

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: “Розробка програмного забезпечення для автоматизації роздрібно
торгівлі”

Виконав: студент IV курсу, групи СН-41

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Ралік І.Р.

(прізвище та ініціали)

Керівник

(підпис)

Готович В.А.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Тернопіль
2021

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

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

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«__» _____ 2021 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Ралік Ігор Романович
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка програмного забезпечення для автоматизації роздрібної торгівлі

Керівник роботи Готович Володимир Анатолійович, к.т.н. ст.викл. кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «02» березня 2021 року № 4/7-171

2. Термін подання студентом завершеної роботи 22 червня 2021р.

3. Вихідні дані до роботи Літературні та інтернет джерела

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

Вступ. 1 Аналіз ринку роздрібної торгівлі та необхідних складових для його автоматизації
1.1 Аналіз ринку роздрібної торгівлі в Україні 1.2. Аналіз необхідних складових програмного
забезпечення для автоматизації роздрібної торгівлі 1.3 Вибір бази даних для програмного
забезпечення 1.4 Вибір програмного забезпечення та мови програмування для створення
1.5 Висновок до першого розділу 2 Реалізація програмного забезпечення для автоматизації
роздрібної торгівлі 2.1 Створення серверної частини 2.2 Створення довідника товарів 2.3
Створення довідника клієнтів 2.4 Створення реєстру прихідних та розхідних документів 2.5
Створення звіту про реалізацію товарів 2.6 Реалізовані методи класів С# 2.7 Висновок до
другого розділу 3. Безпека життєдіяльності, основи хорони праці. Висновки. Перелік
джерел. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Слайд 1. Титульна сторінка. Слайд 2. Актуальність теми роботи. Слайд 3. Мета та завдання
дослідження. Слайд 4. Ринок роздрібної торгівлі. Слайд 5. Зв'язки між таблицями Слайд 6.
Архітектура «Клієнт-сервер». Слайд 7. Зв'язки між таблицями в SQL. Слайд 8. Довідник
Товарів. Слайд 9. Функції довідника товарів. Слайд 10. Довідник клієнтів. Слайд 11. Реєстр
прихідних документів. Слайд 12. Вікно поточних залишків. Слайд 13. Звіт про реалізацію
Товару. Слайд 14. Видгляд стартового меню програми. Слайд 15. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Гурик О.Я., доцент кафедри МТ		

7. Дата видачі завдання 25 січня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	26.01.2021	<i>Виконано</i>
2.	Підбір джерел про автоматизацію роздрібної торгівлі	26.01.2021-29.01.2021	<i>Виконано</i>
3.	Переклад та опрацювання джерел про бази даних для програмного забезпечення	30.01.2021-05.02.2021	<i>Виконано</i>
4.	Виконання дослідження щодо ринку роздрібної торгівлі	06.02.2021-09.02.2021	<i>Виконано</i>
5	Розроблення програмного забезпечення для автоматизації роздрібної торгівлі	10.02.2021-19.02.2021	<i>Виконано</i>
5.	Оформлення розділу «Аналіз ринку роздрібної торгівлі та необхідних складових для автоматизації»	20.02.2021-24.02.2021	<i>Виконано</i>
6.	Оформлення розділу «Реалізація програмного забезпечення для автоматизації роздрібної торгівлі»	07.06.2021-08.06.2021	<i>Виконано</i>
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	09.06.2021-10.06.2021	<i>Виконано</i>
8.	Виконання завдання до підрозділу «Основи охорони праці»	10.06.2021-12.06.2021	<i>Виконано</i>
9.	Оформлення кваліфікаційної роботи	19.06.2021	<i>Виконано</i>
10.	Нормоконтроль	20.06.2021	<i>Виконано</i>
11.	Перевірка на плагіат	20.06.2021	<i>Виконано</i>
12.	Попередній захист кваліфікаційної роботи	20.06.2021	<i>Виконано</i>
13.	Захист кваліфікаційної роботи	22.06.2021	

Студент

_____ (підпис)

Ралік І.Р.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Готович В.А.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Розробка програмного забезпечення для автоматизації роздрібної торгівлі // Кваліфікаційна робота освітнього рівня «Бакалавр» // Ралік Ігор Романович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-41 // Тернопіль, 2021 // С. 53 , рис. – 13, табл. – 0, кресл. – 0, додат. – 2, бібліогр. – 30.

