефективною роботою.

DB2 є сімейством систем управління реляційними базами даних, що випускаються корпорацією IBM. Дана СУБД також є одним із постійних лідерів на ринку СУБД, зокрема щодо продуктивності, можливостей масштабування, рівня технічної реалізації тощо. До переваг DB2 можна віднести мультиплатформенность, тобто. підтримку кількох операційних систем, простоту управління та використання, розширюваність та наявність досить гарної безкоштовної версії. Недоліками IBM DB2, як і Oracle, є досить висока вартість СУБД та її супроводу.

На відміну від двох, наведених вище СУБД, СУБД Microsoft SQL Server орієнтована лише на підтримку платформ Windows. Важливими позитивними характеристиками, якими володіє SQL Server, є відносна простота адміністрування та використання, швидкодія, високі функціональні можливості сервера СУБД, масштабованість (може бути застосована як для невеликих мереж, так і для мереж рівня підприємства), наявність засобів автоматичного налаштування параметрів конфігурації тощо .д. До основних недоліків SQL Server можна віднести неможливість роботи з усіма платформами, крім Windows, програмованість і можливу нестачу засобів роботи.

У результаті розгляду трьох провідних СУБД, їх основних характеристик, переваг і недоліків, можна зробити висновок про те, що вони мають дуже широкі можливості і не поступаються один одному в таких важливих характеристиках, як продуктивність, висока швидкість обробки даних і масштабованість. Однією з найбільш важливих відмінних рис Microsoft SQL Server від Oracle Database і IBM DB2 є те, що вона не є мультиплатформенною, проте вартість даної СУБД і її супроводу трохи нижче. Можливості ж для аудиту бази даних є у всіх трьох, представлених вище СУБД: у них є як підтримка процедур і тригерів, що

зберігаються, так і вбудованих засобів аудиту, крім того всі три продукти мають системи аудиту, що випускаються у вигляді окремих продуктів, що здійснюють моніторинг даних по журналу транзакцій.

Очевидно, що при виборі СУБД основним критерієм є оцінка того, наскільки вона задовольняє основним вимогам до інформаційної системи та програмного забезпечення. Підсистему аудиту, що розробляється, планується використовувати на платформі Windows. І зважаючи на те, що Microsoft SQL Server не поступається іншим СУБД, задовольняє вимогам інформаційної системи супроводу ремонту і вже використовується на підприємстві, підсистема аудиту бази даних для цієї інформаційної системи буде розроблена з використанням саме цієї СУБД.

2.2 Структура журналу аудиту

Способи відстеження змін даних слід розглядати окремо від аудиту змін структури бази даних. Вибір рішення для аудиту змін даних залежить від тривалості зберігання та від того, в якому обсязі необхідно зберігати інформацію. Також необхідно враховувати ступінь журналування: у деяких випадках достатньо відслідковувати лише факт зміни даних, в інших потрібне збереження декількох параметрів, таких як вид модифікації, старі та нові значення тощо.

У найпростішому випадку зберігаються останні зміни даних. При збереженні результатів аудиту даних використовуються додаткові таблиці, пов'язані зі змінною таблицею ставленням один до одного. Недоліком такого методу є неможливість фіксації операції видалення даних. Подібним чином фіксується факт зміни даних.

Вищевикладений метод ускладнюється шляхом поділу обробки трьох операцій INSERT, DELETE та UPDATE. При цьому отримана інформація зберігається у трьох додаткових таблицях кожної таблиці бази даних. У першу таблицю записується загальна інформація про зміни (ким були внесені зміни, з якого комп'ютера, чи вдала спроба зміни). Перша таблиця пов'язана з двома

іншими ставленням один до одного. У другу таблицю заноситься повний рядок даних при операціях вставки (INSERT) та видалення (DELETE). Третя таблиця призначена фіксувати зміненого значення під час операції UPDATE. В результаті виходить докладний варіант аудиту, без значних втрат продуктивності. Недоліком є безліч додаткових таблиць.

Іншим варіантом структури журналу аудиту є створення єдиної таблиці, у якій фіксуються зміни. Така таблиця може містити первинний ключ таблиці, що модифікується, її назва, назва зміненого поля і його нове значення. У разі відстеження зміни структури бази даних, таблиця містить тип події (створення, видалення або зміна об'єкта), тип об'єкта, назва об'єкта. Цей метод дозволяє реалізувати універсальні всім таблиць тригери, але неефективно витрачає ресурси системи, оскільки містить багато надлишкової інформації кожної записи. Крім того, виникає проблема запису значень різних типів у полі з фіксованим типом. Ця проблема може бути вирішена декількома способами:

1) Запис всіх змін у одній колонці. При такому підході загальна таблиця аудиту може мати стовпець, в якому всі значення модифікованих полів представляють один рядок, такий як рядок XML.

2) Запис на полі з універсальним типом (sql_variant). У такому підході кожен запис таблиці аудиту містить значення зміненого стовпця для конкретної таблиці. Також використовується явне або неявне приведення типів, однак знижується швидкість взаємодії з журналом аудиту.

Наступний метод передбачає копіювання основної таблиці та запис всіх змін у створену таблицю. Копія містить усі поля основної таблиці, а також службові поля, в яких може записуватися інформація про тип операції над даними, ім'я користувача, який зробив операцію, дату зміни. У журналі аудиту створюється запис, навіть якщо зміни зазнало всього одне поле у вихідній таблиці, що, як і в попередньому випадку, призводить до значного збільшення займаного простору на диску. Переваги полягають у простішій реалізації перегляду даних у таблиці аудиту та підвищеної швидкості обробки запитів.

2.3 Вбудовані засоби відстеження змін

При реалізації підсистеми аудиту можна скористатися стандартними засобами, вбудованими у СУБД. Починаючи з версії Microsoft SQL Server 2008 існує кілька таких рішень. Їх можна комбінувати та використовувати для однієї і тієї ж бази даних. Використання вбудованих засобів аудиту має ряд переваг у порівнянні з розробкою спеціальних тригерів та таблиць аудиту:

- зменшення часу на реалізацію системи аудиту;
- відсутня потреба у зміні структури бази даних;
- є вбудовані засоби очищення даних;
- зменшення впливу на продуктивність системи.

Журнал транзакцій є спеціальним файлом, який необхідний будь-якій базі даних для її належного функціонування. Він містить записи, створені в процесі ведення журналу, і він використовується для повторного читання цих записів під час відновлення. Так само, як простір, зайнятий власне записами журналу, транзакція в журналі транзакцій також резервує простір для будь-яких потенційних записів журналу, які були б потрібні в разі необхідності скасувати транзакцію і виконати відкат.

При створенні нової бази даних журнал транзакцій порожній. У міру виникнення транзакцій записи журналу послідовно записуються до журналу транзакцій, і з цього випливає, що створення кількох файлів журналу транзакцій не дасть жодного виграшу у продуктивності. Журнал транзакцій використовуватиме всі файли журналу по черзі.

Фізична архітектура журналу транзакцій містить частини, які називаються віртуальними файлами журналів. Це допоміжні засоби полегшення внутрішнього управління журналом транзакцій. Фізично записи журналу зберігаються в одному або декількох файлах LDF, які і утворюють журнал транзакцій.

Основною метою файлу LDF є забезпечення концепції ACID (атомарність, узгодженість, ізольованість, довговічність).

Атомарність (Atomicity) гарантує, що жодна транзакція не буде зафіксована в системі частково, а чи будуть виконані всі її підопераційї або не буде виконано жодної.

Узгодженість (Consistency) означає, що система перебуває у узгодженому стані перед початком транзакції і має залишатися у ньому після завершення транзакції.

Ізольованість (Isolation) передбачає, що під час виконання транзакції інші процеси нічого не винні бачити дані у проміжному стані. Паралельні транзакції наводять базу даних у стан, начебто транзакції відбувалися послідовно, одна одною.

Довговічність (Durability) означає, що, якщо користувач отримав підтвердження про виконання транзакції від системи, він може бути впевнений у тому, що зроблені ним зміни не будуть скасовані через будь-який збій.

Журнал транзакцій є кільцевим файлом, оскільки записи на початку журналу транзакцій очищаються. Коли процедура ведення журналу досягає кінця журналу транзакцій, вона знову повертається на початок і починає писати поверх того, що було раніше.

Microsoft SQL Server не виконує протоколювання у випадках, коли можуть виникнути проблеми з нестачею дискового простору при виявленні швидкого збільшення журналу транзакцій.

Для деяких операцій, наприклад, CREATE INDEX, Microsoft SQL Server не веде журнал для кожної нової сторінки. Натомість Microsoft SQL Server записує достатньо інформації, щоб визначити, як CREATE INDEX відпрацював, і прийняти рішення фіксації зміни або здійснення відкату.

2.3.1 Change Tracking

Функція Change Tracking не призначена для отримання всієї інформації про зміни, вона призначена, щоб визначити, чи змінено рядок чи ні з найменшими витратами ресурсів системи.

Change Tracking визначає ID зміненого рядка, але не надає інформацію про змінені значення. Тому під час операцій UPDATE або DELETE не можна дізнатися вихідне значення до оновлення або видалення запису. Зміни в стовпцях, що виконуються, виконання оператора SELECT і доступ до об'єкта бази даних також не відстежуються.

Change Tracking – це синхронний процес, який може використовуватися у програмах, призначених для односторонньої та двосторонньої синхронізації даних, оскільки може синхронізувати декілька баз даних. Прикладом застосування двосторонньої синхронізації є програма, яка використовує тимчасове з'єднання клієнта та сервера. У цьому випадку клієнтська програма запитує і надсилає дані в локальне сховище. Коли з'єднання між клієнтом і сервером доступне, програма синхронізується з сервером і спрямовує потоки даних.

2.3.2 Change Data Capture

Change Data Capture також збирає інформацію про зміни даних – вставку, видалення та оновлення, але надає більш детальну інформацію, ніж Change Tracking. Для операцій INSERT і DELETE Change Data Capture захоплює весь рядок, який було вставлено чи видалено. Для операції UPDATE ця функція захоплює два рядки – до і після оновлення, таким чином доступні як старі, так і нові значення.

Функція Change Data Capture не вимагає зміни структури існуючих таблиць, вся інформація зберігається у таблицях змін. Перші стовпці таблиці змін зберігають певну інформацію про транзакцію, у тому числі тип операції (видалення, вставка, оновлення) та ідентифікатор зміненого стовпця. Інші стовпці ідентичні вихідній таблиці і зберігають захоплену інформацію. Якщо

структура вихідної таблиці змінюється, структура таблиці змін відповідно оновлюється.

Як і Change Tracking, Change Data Capture є вбудовані засоби очищення, які видаляють застарілу інформацію після закінчення заданого часу. Інша подібність між цими двома функціями полягає в тому, що жодна з них не надає інформацію про те, хто і коли змінив дані.

2.3.2 SQL Server Audit

SQL Server Audit надає більш детальну інформацію про події, ніж Change Tracking та Change Data Capture. Ця функція надає інформацію про те, хто здійснив маніпуляції в базі даних, що було змінено і коли, а також забезпечує фільтрацію подій, що відстежуються. У той же час, SQL Server Audit не фіксує ім'я хоста або IP адресу комп'ютера, що викликав подію, а для деяких дій (наприклад, SELECT і UPDATE) не відстежуються відомості про дані, до яких було застосовано цю дію.

SQL Server Audit відстежує та протоколює події на двох рівнях – рівні сервера та рівні бази даних. Рівні аудиту налаштовуються незалежно один від одного, що забезпечує більшу гнучкість та ретельність аудиту. Результати аудиту можуть бути збережені у двійковому файлі, журналі подій Windows або журналі подій SQL Server.

Для відстеження подій необхідно створити об'єкт аудиту на цільовому примірнику SQL Server. Багато об'єктів аудиту можуть бути визначені на одному екземплярі SQL Server, кожен об'єкт зберігатиме інформацію у своєму власному файлі. Далі створюється специфікація аудиту сервера чи бази даних існуючого об'єкта. Вона визначає, які події чи групи дій відстежуватимуться. Специфікація на рівні бази даних надає більше інформації про подію та дозволяє краще фільтрувати дані, ніж специфікація на рівні сервера.

2.3.3 SQL Server Profiler

Інструмент SQL Profiler призначений для відстеження всіх подій, що відбуваються в SQL Server. Він використовується для аналізу продуктивності роботи бази даних та пошуку проблем під час виконання запитів. Profiler також може бути використаний для аудиту, у тому числі і аудиту безпеки.

SQL Profiler має візуальний інтерфейс і дозволяє зберігати інформацію про кожну подію в таблиці або файл аудиту у форматі XML. Події поділяються на кілька категорій: активність звичайного користувача, дії адміністратора бази даних, події сервера тощо. Інформація, що реєструється, містить дату і час події, ідентифікатор користувача, тип події, результат, джерело, імена об'єктів, до яких здійснюється доступ.

Крім функцій безпосередньо аудиту, SQL Profiler використовується для налагодження SQL коду. SQL Server забезпечує тонкі можливості моніторингу кожної категорії подій. Наприклад, фіксування процесорного часу, що витрачається на обробку запиту, необхідне оцінки продуктивності.

2.4 Огляд існуючих програм для аудиту

Існує ряд сторонніх продуктів, призначених для спрощення відстеження змін даних у базах даних. Lumigent Entegra, ApexSQL Audit, Log Explorer та NetWrix SQL Server Change Reporter розроблені спеціально для SQL Server.

Ключовими функціями NetWrix є:

- фіксування змін, що вносяться до баз даних, таблиці, схеми, користувачів, повноваження, збережені процедури, логіни та інші об'єкти SQL Server;
- фіксування всіх змін, що вносяться до вмісту бази даних;
- фіксування на ранніх стадіях неавторизованих та небажаних змін, які можуть призвести до «падіння» сервера та бази даних;
- відображення у звітах інформації про те, хто вчинив зміну, коли це сталося, де та з якої робочої станції;
- створення звітів за запитами;

- зберігання даних, зібраних під час аудиту, та можливість створення звітів на підставі цих даних за будь-який період часу.

Log Explorer використовує журнали транзакцій SQL Server разом із журналами користувачів, таким чином отримуючи всю необхідну інформацію. Log Explorer надає такі можливості:

- фіксування всіх змін бази даних;
- скасування або повторення транзакції, внесення необхідних змін;
- відкат рядка до певного моменту часу;
- пошук за віддаленими записами;
- аналіз навантаження на систему.

Log Explorer також має додаткову перевагу – миттєве повідомлення про зміни об'єктів. Програма працює на сервері, відстежує ці події, а потім надсилає докладні звіти електронною поштою.

Пакет Lumigent Entegra дає можливість користувачеві вести контроль та журналювання всіх дій, пов'язаних із переглядом даних, а також визначати зміни даних та структури баз даних. Це допомагає здійснювати та покращувати засоби захисту інформації, підвищуючи при цьому ступінь контролю та знижуючи ризики.

Пакет Entegra є повноцінним альтернативним варіантом більш традиційних і менш універсальних засобів, наприклад, таких, як процедури, що зберігаються, і тригери, експлуатація яких вимагає досить високої кваліфікації і значних вкладень. Крім того, Lumigent Entegra дозволяє відмовитися від інших, не менш складних способів аудиту, пов'язаних із внесенням змін до корпоративних додатків, що вже працюють. Всі ці інструменти не мають можливості фіксувати інформацію про перегляд даних, можуть пропускати зміни у структурі та схемі бази даних, а також не враховують прямий доступ до бази даних та можуть негативно впливати на продуктивність сервера СУБД.

У пакеті Lumigent Entegra для збору інформації про доступ співробітників до даних існують спеціальні агенти аудиту, які не використовують процедури

сервера СУБД, що зберігаються. Зберігання всіх зібраних даних у централізованому загальному сховищі забезпечує зручне управління та дозволяє виконати всі вимоги щодо визначення операцій з даними. Єдина адміністративна консоль полегшує налаштування підконтрольних серверів, включаючи операції із зазначенням конкретних типів операцій із БД, які слід фіксувати. Функції оповіщення допомагають вирішувати проблеми, що виникають, в найстисліші терміни.

Найпопулярнішим продуктом компанії ApexSQL є Universal Studio. Повний пакет програмних інструментів Universal Studio включає такі функції:

- ведення аудиту баз даних та їх відновлення;
- переведення даних у пакети T SQL сценаріїв, їх запуск та встановлення;
- розробка та очищення процедур, що не використовуються;
- документація у форматі .doc;
- впровадження та редагування SQL Server;
- генерація коду;
- синхронізація баз даних;
- створення звітів у HTML.

Програма ApexSQL Audit входить у пакет Universal Studio і призначена для моніторингу інформації та звітності на SQL сервері, автоматично генеруючи тригерну схему щодо аудиту бази даних Microsoft SQL Server.

ApexSQL Audit має такі ключові особливості:

- дані, отримані в ході аудиту, можуть бути зафіксовані у новій базі даних;
- створення звітів, що настроюється;
- моніторинг змін схеми бази даних;
- підтримка SQL 2005;
- сформовані звіти можуть бути переведені у формати Excel, txt, HTML, а також їх можна надрукувати;

- перегляд об'єктів DDL безпосередньо з програми та багато інших функцій та можливостей.