Ключові слова: база даних, облік, C#, програмне забезпечення, метод, SQL, автоматизація.

Кваліфікаційна робота присвячена розробці програмного забезпечення для автоматизації роздрібної торгівлі.

Мета роботи: розробка програмне забезпечення, яке дозволить вести облік товару в магазині будь-якого типу.

В першому розділі кваліфікаційної роботи розглянуто та проаналізовано ринок роздрібної торгівлі в Україні та необхідність даного типу програмного забезпечення, а також було проаналізовано та вибрано необхідні функції для програми.

В другому розділі кваліфікаційної роботи розглянуто практичну реалізацію програмного забезпечення для автоматизації роздрібної торгівлі.

ANNOTATION

Software development for automated retailing// Qualification work of the educational level "Bachelor" // Ralik Ihor Romanovich // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, group SN -41 // Ternopil, 2021 // P. 53, pic. 13, table. 0, draw. 0, appendix. 2, bibliogr. 30.

Keywords: database, accounting, C #, software, method, SQL, automation.

Qualification work dedicated to software development for automated retailing.

The goal of the work: development of software that will keep records of goods in the store of any type.

The first section of the qualification work developed and analyzed the retail market in Ukraine and the need for this type of software, as well as analyzed and selected the necessary functions for the programs.

In another section of the qualification work the practical implementation of software development of automated retailing.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – Програмне забезпечення.

SQL (англ. Structured query language – мова структурованих запитів) – декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних.

СУБД – система управління базами даних, комплекс програмних і мовних засобів, необхідних для створення баз даних, підтримання їх в актуальному стані та організації пошуку в них необхідної інформації.

ПК – Персональний комп'ютер.

C++, C#, Java – об'єктно-орієнтовані мови програмування.

Microsoft Visual Studio – середовище для створення програмного забезпечення.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ РИНКУ РОЗДРІБНОЇ ТОРГІВЛІ ТА НЕОБХІДНИХ СКЛАДОВИХ ДЛЯ АВТОМАТИЗАЦІЇ.....	8
1.1 Аналіз ринку роздрібної торгівлі в Україні.....	8
1.2 Аналіз необхідних складових програмного забезпечення для автоматизації роздрібної торгівлі	10
1.3 Вибір бази даних для програмного забезпечення.....	11
1.4 Вибір програмного забезпечення та мови програмування для створення	14
1.5 Висновок до першого розділу	15
РОЗДІЛ 2. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ РОЗДРІБНОЇ ТОРГІВЛІ	17
2.1 Створення серверної частини.....	17
2.2 Створення довідника товарів	19
2.3 Створення довідника клієнтів	22
2.4 Створення реєстру прихідних та розхідних документів	24
2.5 Створення звіту про реалізацію товарів.....	30
2.6 Реалізовані методи класів C#	31
2.7 Висновок до другого розділу	34
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ	35
3.1 Працездатність людини-оператора.....	35
3.2 Загальні вимоги безпеки з охорони праці для користувачів ПК.....	37
3.3 Висновок до третього розділу	39
ВИСНОВКИ.....	40
ПЕРЕЛІК ДЖЕРЕЛ	41
ДОДАТКИ	

ВСТУП

Актуальність теми. З кожним днем сучасний світ стає все більш комерціалізований, з'являється багато нових компаній в сферах продажу товарів. Список товарів компанії може складати від кількох до тисячі позицій, через це виникає потреба в автоматизації обліку товару. Для цього в даній кваліфікаційній роботі було створено програмне забезпечення, яке дозволить максимально автоматизувати облік для компаній.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є:

- Проаналізувати необхідність програмного забезпечення для автоматизації роздрібної торгівлі.
- Проаналізувати технології, які найкраще підійдуть для даного типу ПЗ.
- Створити програмне забезпечення, яке допоможе компаніям витратити менше часу на облік товарів.