ApexSQL Log – це інструмент Universal Studio, здатний читати активні журнали транзакцій, окремі LDF файли та резервні копії журналів транзакцій, як звичайних, так і тих, що зберігаються в стислому вигляді. Цей інструмент може переглядати всі операції (як DML, так і DDL) та аналізувати, які дані були змінені за допомогою цих операцій. Крім того, можна переглядати логічний вміст LDF файлу, створювати Undo/Redo скрипти, історію зміни будь-якого рядка в базі даних та багато іншого.

Для того, щоб отримати зміни, що відбулися в базі даних журналу транзакцій (його активної частини та резервних копій), ApexSQL Log реконструює всі події в контексті змін. Для досягнення цієї мети ApexSQL Log проходить по всьому ланцюжку журналу транзакцій, включаючи останню повну резервну копію всієї бази даних.

Основні переваги ApexSQL Log:

- об'єднання різних джерел із даними журналу транзакцій в один спільний документ;
- об'єднання активного журналу транзакцій з його резервними копіями та окремими LDF файлами для отримання більш детальної інформації про зміни;
- повна реконструкція операції UPDATE, у тому числі виведення рядка до та після зміни;
- перегляд історії всіх операцій DML над рядками, включаючи час та користувача, який вніс зміни;
- можливість застосування різних фільтрів, які забезпечують швидкий пошук інформації;
- можливість отримання ідентифікатора старої вже віддаленої таблиці для відновлення запису;
- для читання та роботи з LDF файлами не потрібні знання T SQL.

Всі програми для аудиту баз даних, як і вбудовані в СУБД засоби відстеження змін у базі, безумовно, мають свої переваги і можуть бути використані для аудиту БД інформаційних систем. Однак у даній роботі для бази даних інформаційної системи супроводу ремонту буде розроблена унікальна підсистема аудиту, тому що в ній потрібно відслідковувати конкретні зміни на рівні бази даних, а сторонні програми для аудиту, як правило, досить складні за рахунок наявності широкого спектру функцій та особливостей, і при цьому вони є платними. А вбудовані в СУБД засоби аудиту, навпаки, мають обмежені функції та певну структуру відстежуваних даних та журналу аудиту, які не можна змінювати.

3 ПРАКТИЧНА ЧАСТИНА

3.1 Розробка підсистеми аудиту

У рамках даної роботи розроблятиметься підсистема аудиту для конкретної інформаційно-технологічної системи супроводу ремонту, призначеної для автоматизації процесу ремонту змінних елементів. У процесі розробки цієї системи виникла необхідність в аудиті даних бази даних у зв'язку з тим, що на даний момент ця база даних містить близько сімдесяти таблиць, але вона знаходиться в стадії розробки, і кількість таблиць може збільшитися, а також дані в ній можуть часто змінюватись та оновлюватися. Відповідно, ця база даних містить величезну кількість даних, і користувач може змінювати її структуру, створюючи, видаляючи або змінюючи таблиці та дані, що зберігаються в них.

Підсистема аудиту бази даних інформаційної системи супроводу ремонту передбачає відстеження змін у БД лише на рівні бази даних, тобто фіксування змін у всіх таблицях бази даних, викликаних командами вставки, видалення чи зміни даних.

3.1.1 Логічне проектування

Любий екземпляр MS SQL Server може мати починаючи з сервера ієрархічну колекцію сутностей. Усі сервери складаються з багатьох баз даних, а кожна з них в свою чергу з колекції об'єктів для захисту. Кожен об’єкт MS SQL Server який захищається, має пов'язані права доступу, що можуть надаватися окремим особам, групам або процесам що є учасниками, які отримали доступ до MS SQL. Платформа безпеки MS SQL Server дає змогу керувати доступом до сутностей, які будуть захищатися, за допомогою перевірки авторизації та автентичності.

- Аутентифікація — це процес входу, в наслідок якого учасник запитує доступ подаючи облікові дані, що перевіряє сервер. Ідентифікація процесу чи користувача відбувається на етапі автентифікації.
- Авторизація — це процес визначення до яких ресурсів і операцій з цими ресурсами можна надати доступ учаснику.
- У MS SQL Server підтримується два режими автентифікації: режим змішаної автентифікації та режим автентифікації Windows.
- Режим автентифікації Windows - це стандартний режим автентифікації. Оскільки присутня тісна інтеграція моделі безпеки SQL Server з Windows, то її часто вважають вбудованою функцією безпеки. Надається можливість входити до SQL Server для певних користувачів та груп Windows. Як результат користувачам Windows, що автентифікувалися, додаткові облікові дані вже не повинні надавати.
- Режим змішаної автентифікації – це автентифікація з використання як засобів SQL Server так і Windows. У цьому випадку пари імен користувачів та паролів зберігаються в SQL Server.

Рекомендують використовувати режим автентифікації Windows. При автентичності Windows для автентифікації користувачів використовуються багато зашифрованих повідомлень. А при автентичності MS SQL Server конфіденційні ідентифікатори за якими здійснюватиметься вхід MS SQL Server передаватимуться через мережу, що є менш захищеними.

При використанні автентифікації Windows користувачі вже увійшли до Windows і їм не потрібно окремо входити ще і в SQL Server. Наступний рядок підключення SqlConnection.ConnectionString визначає автентифікацію Windows, не вимагаючи імені користувача або пароля.

За потреби використовувати змішаний режим автентифікації потрібно створити ідентифікатори для входу SQL Server, які зберігатимуться у ньому ж. Опісля ці ж ідентифікатори потрібно буде вводити під час виконання.

У MS SQL Server підтримуються наступні механізми шифрування:

- симетричні ключі шифрування;
- асиметричні ключі шифрування;
- сертифікати;
- функції Transact-SQL. У міру того як вставляються або оновлюються якісь певні окремі елементи за допомогою функцій Transact-SQL.

Підписана цифровим підписом інструкція, що пов’язує ідентифікатори користувача, пристрою або служби зі значенням відкритого ключа та має відповідний закритий ключ називають сертифікат або сертифікатом відкритого ключа. Сертифікати надаються та підписуються центром сертифікації.

Як правило, у сертифікатах містять наступні відомості.

- Ідентифікаційні дані: адреса електронної пошти, ім’я тощо;
- Інтервал часу, або термін протягом якого сертифікат варто вважати дійсним;
- Цифровий підпис постачальника. Підпис який підтверджує зв’язок між відкритим, закритим ключем і ідентифікаційними даними. (Під час створення цифрового підпису певні конфіденційні дані відправника разом із звичайними даними перетворюються на тег який називають підписом.);
- Відкритий ключ суб'єкта;
- Дані постачальника сертифіката;
- Прозоре шифрування даних.

Шифрування за допомогою симетричного ключа є спеціальним випадком прозорого шифрування даних. Використовуючи симетричний ключ, який ще називають ключем шифрування бази даних, TDE шифрує всю базу даних. Ключ шифрування захищений сертифікатами або іншими ключами, які вже в свою чергу захищаються асиметричним ключем або головним ключем, що зберігається в модулі розширеного керування ключами.

Для реалізації цієї підсистеми аудиту було обрано підхід до аудиту на основі тригерів, оскільки цей підхід у цьому випадку має ряд переваг над аудитом журналу транзакцій.

На етапі логічного проектування має бути сформована логічна структура підсистеми аудиту бази даних, саме логічна структура її таблиць і тригерів.

Як було зазначено раніше журнал аудиту являє собою систему з трьох таблиці, створюваних для кожної таблиці бази даних (таблиця AuditInf – таблиця для запису загальної інформації про дані, що змінюються; таблиця UpdateLog призначена для фіксації змін, що відбулися в результаті виконання операції зміни (UPDATE); для фіксації змін даних при виконанні операцій вставки (INSERT) та видалення (DELETE) служить таблиця InstDelLog). Таблиці аудиту реалізуються згідно зі схемою, представленою малюнку 1.

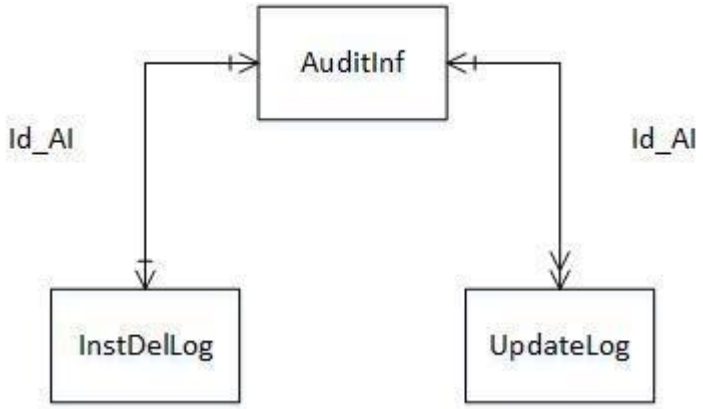

Рис. 1. Схема журналу аудиту

Таблиця InstDelLog також містить поля id_LOG (ідентифікатор, що є первинним ключем) та id_AI (Зовнішній ключ до загальної таблиці аудиту AuditInf). З іншого боку, до них приєднуються змінені чи видалені стовпці вихідної таблиці. Таким чином структура таблиці InstDelLog для кожної таблиці, що аудується, буде унікальна.