Практичне значення одержаних результатів. На практиці отриманий результат спрощує компанії, яка користуватиметься даним програмним забезпеченням, весь облік та аудит товарів. Таким чином компанія буде мати весь довідник товарів, довідник постачальників, ціни на товари та інше у електронному варіанті, матиме змогу вводити прихідні накладні товарів та спостерігати за залишками на складі.

РОЗДІЛ 1. АНАЛІЗ РИНКУ РОЗДРІБНОЇ ТОРГІВЛІ ТА НЕОБХІДНИХ СКЛАДОВИХ ДЛЯ АВТОМАТИЗАЦІЇ

1.1 Аналіз ринку роздрібноЇ торгівлі в Україні

У теперішній ринковій економіці України комерційна організація функціонує в жорстоких умовах конкурентного середовища, особливо в сфері продажу, тому компанії повинні докладати максимум зусиль для втримання своїх позицій. Організацію бухгалтерського обліку необхідно побудувати таким чином, щоб обліково–аналітична інформація могла бути отримана за будь-який проміжок часу за будь-яким напрямом управління продажами товарів [1]. Для цього потрібне програмне забезпечення, яке дасть можливість створювати звіт по виконаних операціях.

У торгівлі, як у жодній іншій галузі господарської системи, сформувалося багате конкурентне середовище. Підприємницька й інвестиційна активність у цій сфері найвища.

Якщо складський облік товарів організований по партіях протягом звітного періоду товар здобувався по однієї і тій же ціні, то базою для визначення покупної вартості реалізованих товарів можуть служити дані первинних документів. При цьому досить кількість реалізованих товарів у натуральному вираженні помножити на ціну їхнього придбання. Складності виникають при придбанні товарів у різних постачальників за різними цінами. У цих випадках для визначення покупної вартості реалізованих товарів можуть бути використані методи оцінки матеріальних запасів:

- по середньозваженій ціні;
- за цінами перших закупівель;
- за цінами останніх закупівель;
- по собівартості одиниці запасів.

В даній кваліфікаційній роботі було обрано продаж товару зі складу – за цінами перших закупівель.

Сучасний споживчий ринок відрізняється високою насиченістю. Товарний дефіцит практично відсутній. Бюрократична система розподілу товарів цілком замінена відносинами вільної купівлі-продажу [2].

Однією із самих складних задач для компанії, що займається торговою діяльністю, є точний і упорядкований облік матеріальних засобів. При великому обороті первинних документів стає дуже складним їх упорядкування. Як правило, багато компаній дотепер, при такому стрімкому розвитку комп'ютерних технологій і програмного забезпечення, не мають чітко налагодженого обліку. Програмні продукти покликані вирішувати дані проблеми та автоматизувувати всю систему продажу.

Господарський облік являє собою сукупність бухгалтерського та статистичного обліку. Облік – одна з найбільш трудомістких функцій керування. Відмітною рисою обліку є велика масовість і однорідність вихідних і підсумкових показників. Як правило, підсумкові показники формуються шляхом багаторазового угруповання по різних ознаках вихідних первинних даних без застосування складних розрахунків [3–4].

Важливим елементом обліку реалізації товарів є визначення вартості, по якій вони списуються з балансу торгової організації.

Є багато програм, які використовуються для автоматизації в різних сферах, використовують від звичайних таблиць в Microsoft Excel до програм від популярних виробників [5]. При використанні Excel функціонал є доволі обмежений, а програмне забезпечення від популярних виробників є досить дорогим задоволенням, де не тільки потрібно придбати програмне забезпечення, але й платити за ліцензію на певний термін. Тому найкращим варіантом є створення власного програмного забезпечення у Microsoft Visual Studio, який дозволить не тільки створити потрібний функціонал, але й постійно його розширювати, додаючи, наприклад, сканер штрих кодів та інші функції.

1.2 Аналіз необхідних складових програмного забезпечення для автоматизації роздрібно́ї торгівлі

Для того, щоб якісно створити програмне забезпечення, потрібно добре проаналізувати дану сферу, в нашому випадку це сфера роздрібно́ї торгівлі. В роздрібно́ї торгівлі є безліч компаній, кожна з яких продає різні товари різних типів, але всі товари мають майже ідентичні характеристики. Після детального вивчення даної сфери перейдемо до створення довідника товарів [6–7].

Довідник товарів – це таблиця формування картки товару з вказаними характеристиками на товар. Основні функціональні можливості: створення нової картки, внесення нової ціни на товар, внесення змін у вже існуючу картку товару [8–9].

З аналогічним функціоналом побудована таблиця довідника партнерів. На відміну від всіх інших таблиць дані не містять знищення товару чи партнера.

Оскільки дані елементи програми задіяні у формуванні первинних документів, на основі якого формується аналіз руху товару і всі інші операції, тому знищення таких документів не допустиме. Користувач, набираючи приходну накладну, може допуститися помилок, які потрібно завчасно виправити. Залишки товарів на вказану дату формуються на основі розрахункових операцій: Прихід товару - розхід товару + Сума залишку на початок вказаного періоду. Також завжди є необхідність оперативно і точно розрахувати реалізацію товарів і відповідно до залишку сформувані замовлення для постачальника для вказаного періоду реалізації. Саме ця функція збереже компанію від помилок при заповненні приходних накладних, що є важливим, особливо при роботі з величезними обсягами даних.

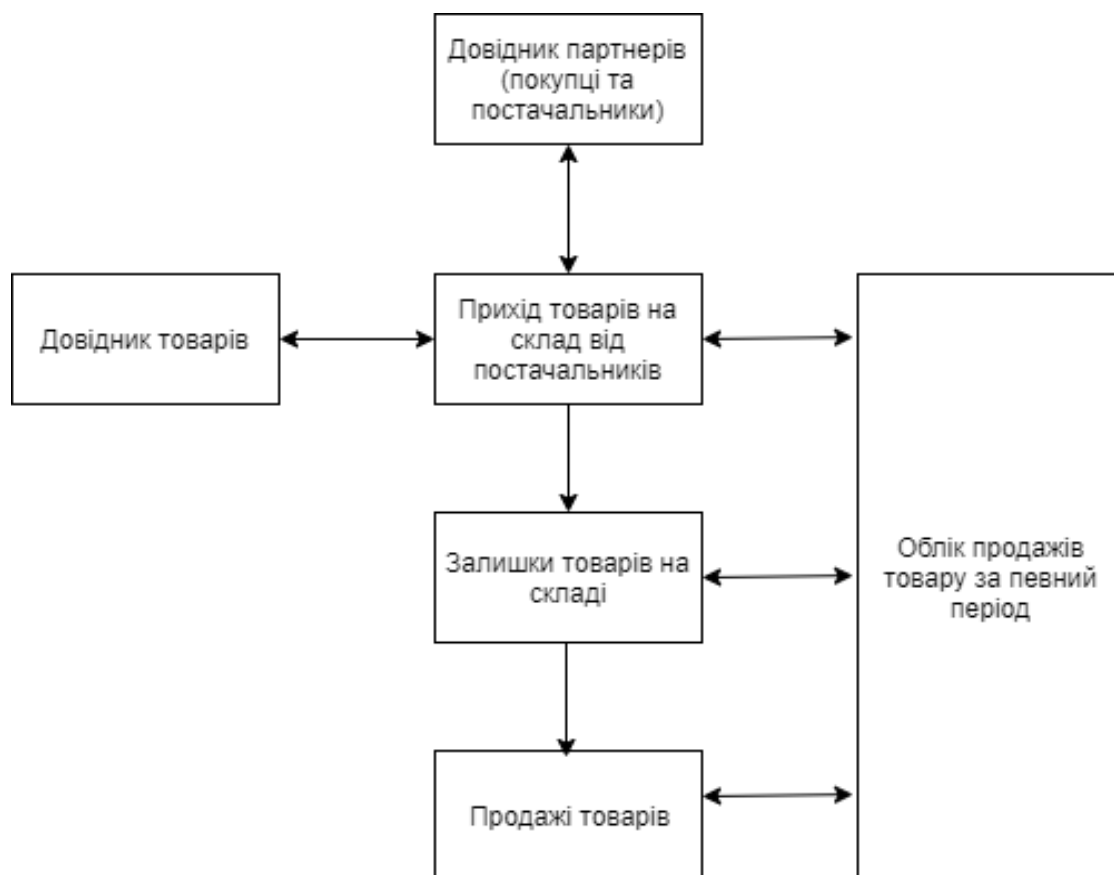


Рисунок 1.1 – Формування зв'язків між таблицями

На рисунку 1.1 зображено сформовані зв'язки між таблицями, їх формування є важливим перед створенням програмного забезпечення, так як потрібно чітко розуміти, між якими таблицями буде відбуватись обмін даними та які параметри вони будуть передавати.