Основна проблема реалізації таблиць аудиту у тому, що вихідна база даних містить велику кількість таблиць. Розробка бази даних продовжується і кількість таблиць може збільшитися. Тому необхідно реалізувати автоматичне визначення існуючих у базі даних таблиць і створення їм таблиць аудиту, і навіть можливість створювати журнал аудиту для конкретної таблиці, заданої користувачем.

3.1.2 Інфологічне проектування

Підсистема аудиту - це функція MS SQL Server, яка дозволяє налаштовувати параметри перевірки та перевіряти події ядра бази даних. Для запису використовуються розширені події. Підсистема має усі необхідні засоби та процеси щоб зберігати, проводити та переглядати налаштування аудиту для різних об’єктів баз даних та серверів.

Функція трасування працює повільніше ніж підсистема аудиту SQL Server, а середовище SQL Server Management Studio спрощує контроль та створення журналів аудиту. Тепер можна робити детальні перевірки, а саме відслідковувати інструкції EXECUTE, UPDATE, INSERT, SELECT та REFERENCEFS для окремих користувачів. В добавок, підсистема аудиту сповна підтримує інструкції CREATE SERVER AUDIT SPECIFICATION та T-SQL CREATE SERVER AUDIT і пов'язані з ними інструкції ALTER та DROP.

Для конфігурації аудиту необхідно вказати місце запису і за його відсутності створити. Аудит можна зберігати в будь-якому файлі, журналі безпеки Windows або журналі програм Windows. Можна присвоїти аудиту ім'я та налаштувати його характеристики, в тому числі шлях до файлу аудиту та його максимальний розмір. Також можна налаштувати аудит так чином щоб у разі збою перевірки робота SQL Server припинялася. При необхідності записувати дані аудиту в кілька журналів, створюється декілька аудитів.

Наступний етап – створення специфікацій аудиту. У специфікації аудиту збирається інформація про версію SQL Server; до неї відносять об'єкти, що

стосуються серверу: членство в серверних ролях. дані облікових записів. Наявна інформація про бази даних яка контролюється в основній базі даних, як от інформація про права доступу до бази. Під час створення специфікації аудиту вказується в котрий аудит надходитимуть дії які спостерігаються. Аудит може мати лише одну активну специфікацію, проте можна створити декілька аудитів сервера ті декілька специфікацій аудиту і перемикатися між ними.

Можна створити специфікації аудиту бази даних які будуть використовуватись для відстеження подій в окремій базі даних. Активною може бути лише одна специфікація аудиту проте до аудиту можна додати декілька специфікацій аудиту баз даних і перемикатися між ними при потребі.

В колекції пов’язаних подій об’єднуються події підсистеми аудиту SQL Server, що використовуються у специфікаціях аудиту сервера і надаються як групи дій аудиту. Подій включені до групи можна буде відслідковувати якщо таку групу додати до специфікації аудиту.

Існує 35 груп дій аудиту сервера і деякі з них тісно пов’язані одне з одним.

Групи подій аудиту, що зібрані на рівні баз даних, можна вказувати в специфікаціях аудиту. Специфікації аудиту баз даних можуть включати певні окремі події аудиту, які дозволяють перевіряти інструкції відповідальні за роботу з даними. Дані події можна налаштувати щоб відстежували усю базу данихабо лише певні об’єкти. Зокрема, дія SELECT, може бути використана для перевірки запитів SELECT, звернених як до окремої таблиці, так і до всієї схеми в цілому. Такі подій можна налаштувати таким чином щоб вони відслідковували дії за окремими ролями або користувачами, наприклад db_writers.

Припустимо, дію типу SELECT можна використати для перевірки запитів SELECT, адресованих до окремої таблиці, від імені користувача Mary, роль FINANCE_DEPT, або роль бази даних Public. Не можна не зазначити, що завдяки цьому в руках адміністратора з’являються набагато ширші можливості контролю який дає великий запас гнучкості під час налаштуванню аудиту.

Рис. 2. Концепція забезпечення безпеки баз даних через механізм ролей

Необхідний захист важливих даних, які зберігаються у базах даних SQL Serverб забезпечується компонентом резервного копіювання та відновлення. Для мінімізації ризику незворотної втрати даних виникає необхідність регулярно створювати резервні копії бази даних в яких в свою чергу зберігатимуться зміни даних, що проводяться. Втрати даних у разі пошкоджень через різні збої захищає добре продумана стратегія резервного копіювання та відновлення. Перевірити обрану стратегію можна виконавши відновлення баз даних використовуючи резервну копію, це допоможе у майбутньому ефективно відреагувати на можливі проблеми.

Маючи правильно створені резервні копії баз даних можна відновити дані після багатьох видів збоїв, в тому числі:

- збої обладнання (наприклад, пошкоджений дисковий накопичувач або безповоротна втрата даних на сервері);
- збій носія;
- стихійні лиха;

- помилки користувачів (наприклад, видалення таблиці помилково).

Стратегія відновлення та резервування складається з частин, що відносяться до відновлення і до резервування. Частина про резервування визначає частоту і тип створення резервних копій, тип і швидкісні характеристики обладнання яке необхідне для створення, метод перевірки резервних копій і місцезнаходження та тип носія для резервних копій (включаючи питання безпеки). Частина про відновлення визначає відповідальну особу за методи проведення і проведення операцій відновлення що дозволяють дотриматися вимог користувачів щодо мінімізації втрат та доступності даних. Рекомендується задокументувати процедури відновлення та резервування і зберігати копію цієї документації у документації за завданням.

В SQL Server є функція, яка дозволяє виконувати відновлення та резервне копіювання з використання служби сховищ великих двійкових об'єктів Windows Azure. Це можна використовувати для резервного копіювання SQL Server розміщеного, наприклад, на віртуальній машині Windows Azure. Безлімітне віддалене зберігання і простота міграції даних у хмару та назад, саме ці переваги надає резервне копіювання у хмару. В даній версії можна застосовувати інструкції RESTORE та BACKUP.

Переваги використання служби сховища великих двійкових об'єктів Windows Azure для резервного копіювання:

- На даний момент для екземплярів SQL Server, запущених у віртуальній машині Windows Azure, створення резервних копій за допомогою служби сховища великих двійкових об'єктів Windows Azure можна виконувати лише створивши підключені диски. Проте, кількість дисків, що можна підключити до віртуальної машини Windows Azure, обмежена. Для надзвичайно великих екземплярів це обмеження становить 16 дисків, а для екземплярів поменше ця кількість менша. При виконанні резервного копіювання безпосередньо в сховищі великих двійкових об'єктів Windows Azure є можливість обійти межу в 16 дисків. Також, файл резервної копії, що зберігається у службі сховища великих двійкових об'єктів Windows Azure,

безпосередньо доступний у віддаленій службі SQL Server у віртуальній машині або локальній службі SQL Server. Повторне від’єднання чи приєднання бази даних чи завантаження та підключення віртуального накопичувача не потрібне;

- Гнучкість, надійність і безлімітне віддалене сховище: Реалізація доступу до даних з-за меж обчислювальної системи – одна з переваг при зберіганні резервних копій за допомогою служби сховища великих двійкових об'єктів Windows Azure, що являється зручним та гнучким рішенням. Щоб одна аварія не вплинула одночасно і на віддалену копію, і на робочий екземпляр - сховище зазвичай розташовують доволі далеко від робочого екземпляру. Окрім цього, отримати доступ до резервних копій можна в будь-якому місці і в будь-який час, в тому числі так само легко можна отримати і доступ для відновлення;
- Відсутність потреби підтримувати апаратуру: служби Windows Azure не потребують підтримки обладнання. Служби Windows Azure самі керують своїм апаратним забезпеченням і забезпечують геореплікацію для захисту та надмірності від збоїв обладнання;
- Архів резервних копій: Служба сховища великих двійкових об'єктів Windows Azure — це найкраща рішення для створення архіву резервних копій, що часто використовується на носії. При використанні фізичних носіїв може бути потрібним фізично транспортувати носій у віддалене приміщення та вживати певні заходи для їхнього захисту. Зберігавши резервні копії у сховищі великих двійкових об'єктів Windows Azure забезпечує доступне, швидке й надійне архівування;
- Економічні переваги: Оплата здійснюється лише за використовуваний об’єм послуг. Може використовуватися як економічно ефективне рішення для віддаленого резервного копіювання та архівування.

Етап інфологічного проектування підсистеми аудиту бази даних передбачає опис предметної області, моделі бази даних, її функціональність, а також особливості, які необхідно враховувати при проектуванні.