1.3 Вибір бази даних для програмного забезпечення

Для початку розберемо, що ж таке база даних, та як їх класифікують, щоб вибрати найкращий варіант, який підійде для даної кваліфікаційної роботи. База даних – це упорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система управління БД [10].

Головним завданням БД є гарантоване збереження значних обсягів інформації та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин: збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних) [11–12].

Існує велика кількість різновидів баз даних, що відрізняються за критеріями. В енциклопедії технологій баз даних визначаються понад п'ятдесят видів БД [13].

Класифікація БД за моделлю даних:

- ієрархічні;
- мережеві;
- реляційні;
- об'єктні;
- об'єктно-орієнтовані;
- об'єктно-реляційні.

Класифікація БД за технологією фізичного зберігання:

- БД у вторинній пам'яті (традиційні);
- БД в оперативній пам'яті (in-memory databases);
- БД у третинній пам'яті (tertiary databases).

Класифікація БД за вмістом:

- географічні;
- історичні;
- наукові;
- мультимедійні.

Класифікація БД за ступенем розподіленості:

- централізовані (зосереджені);
- розподілені.

Для централізованих баз даних з мережевим доступом припускають такі архітектури систем:

- файл-сервер;

– клієнт-сервер.

Розглянемо детальніше архітектури баз даних. Архітектура “Файл-сервер” – архітектура систем БД з мережевим доступом передбачає виділення однієї з машин мережі в якості центральної (сервера). На такій машині зберігається спільно використовувана централізована база даних. Усі інші машини мережі виконують функції робочих станцій, за допомогою яких підтримується доступ користувальницької системи до централізованої бази даних [14–15]. Файли бази даних відповідно до призначених для користувача запитів передаються на робочі станції, де в основному і проводиться обробка. При великій інтенсивності доступу до одних і тих же даних продуктивність інформаційної системи падає [16].

Архітектура “Клієнт-сервер” – у цій концепції мається на увазі, що крім зберігання централізованої бази даних сервер бази даних повинен забезпечувати виконання основного обсягу обробки даних. Запит на дані, який видається клієнтом, породжує пошук і видалення даних на сервері. Витягнуті дані транспортуються по мережі від сервера до клієнта. Специфікою архітектури клієнт-сервер є використання мови запитів SQL [17].

Вивчивши, які бази даних бувають, для даної роботи було обрано мову запитів SQL. Основні переваги мови SQL, які і стали причиною для вибору мною саме цієї мови, полягають в наступному [18]:

- стандартизованість – використання мови SQL в програмах стандартизовано міжнародними організаціями;
- незалежність від конкретних СУБД – реляційну базу даних можна перенести з однієї СУБД на іншу;
- таблична структура реляційної БД добре зрозуміла, а тому мова SQL є простою для вивчення;
- можливість створення інтерактивних запитів – SQL забезпечує користувачам доступ до даних без написання програми;
- можливість програмного доступу до БД – мову SQL легко використовувати в додатках, яким необхідно звертатися до баз даних;

– забезпечення різного представлення даних – дані з різних частин БД можуть бути скомбіновані і представлені у вигляді однієї простої таблиці.

1.4 Вибір програмного забезпечення та мови програмування для створення

Для написання програмного забезпечення потрібно вибрати програмне забезпечення для його створення та обрати мову програмування. Обирати потрібно залежно від того, яке програмне забезпечення нам потрібно створити. Цей вибір є важливим, оскільки від цього залежить наскільки функціональним ми зможемо зробити наш продукт. Якщо ви створюєте веб-сайт, під ці цілі найкраще підійде PHP чи Java, в нашому ж випадку нам потрібно створити повноцінну програму, яка буде з'єднуватись з базою даних [19–20]. Для створення повноцінної програми підійде Microsoft Visual Studio з використанням мови C#, так як вона найкраще підходить під даний тип програмного забезпечення, та дозволить нам розширювати функціонал за допомогою підтримки безлічі плагінів. Розглянемо переваги вибору Microsoft Visual Studio [21–22].