Підсистема аудиту бази даних описаної вище інформаційної системи повинна містити відомості про всі зміни, що відбуваються в таблицях бази даних,

тобто. про видалення, вставку та зміну даних у них. Підсистема аудиту розробляється для того, щоб у разі відмови або будь-яких некоректних дій або помилок з боку користувачів можна було подивитися в журналі аудиту, що сталося, коли це сталося і ким були вчинені ці дії.

Відповідно до поставленого завдання, методу аудиту та структури журналу аудиту, обраних раніше, підсистема аудиту будується з урахуванням таких особливостей:

1. Для реалізації підсистеми необхідно створити загальний для всіх таблиць бази даних тригер, що записує зміни даних БД в журнал аудиту.

2. Журнал аудиту є додатково створені у базі даних таблиці, у яких зберігаються всі зміни вихідних таблиць.

3. Кожна вихідна таблиця бази даних має три таблиці для аудиту. У першу таблицю записується загальна інформація про зміни (ким були внесені зміни, з якого комп'ютера, чи вдала спроба зміни). У другу таблицю заноситься повний рядок даних під час операцій вставки та видалення. Третя таблиця призначена фіксувати зміненого значення під час операції UPDATE.

4. Загальна таблиця для аудиту пов'язана з другою таблицею ставленням один до одного, а з третьою – ставленням один до багатьох.

3.1.3 Фізичне проектування

Для того, щоб закріпити за кожною з вихідних таблиць тригер, що фіксують зміни в них, необхідно автоматизувати цей процес, описавши ці дії один раз, але для всіх таблиць. Тому після створення таблиць для аудиту, в яких зберігатиметься інформація про всі зміни даних у базі даних, необхідно створити один тригер, який записуватиме всі ці зміни до потрібних таблиць.

Щоб можна було без додаткових дій запустити підсистему аудиту, її структура буде виглядати так: буде створено процедуру, запустивши яку підсистема аудиту і почне свою роботу, в ній спочатку будуть визначені всі

таблиці в базі даних, потім створені таблиці для аудиту і запущено тригер, записує у яких зміни даних.

Після створення таблиць для аудиту, в яких зберігатиметься інформація про всі зміни даних у базі даних, було створено один тригер, який записуватиме всі ці зміни до потрібних таблиць.

Тригери є особливим типом процедур, що зберігаються, які автоматично запускаються сервером при спробі користувача змінити дані в таблицях, з якими пов'язані ці тригери. Кожен тригер може бути прив'язаний лише до конкретної таблиці, і всі вироблені їм модифікації даних представляються як транзакція. У нашому випадку тригери для кожної з вихідних таблиць матимуть однакову структуру, тому є сенс створити єдиний тригер і використовувати його для кожної таблиці, підставляючи певні значення.

Тригер є дуже корисним, але водночас і дуже небезпечним засобом, тому що при його неправильному проектуванні та запуску можуть статися серйозні негативні наслідки, наприклад, може бути видалена ціла база даних або її частина. Саме тому тригери потребують ретельної перевірки та налагодження. У реалізації СУБД Microsoft SQL Server використовується структура створення чи зміни тригера, представлена малюнку 7.

```
CREATE [ OR ALTER ] TRIGGER [ schema_name . ]trigger_name
ON { table | view }
[ WITH <dml_trigger_option> [ ,...n ] ]
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
[ WITH APPEND ]
[ NOT FOR REPLICATION ]
AS { sql_statement  [ ; ] [ ,...n ] | EXTERNAL NAME <method specifier [ ; ] > }

<dml_trigger_option> ::=
    [ ENCRYPTION ]
    [ EXECUTE AS Clause ]

<method_specifier> ::=
    assembly_name.class_name.method_name
```

Рис. 3. Структура тригера в Microsoft SQL Server

У нашому випадку створено тригер, який викликається замість виконання команд (тригер INSTEAD OF), тобто, перевизначає дії тригерних операцій, і реагує всі три можливі у разі команди: DELETE, INSERT і UPDATE.

3.2 Стрес-тестування

Стрес-тестування виконується для знаходження точки відмови системи. В цьому тестуванні база даних навантажується таким чином, що система виходить з ладу в одній точці. Ця точка називається точкою відмови системи баз даних.

Певний стан транзакцій бази даних потребує значних зусиль. Як результат необхідне правильне планування щоб уникнути любих проблем часу і затрат.

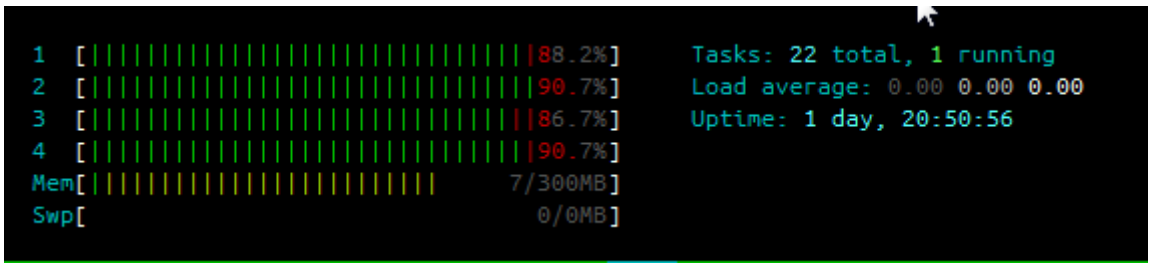

Рис. 4. Навантаження на систему під час стрес-тесту

4 ОХОРОНА ПРАЦІ ТА ФАКТОРИ ЩО ВПЛИВАЮТЬ НА ФУКНЦІОНАЛЬНИЙ СТАН КОРИСТУВАЧІВ КОМП'ЮТЕРІВ

4.1 Охорона праці

В Україні існує Закон України «Про охорону праці», а також інші підзаконні нормативно-правові акти. У відповідності до вимог ст. 153 Кодексу законів про працю України та ст. 6 Закону України «Про охорону праці» на всіх підприємствах, в установах, організаціях створюються безпечні і нешкідливі умови праці.

Згідно закону України "Про охорону праці", в останній редакції 2018 року, охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних, лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людини в процесі трудової діяльності. Дія цього Закону поширюється на всіх юридичних та фізичних осіб, які використовують найману працю відповідно до законодавства, та поширюється на всіх працівників.

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. Власник або уповноважений ним орган повинен впроваджувати сучасні засоби техніки безпеки, які запобігають виробничому травматизму, і забезпечувати санітарно-гігієнічні умови, що запобігають виникненню професійних захворювань працівників. Стаття 158 Кодексу законів про працю України встановлює обов'язок власника або уповноваженого ним органу вживати заходів щодо полегшення і оздоровлення умов праці працівників шляхом впровадження прогресивних технологій, досягнень науки і техніки, засобів механізації та автоматизації виробництва, вимог ергономіки, позитивного досвіду з охорони праці, зниження та усунення запиленості та

загазованості повітря у виробничих приміщеннях, зниження інтенсивності шуму, вібрації, випромінювань тощо. А згідно з ч. 1 ст. 13 Закону України «Про охорону праці» роботодавець зобов'язаний створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці.

Роботодавець зобов'язаний створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці. З цією метою роботодавець забезпечує функціонування системи управління охороною праці, а саме:

- створює відповідні служби і призначає посадових осіб, які забезпечують вирішення конкретних питань охорони праці, затверджує інструкції про їх обов'язки, права та відповідальність за виконання покладених на них функцій, а також контролює їх дотримання;
- розробляє за участю сторін колективного договору і реалізує комплексні заходи для досягнення встановлених нормативів та підвищення існуючого рівня охорони праці;
- забезпечує виконання необхідних профілактичних заходів відповідно до обставин, що змінюються;
- впроваджує прогресивні технології, досягнення науки і техніки, засоби механізації та автоматизації виробництва, вимоги ергономіки, позитивний досвід з охорони праці тощо;
- забезпечує належне утримання будівель і споруд, виробничого обладнання та устаткування, моніторинг за їх технічним станом;
- забезпечує усунення причин, що призводять до нещасних випадків, професійних захворювань, та здійснення профілактичних заходів, визначених комісіями за підсумками розслідування цих причин;

- організовує проведення аудиту охорони праці, лабораторних досліджень умов праці, оцінку технічного стану виробничого обладнання та устаткування, атестацій робочих місць на відповідність нормативно-правовим актам з охорони праці в порядку і строки, що визначаються законодавством, та за їх підсумками вживає заходів до усунення небезпечних і шкідливих для здоров'я виробничих факторів;
- розробляє і затверджує положення, інструкції, інші акти з охорони праці, що діють у межах підприємства (далі – акти підприємства), та встановлюють правила виконання робіт і поведінки працівників на території підприємства, у виробничих приміщеннях, на будівельних майданчиках, робочих місцях відповідно до нормативно-правових актів з охорони праці, забезпечує безоплатно працівників нормативно-правовими актами та актами підприємства з охорони праці;
- здійснює контроль за додержанням працівником технологічних процесів, правил поводження з машинами, механізмами, устаткуванням та іншими засобами виробництва, використанням засобів колективного та індивідуального захисту, виконанням робіт відповідно до вимог з охорони праці;
- організовує пропаганду безпечних методів праці та співробітництво з працівниками у галузі охорони праці;
- вживає термінових заходів для допомоги потерпілим, залучає за необхідності професійні аварійно-рятувальні формування у разі виникнення на підприємстві аварій та нещасних випадків.
- Роботодавець несе безпосередню відповідальність за порушення зазначених вимог.

Вимоги безпеки до робочих місць працівників з екранними пристроями передбачають:

1. Робочі місця працівників з екранними пристроями мають бути спроектовані так і мати такі розміри, щоб працівники мали простір для зміни робочого положення та рухів.

2. Для забезпечення безпеки та захисту здоров’я працівників усе випромінювання від екранних пристроїв має бути зведене до гранично допустимого рівня з погляду безпеки та охорони здоров’я працівників.

3. Організація робочого місця працівника з екранними пристроями має забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним, антропологічним, психофізіологічним вимогам, а також характеру виконуваних робіт.

4. Освітлення робочого місця працівника з екранними пристроями має створювати відповідний контраст між екраном і навколишнім середовищем (з урахуванням виду роботи) та відповідати вимогам ДСанПІН 3.3.2.007-98.

5. Мікроклімат виробничих приміщень з робочими місцями працівників з екранними пристроями має підтримуватись на постійному рівні та відповідати вимогам Санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042-99, затверджених постановою Головного державного санітарного лікаря України від 01 грудня 1999 року № 42 (далі - ДСН 3.3.6.042-99).

6. Робочий стіл або робоча поверхня повинні бути достатнього розміру та мати поверхню з низькою відбивною здатністю, допускати гнучкість під час розміщення екрана, клавіатури, документів і відповідного устаткування.

7. Робоче крісло має бути стійким і дозволяти працівнику з екранними пристроями легко рухатися та займати зручне положення. Сидіння має регулюватися по висоті, спинка сидіння - як по висоті, так і по нахилу.

4.2 Фактори що впливають на функціональний стан користувачів комп’ютерів

Трудова діяльність користувачів комп'ютерів (ВДТ) відбувається у певному виробничому середовищі, яке впливає на їх функціональний стан.

Найбільш значимі — фізичні фактори виробничого середовища, до яких належать електромагнітні хвилі різних частотних діапазонів, електростатичні поля, шум, параметри мікроклімату та ціла низка світлотехнічних показників. Вплив хімічних та, особливо, біологічних факторів виробничого середовища на користувачів комп'ютерів — значно менший.

Трудовий процес суттєво впливає на психофізіологічні можливості користувачів комп'ютерів, оскільки їх діяльність характеризується значними статичними фізичними навантаженнями; недостатньою руховою активністю; напруженнями сенсорного апарату, вищих нервових центрів, які забезпечують функції уваги, мислення, регуляції рухів. Окрім того, трудовий процес користувачів комп'ютерів відзначається значними інформаційними навантаженнями.

Професійні якості та виробничий досвід, які визначають внутрішні засоби діяльності, обумовлюють надійну та безпомилкову діяльність користувачів комп'ютерів, дозволяють знаходити безпечні методи розв'язання виробничих завдань навіть у нестандартних ситуаціях.

Зовнішні засоби діяльності, які в основному визначаються ергономічними показниками щодо організації робочого місця, форми та параметрів його елементів, просторового розташування основного і допоміжного устаткування, можуть суттєво знизити фізичні та психофізіологічні навантаження, що діють на користувачів комп'ютерів.

Особливості роботи користувачів комп'ютерів

У професійних операторів частіше зустрічаються порушення органів зору, опорно-рухового апарату, центральної нервової, серцево-судинної, імунної та статевої систем, захворювання шкіри. Зафіксована значна кількість скарг операторського персоналу на загальне недомагання, передчасне стомлювання, головний біль, порушення функцій органів зору, які здійснювали несприятливий психофізіологічний вплив на самопочуття та працездатність операторів.

Сучасна професія користувача ВДТ належить до розумової праці, яка характеризується: високою напруженістю зорових функцій; одноманітною

позою; великою кількістю стереотипних висококоординованих рухів, що виконуються лише м'язами кистей рук на фоні малої загальної рухової активності; значним нервово-емоційним компонентом, особливо в умовах дефіциту часу; роботою з великими масивами інформації, що викликає активізацію уваги та інших вищих психічних функцій. Крім того, при роботі з дисплеями на електронно-променевих трубках виникає вплив на користувача цілої низки факторів фізичної природи — електростатичні поля, радіочастотне та рентгенівське випромінювання тощо.

Діяльність професіоналів можна поділити на три групи:

1. Діяльність, яка пов'язана з виконанням нескладних багаторазово повторюваних операцій, що не вимагають великого розумового напруження. Наприклад, робота операторів комп'ютерного набору, працівників довідкових служб.

2. Діяльність, яка пов'язана із здійсненням логічних операцій, що постійно повторюються. Це робота інженера-економіста, інженера-проектувальника, оператора автоматизованого виробництва.

3. Діяльність, коли в процесі роботи необхідно приймати рішення за відсутності заздалегідь відомого алгоритму. Наприклад, робота інженера-програміста, диспетчерів руху залізничного транспорту, аеропортів тощо.

У користувачів, які інтенсивно використовують комп'ютер в умовах значних розумових напружень досить часто (40—70%) виникають психологічні та поведінкові порушення (нервозність, роздратування, тривога, нерішучість, замкнутість тощо). Серед користувачів ВДТ в США і Європі значного поширення набуло специфічне захворювання, яке отримало назву синдром комп'ютерного стресу (СКС). СКС супроводжується головним болем, запаленням очей, алергією, роздратованістю, млявістю і депресією. Інформаційне перевантаження користувачів ВДТ супроводжується низкою специфічних захворювань, які називають інформаційними. Першим симптомом їх є головний біль. Дослідження, проведені в США, Німеччині, Швейцарії та інших країнах, показали, що робота з обслуговування ВДТ супроводжується

підвищеним напруженням зору, інтенсивністю і монотонністю праці, збільшенням статичних навантажень, нервово-психічним напруженням, впливом різного виду випромінювань та ін. Внаслідок цього серед операторів ВДТ, як зазначають фахівці Всесвітньої організації охорони здоров'я, частіше, ніж в інших групах працюючих, трапляються такі професійні захворювання, як передчасна стомлюваність, погіршення зору, м'язові і головні болі, психічні й нервові розлади, хвороби серцево-судинної системи, онкологічні захворювання та ін. Вважається, що стан організму операторів ВДТ визначається комплексним впливом факторів трудового процесу і середовища, значення яких є неоднаковим. На операторів з малим стажем роботи на ВДТ домінуючий вплив чинять фактори середовища, а на операторів зі стажем понад 5 років - фактори трудового процесу.

ВИСНОВКИ

У ході виконання цієї роботи було розроблено підсистему аудиту бази даних інформаційної системи супроводу ремонту. Актуальність даної теми обумовлена необхідністю автоматизації процесу відстеження можливих змін у базі даних для запобігання помилок користувачів та несанкціонованих дій щодо бази даних, а також можливості її відновлення.

Для реалізації цієї підсистеми аудиту було проведено теоретичну та практичну роботу, з якої можна зробити такі основні висновки:

- підхід до аудиту бази даних доцільно вибирати залежно від структури бази даних та об'єктів, що зазнають аудиту;
- вбудовані в СУБД засоби відстеження змін та програми для аудиту найчастіше засновані на моніторингу журналу транзакцій, що може сильно впливати на продуктивність системи;
- аудит за допомогою тригерів можна реалізувати самостійно, налаштувавши при цьому, які конкретно зміни у базі даних фіксуватимуться;
- залежно від вимог до системи розміщення журналу аудиту може бути здійснено в окремому файлі або у додаткових таблицях;
- важливим етапом розробки підсистеми аудиту є вибір СУБД, який повинен задовольняти основним вимогам, що висуваються до інформаційної системи та програмного забезпечення.

Головним результатом, отриманим у ході виконання роботи і відповідним переліку поставлених завдань, є розробка підсистеми аудиту для бази даних конкретної інформаційної системи за попередньо обраним методом аудиту за допомогою тригерів та з використанням журналу аудиту, що є по три додаткові таблиці до кожної вихідної таблиці БД. Ця підсистема аудиту була розроблена з використанням СУБД Microsoft SQL Server. Незважаючи на те, що дана підсистема аудиту була розроблена для конкретної бази даних інформаційної

системи супроводу ремонту, вона може бути застосована і до систем, що містять бази даних, реалізовані за допомогою СУБД Microsoft SQL Server.