За допомогою Visual Studio користувачам надається допомога в кодуванні в реальному часі незалежно від мови програмування, яку вони використовують. Вбудований в платформу IntelliSense пропонує підказки, описи API та автоматично заповнює рядки для кращої швидкості написання коду [23].

Пошук та діагностика помилок інколи може бути складним завданням, але Visual Studio має безліч інструментів, що полегшують його. Платформа має підтримку налагодження для всіх включених мов [24]. Процес також може виконуватися локально, віддалено та в середині виробництва. Це дозволяє розробникам розгортати програми на робочому столі або емулятори на мобільних пристроях та інші методи налагодження.

Microsoft Visual Studio оснащений платформою для тестування програм, яка дозволяє розробникам переконатися, що вони готові до запуску

програмного продукту. Вони можуть зробити це на бажаній мові та в тестовій системі, роблячи це з обмеженими зусиллями. Таким чином, вони можуть направити більшу частину своєї енергії на інші фази створення [25].

Visual Studio пропонує можливість налаштування програмного застосунку для кожного користувача. Також можна розширити функціональні можливості платформи за допомогою розширень та доповнень, доступних на Visual Studio Marketplace [26]. Розробники навіть можуть публікувати власні розширення.

Також розглянемо вибір мови програмування. Для наших цілей найкращим вибором буде мова С#. У С# є кілька переваг, саме тому це така поширена мова в більшості проектів розробки [27].

По-перше, він добре інтегрується з Windows. Не потрібні особливі конфігурації, щоб програма С# працювала у середовищі Windows. А також, якщо у вас є команда розробників, Microsoft полегшила команді спільну роботу над одним проектом. Visual Studio має інтегровані інструменти для контролю змін та злиття коду. Корпорація Майкрософт пропонує версії свого програмного забезпечення та корпоративні опції для великих команд розробників [28].

С# – це компільована мова, що означає, що код, що зберігається на загальнодоступному сервері, є у двійковій формі. Якщо ваш сервер буде зламаний, хакер автоматично не матиме доступу до вашого вихідного коду. З іншими поширеними мовами, такими як PHP, хакер отримує доступ до вихідного коду, який потім може надати йому доступ до паролів бази даних. За допомогою С# хакер повинен декомпілювати або зламати ваше програмне забезпечення, перш ніж він зможе побачити компоненти [29].

1.5 Висновок до першого розділу

В першому розділі кваліфікаційної роботи для програмної реалізації ПЗ для автоматизації роздрібної торгівлі були обрані мови SQL і С#, і відповідно,

програмне забезпечення Microsoft SQL Server 2012 та Microsoft Visual Studio 2017. Розглянута непроцедурна мова SQL (Structured Query Language – структурована мова запитів) орієнтована на операції з даними, представленими у вигляді логічно взаємопов'язаних сукупностей таблиць. Особливість цієї мови полягає в тому, що вони орієнтовані більшою мірою на кінцевий результат обробки даних, ніж на процедуру цієї обробки. SQL сам визначає, де знаходяться дані, які індекси і навіть які найбільш ефективні послідовності операцій слід використовувати для їх отримання. Крім того, мова роботи з базами даних повинна вирішувати всі зазначені вище завдання при мінімальних зусиллях з боку користувача, а структура і синтаксис його команд - досить прості і доступні для вивчення. Але при всіх перерахованих вище плюсах, структурована мова запитів SQL не має достатніх ресурсів для створення складних прикладних програм. Тому в різних СУБД вона використовується разом з мовами програмування високого рівня. Для реалізації проекту база даних, будемо використовувати додатки, що працюють в середовищі .NET Framework. Завдяки безлічі нововведень C# забезпечує можливість швидкої розробки додатків. Процес побудови C# в порівнянні з C++ простий і є більш гнучким, ніж в Java. Немає окремих файлів заголовка, а методи і типи не потрібно оголошувати в певному порядку. Також з переваг, це те, що у вихідному файлі C# може бути визначено будь-яке число класів, структур, інтерфейсів та подій.