До переваг розробленого продукту можна віднести відсутність значних втрат продуктивності системи при роботі з підсистемою аудиту, проте продукт не є ідеальним, а його основний недолік – безліч додаткових таблиць.

Надалі об'єкт розробки може бути вдосконалений за рахунок розширення функціоналу підсистеми аудиту, наприклад, вона може включати відстеження змін не тільки в таблицях бази даних, але і в її структурі.

У результаті підготовки теоретичної частини роботи мною був проведений огляд підходів до аудиту баз даних та обраний підхід до аудиту на основі тригерів, а також обґрунтовано вибір програмного забезпечення для розробки підсистеми аудиту. Результатами написання теорії став опис підходів до реалізації структури журналу аудиту та вибір як журнал аудиту додаткових таблиць..

Результатом виконання практичної частини стала розробка підсистеми аудиту бази даних інформаційної системи супроводу ремонту, а саме її інфологічне, логічне та фізичне проектування.

БІБЛІОГРАФІЯ

1. 10 найкорисніших функцій SQL Server 2008 для адміністратора бази даних. URL: http://sqlcom.ru/dba-tools/top-10-helpful-functions-for-dba/ (дата звернення: 13.04.2017).

2. Аудит з журналу транзакцій. URL: https://www.osp.ru/os/2012/01/13012925/ (дата звернення: 12.04.2017).

3. Нільсен П. Microsoft SQL Server 2005. Біблія користувача: Пер. з англ. - М.: ТОВ “І.Д. Вільямс, 2008. - 1232 с.: іл.

4. Станек У.Р. Microsoft SQL Server 2005. Довідник адміністратора/Пер. з англ. - М: Видавництво «Російська Редакція», 2006. - 544 с.: іл.

5. Auditing Advice. URL: http://sqlmag.com/t-sql/auditing-advice (дата звернення: 14.04.2017).

6. Аудит змін структури БД, даних та протоколювання дій користувача на прикладі СУБД Oracle. URL: http://rsdn.org/article/db/db_audit.xml (дата звернення: 14.04.2017).

7. Кривоніс Н. Журналування змін структури БД та даних. URL: http://www.compdoc.ru/bd/sql/log_change_of_structure_bd/ (дата звернення: 14.04.2017).

8. Система управління базами даних. URL: https://ua.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85 (дата звернення: 14.04.2017).

9. Нікітін М. Чи закінчилася епоха реляційних СУБД? URL: http://www.cnews.ru/reviews/free/marketBD/articles/articles2.shtml (дата звернення: 12.04.2017).

10. Полякова Л. Тригери: створення та застосування. URL: http://www.intuit.ru/studies/courses/5/5/lecture/148?page=2 (дата звернення: 03.05.2017).

11. Іцик Бен-Ган. Microsoft SQL Server 2008. Основи T-SQL: Пров. з англ. - СПб.: БВХ-Петербург, 2009. - 432 с.: іл.

12. Фараонов В.В. Delphi 2005. Розробка програм для баз даних та Інтернету. - СПб.: Пітер, 2006. - 603 с.: Іл.

13. Cantu M. Mastering Delphi 7. CA: Sybex Inc., 2003.

14. Roff JT. ADO: ActiveX Data Objects. CA: O'Reilly & Associates, 2001.

15. Cantu M. Data access dilemma. URL: http://www.marcocantu.com/papers/DataADO.htm (дата звернення: 14.05.2017).

16. C. J. Date SQL and Relational Theory: How to Write Accurate SQL Code. O'Reilly, 2009.

17. Spetic A. Transact-SQL Cookbook. O'Reilly, 2002.

18. Shah R. Microsoft SQL Server 2012 Performance Tuning Cookbook 2012.

19. A. Molinaro SQL Cookbook: Query Solutions and Techniques for Database Developers O'Reilly, 2005.

20. Z. Pavlovic, M. Veselica Oracle Database 12c Security Cookbook 2016.

21. M. Gertz, S. Jajodina Handbook of Database Security: Applications and Trends 2008.

ДОДАТКИ

Додаток А

Лістинг головної процедури

```
CREATE PROCEDURE new_proc
AS
DECLARE @Value AS varchar(100), @str_name AS varchar(200), @SQL AS varchar(8000);
CREATE TABLE MainAudInf (ID int NOT NULL IDENTITY PRIMARY KEY,
Tab_NAME sysname,
isAud bit);
CREATE TABLE T_NAME (TAB_NAME sysname)
INSERT INTO T_NAME
SELECT TABLE_NAME
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_NAME != 'sysdiagrams' AND TABLE_NAME != 'T_NAME' AND TABLE_NAME != 'MainAudInf'

-- Оголошуємо курсор curName
DECLARE curName CURSOR FOR
SELECT TAB_NAME
FROM T_NAME

- Відкриваємо курсор
OPEN curName

- Отримуємо перше значення
FETCH NEXT FROM curName INTO @Value
- Цикл за записами
WHILE @@FETCH_STATUS = 0
BEGIN

--таблиця із загальною інформацією про аудит
SET @str_name = @Value + 'AuditInf';
SET @SQL = 'CREATE TABLE' + @str_name +
' (id int NOT NULL IDENTITY PRIMARY KEY,
type char(6) NOT NULL,
table_name nvarchar(200) NOT NULL,
date_change datetime NOT NULL DEFAULT(GETDATE()),
user_change SYSNAME NOT NULL DEFAULT(SUSER_SNAME()),
app SYSNAME NOT NULL DEFAULT(APP_NAME()) ";
exec(@SQL);

--таблиця з детальною інформацією під час операції UPDATE
```

```
DECLARE @c_n varchar(50), @c_t varchar(50);
EXEC ident_PK @Value, @c_n OUTPUT, @c_t OUTPUT;
SET @str_name = @Value + 'UpdateLog';
SET @SQL = 'CREATE TABLE' + @str_name +
' (column_name SYSNAME NOT NULL,
old_value sql_variant NULL,
new_value sql_variant NULL)';
exec(@SQL);
SET @SQL = 'SELECT a.*, b.' + @c_n + 'as id_rec INTO helpTable FROM' +
@str_name + 'as a cross JOIN' + @Value + 'as b';
exec(@SQL);
truncate table helpTable;
ALTER TABLE helpTable Додати id_LOG int IDENTITY PRIMARY KEY;
SET @SQL = 'ALTER TABLE helpTable ADD id_AI INT NOT NULL
REFERENCES '
+ @Value + 'AuditInf';
exec(@SQL);
SET @SQL = 'DROP TABLE' + @str_name;
exec(@SQL);
EXEC sp_rename 'helpTable', @str_name;

--таблиця з детальною інформацією при операціях Insert та Delete
SET @str_name = @Value + 'InstDelLog';
SET @SQL = 'select * into' + @str_name + 'from' + @Value + 'where 0 = 1';
exec(@SQL);
SET @SQL = 'ALTER TABLE' + @str_name +
' add id_AI INT NOT NULL REFERENCES ' + @Value + 'AuditInf,' +
' id_LOG INT NOT NULL IDENTITY PRIMARY KEY ';
exec(@SQL);

-- створення тригерів
EXEC trig_aud_proc @ Value;
INSERT INTO MainAudInf VALUES (@Value, 1);

- Створення уявлень
SET @SQL =
'CREATE VIEW Update_Table_' + @Value +
'AS
SELECT a.date_change, a.user_change, a.app,
b.column_name, b.old_value, b.new_value, b.id_rec
FROM '+ @Value + 'AuditInf AS a,' + @Value + 'UpdateLog AS b
WHERE a.id=b.id_AI';
exec(@SQL);
SET @SQL =
'CREATE VIEW Delet_Table_' + @Value +
```

```
'AS
SELECT a.date_change, a.user_change, a.app, b.*
FROM '+ @Value + 'AuditInf AS a, '+ @Value + 'InstDelLog AS b
WHERE a.id=b.id_AI and a.type = "Delete"';
exec(@SQL);
SET @SQL =
'CREATE VIEW Insert_Table_' + @Value +
'AS
SELECT a.date_change, a.user_change, a.app, b.*
FROM '+ @Value + 'AuditInf AS a, '+ @Value + 'InstDelLog AS b
WHERE a.id=b.id_AI and a.type = "Insert"';
exec(@SQL);
FETCH NEXT FROM curName INTO @Value
END

- Закриваємо курсор
CLOSE curName
DEALLOCATE curName
DROP TABLE T_NAME
GO
```

Додаток Б

Лістинг тригера

```
CREATE PROCEDURE trig_aud_proc @name_Table nvarchar(200)
AS
DECLARE @SQL varchar(8000),
@o_type_U AS char(6), @o_type_I AS char(6), @o_type_D AS char(6),
@name_audinf AS varchar(200), @name_audinsdel AS varchar(200),
@name_audupd AS varchar(200), @Value1_cop AS varchar(100),
@PK_n varchar(50), @PK_t varchar(50);
EXEC ident_PK @name_Table, @PK_n OUTPUT, @PK_t OUTPUT;
SET @o_type_U = 'Update';
SET @o_type_I = 'Insert';
SET @o_type_D = 'Delete';
SET @name_audinf = @name_Table + 'AuditInf';
SET @name_audupd = @name_Table + 'UpdateLog';
SET @name_audinsdel = @name_Table + 'InstDelLog';
SET @SQL = 'CREATE TRIGGER' + @name_Table + '_trig_aud ON' +
@name_Table +
' FOR INSERT, UPDATE, DELETE AS
DECLARE @o_type AS char(6), @headerid AS int, @Value1 AS varchar(100),
@Value2
AS varchar(100)
BEGIN
SELECT * INTO my_inserted FROM inserted;
SELECT * INTO my_deleted FROM deleted;
IF EXISTS(SELECT * FROM inserted)
IF EXISTS(SELECT * FROM deleted)
SET @o_type = "Update"; '+
'ELSE
SET @o_type = "Insert"; '+
'ELSE
SET @o_type = "Delete"; '+
'INSERT INTO' + @name_audinf + '(type, table_name)
VALUES(@o_type, '' + @name_Table + ''); '+
'SET @headerid = SCOPE_IDENTITY();
IF (@o_type = "Update")
BEGIN
DECLARE curName1 CURSOR FOR
select COLUMN_NAME from INFORMATION_SCHEMA.columns
where TABLE_NAME= '' + @name_Table + '';
OPEN curName1
FETCH NEXT FROM curName1 INTO @Value1
WHILE @@FETCH_STATUS = 0
```

```
BEGIN
DECLARE @c_n varchar(50), @c_t varchar(50);
EXEC ident_PK '" + @name_Table + "', @c_n OUTPUT, @c_t OUTPUT;
DECLARE @Value_help3 AS varchar(100);
SET @Value_help3 = "my_inserted." + @c_n;
EXEC sp_rename @Value_help3, 'prim_help_ID', 'COLUMN';
if ((@Value1 != "prim_help_ID") AND (@Value1 != @c_n))
BEGIN
DECLARE @Value_help1 AS varchar(100);
SET @Value_help1 = "my_inserted." + @Value1;
EXEC sp_rename @Value_help1, 'change_help_ID', 'COLUMN';
DECLARE @Value_help2 AS varchar(100);
SET @Value_help2 = "my_deleted." + @Value1;
EXEC sp_rename @Value_help2, 'change_help_ID', 'COLUMN';
if (select change_help_ID from my_inserted) != (select change_help_ID from
my_deleted)
BEGIN
INSERT INTO '+ @name_audupd +' (id_AI, column_name, old_value,
new_value, id_rec)
SELECT @headerid, @Value1,
(select change_help_ID from my_deleted), (select change_help_ID from
my_inserted),
(select prim_help_ID from my_inserted);
END
EXEC sp_rename "my_inserted.change_help_ID", @Value1, "COLUMN";
EXEC sp_rename "my_deleted.change_help_ID", @Value1, "COLUMN";
END
EXEC sp_rename "my_inserted.prim_help_ID", @c_n, "COLUMN";
FETCH NEXT FROM curName1 INTO @Value1
END
CLOSE curName1
DEALLOCATE curName1
END
IF (@o_type = "Delete") or (@o_type = "Insert")
BEGIN
select * into tmp from my_inserted
UNION ALL
SELECT * from my_deleted;
ALTER TABLE tmp add id_AI INT;
UPDATE tmp SET id_AI = @headerid;
INSERT INTO' + @name_audinsdel +
' SELECT *
FROM tmp;
DROP TABLE tmp;
```

```
END
DROP TABLE my_inserted
DROP TABLE my_deleted
END'
exec(@SQL);
update MainAudInf set isAud = 1 where Tab_NAME = @name_Table;
GO
```

Додаток В

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

ІХ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

8–9 грудня 2021 року

ТЕРНОПІЛЬ

2021

УДК004.056

В. М. Дрозд, О. М. Сміх

Тернопільський національний технічний університет ім. І. Пулюя

МЕТОДИ ТА ЗАСОБИ ПРОВЕДЕННЯ АУДИТИ СИСТЕМИ УПРАВЛІННЯ БАЗАМИ ДАНИХ

V. M. Drozd, O. M. Smikh

METHODS AND MEANS OF CONDUCTING AUDIT OF DATABASE MANAGEMENT SYSTEM

У 2016 році в США, федеральне бюро розслідувань заявило, що виявило серйозні проблеми з безпеку системи реєстрації виборців. Невідомі намагалися “взламати” базу даних національного комітету демократичної партії США. Як результат під час передвиборчої кампанії в 2016 році, документи, які призначалися лиш для внутрішнього використання, були опубліковані у відкритому доступі.

Аудит баз даних – це комплекс дій, результатом яких є оцінка двох параметрів які визначають ефективність інформаційного масиву, а саме раціональність і безпека. З точки зору оптимізації процесів, прямо або опосередковано впливаючих на коефіцієнт корисної дії бази даних, аудит інформаційного масиву повинен бути відправною точкою.

На рисунку 1 зображено Oracle Audit Vault – сукупність засобів для запобігання невірних дій при роботі з даними, виявлення і блокування потенційних вразливостей, а саме: налаштування нотифікацій, звітів з специфічними користувацькими запитами, можливістю конструювання нових звітів, впровадження доступів.

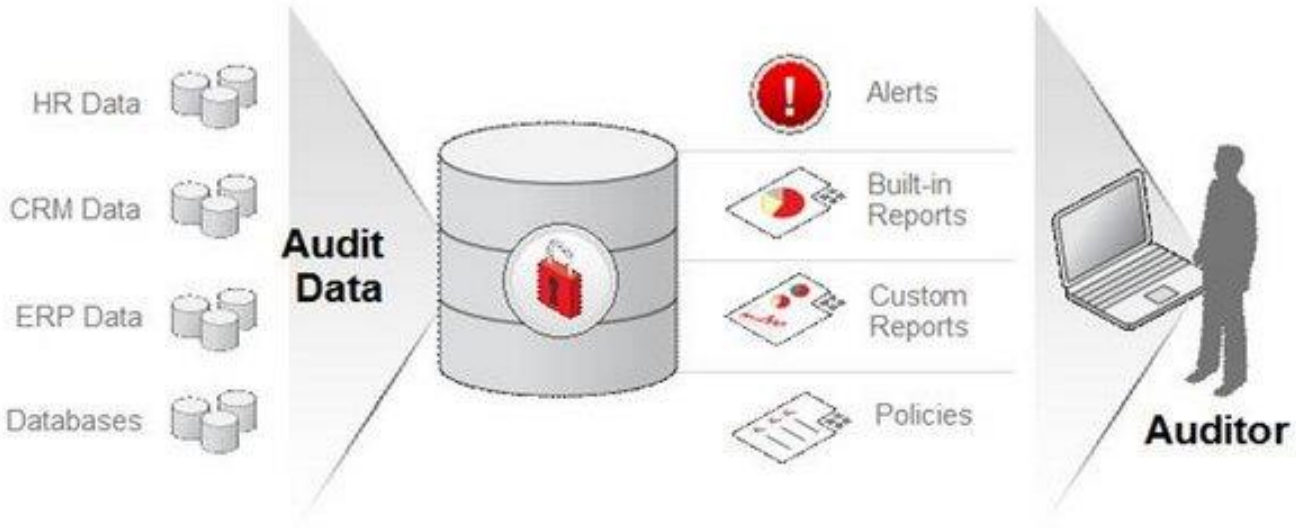

Рисунок 1 – Oracle Audit Vault

Oracle Audit Vault – це засіб для запобігання невірних дій при роботі з даними, виявлення і блокування потенційних вразливостей.

Для забезпечення захисту даних в мережах баз даних потрібно встановити певні правила, що можуть забезпечити комплексний захист. Бази даних - це основа для будь-якої комп’ютерної системи, що створює можливість для веб-ресурсів використовувати динамічний контент.

Аудит є невід’ємною частиною при проектуванні захищеної бази даних. Адже, дозволяє відслідковувати хто, що, коли і з якими даними робив. Важливо мати такі дані, оскільки може бути багато вразливостей пов’язаних безпосередньо із самою базою даних.

Отже, проведення аудиту є надзвичайно важливою частиною при проектуванні будь-якої бази даних. Завдяки аудиту можна позбутися великої кількості вразливостей, а також унеможливити виконання будь-яких дій без журналювання.