РОЗДІЛ 2. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЗАЦІЇ РОЗДРІБНОЇ ТОРГІВЛІ

2.1 Створення серверної частини

Для побудови програмного забезпечення використано технологію “клієнт-сервер”, детальніше про цю технологію було описано у першому розділі. Дана технологія стосовно до СУБД зводиться до поділу системи на дві частини - додаток-клієнт (front-end) і сервер бази даних (back-end). Як основний інтерфейс між клієнтською й серверною частинами виступає мова баз даних SQL. Клієнтська частина створена засобами об’єктно-орієнтованої мови програмування C#. В клієнтській частині зберігаються форми, запити та все інше, що стосується безпосередньої роботи програми. MS SQL Server – це архітектура клієнт-сервер. Процес MS SQL Server починається з того, що клієнтська програма надсилає запит [30]. SQL Server приймає, обробляє та відповідає на запит обробленими даними. База даних в даній роботі містить один основний файл. Там зберігаються всі важливі дані, що стосуються таблиць, подань, тригерів тощо. Схема роботи клієнт-серверної архітектури в даній кваліфікаційній роботі зображена на рисунку 2.1.



Рисунок 2.1 – Схема роботи клієнт-серверної архітектури

В даній кваліфікаційній роботі, в основному використовувався клас Windows Form. Також було використано бібліотеки DevExpress, для кращої кастомізації вікон з таблицями.

В даному проєкті присутні класи з використанням таблиць (ProductsZal, Products, Clients) для створення яких було використано GridControle та LayoutControl для полегшення роботи з таблицями.

Дана програма написана з використанням мов програмування C# та msSQL. Програму можна було описати і без використання msSQL, описавши таблиці в самій програмі, але було вирішено використати саме цю базу даних, оскільки це спрощує роботу на стороні клієнта та дозволяє зберігати доволі великі об'єми інформації на стороні серверу. Для цієї взаємодії було використано клас sqlDataAdapter, який служить свого роду мостом між SQL та C#. За допомогою методів даного класу, вдалось реалізувати роботу між програмою та базою даних. Також в програмі виконано функціонал додавання, редагування та знищення даних в таблицях, які змінюються в режимі реального часу. За допомогою методів та алгоритмів даної програми, може функціонувати невеликий магазин.

При запуску програми, в лівому верхньому куті присутнє меню з навігацією – вікна довідників, довідник клієнтів та довідник товарів. Кожне вікно добавляння нового товару чи нового клієнта – описане окремо. Ці довідники взаємодіють з базами даних на стороні сервера, і до них прив'язані певні процедури.

Серверна частина описана в 4 таблицях та 14 процедурах, в тому числі , деякі таблиці для клієнтської частини було створено віртуально за допомогою Select, Left Join, Inner Join. Зв'язки між таблицями наведені на рисунку 2.2.

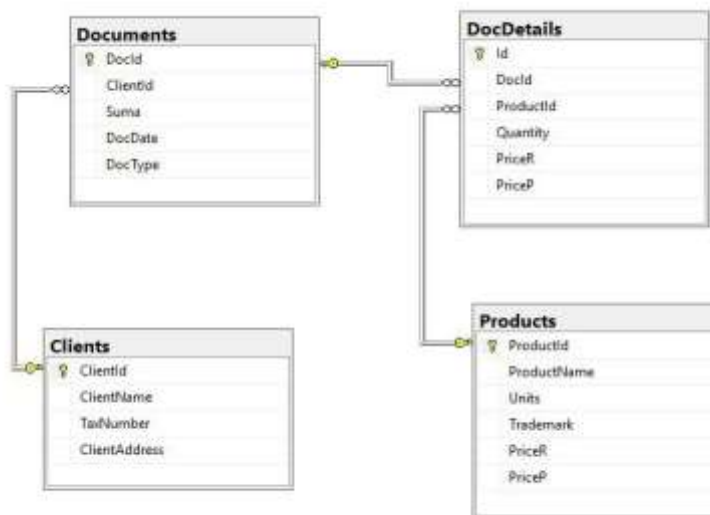


Рисунок 2.2 – Зв'язки між таблицями

В таблицях були задані зв'язки зображені на рисунку 2.2, попередньо ці зв'язки були сформовані у першому розділі роботи на рисунку 1.1. Тепер ж при програмній реалізації було чітко задано, якими параметрами будуть обмінюватись таблиці. Були також задані головні ідентифікатори для кожної з таблиць.

2.2 Створення довідника товарів

Довідник товарів – реєстр даних в якому накопичуються картки товару з вказаними характеристиками товару. Даний довідник містить такі функції: створення нової картки товару, внесення змін у вже існуючу картку товару та знищення картки товару.

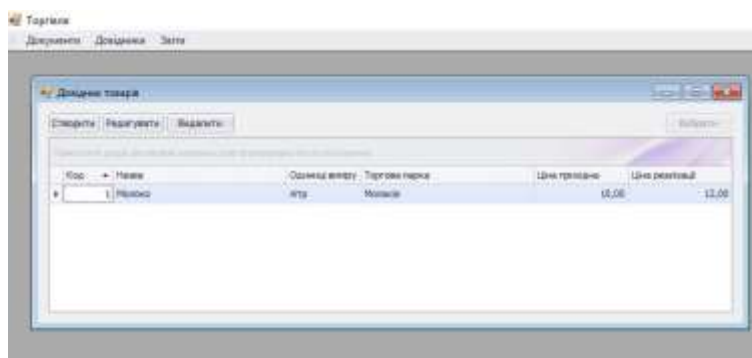
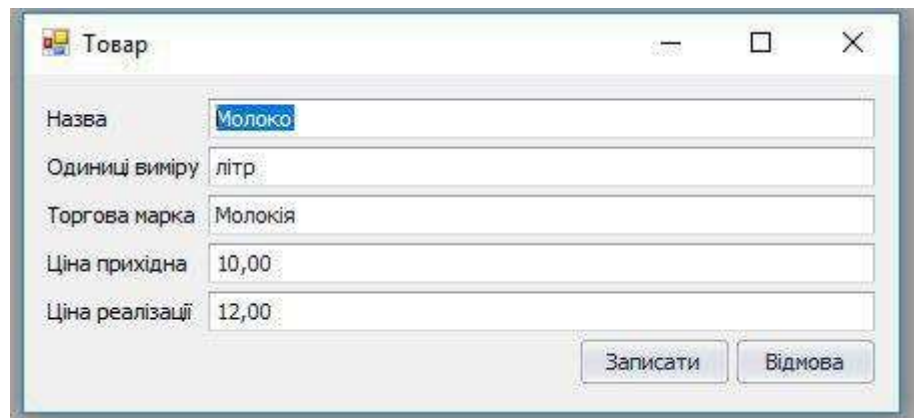


Рисунок 2.3 - Вікно довідника товарів



Назва	Молоко
Одиниці виміру	літр
Торгова марка	Молокія
Ціна придна	10,00
Ціна реалізації	12,00

Рисунок 2.4 – Картка товару

В даному вікні на рисунку 2.4 ми можемо вказати назву товару, одиниці виміру, торгову марку, придну ціну та ціну реалізації товару.

Для роботи з даними таблиці “Products” було написано 3 збережені процедури:

- [dbo].[ProductsUpdate] – реалізована функція додавання Insert та оновлення Update даних;
- [dbo].[ProductsSelectAll] – реалізована функція відбору даних Select;
- [dbo].[ProductsDelete] – реалізована функція знищення Delete.

Функція Insert описана в процедурі Update, і виконана функцією If. При отриманні значення $id = 0$ – в нас створюється новий запис, при отриманні певного значення, наприклад $= 1$ – функція Update дозволить нам відредагувати вже доступні значення в таблицях. Код даної частини програми наведено у лістингу 2.1.

Лістинг 2.1 – Функція додавання та оновлення даних у таблиці продуктів

GO

```
CREATE PROCEDURE [dbo].[ProductsUpdate]
@ProductId int output, @ProductName varchar(100), @Units
varchar(20), @Trademark varchar(100),
@PriceR numeric (9,2), @PriceP numeric (9,2)
```

AS

BEGIN

```

        if @ProductId = 0 -- Створення нового товару
            begin
                insert into Products (ProductName, Units, Trademark,
PriceR, PriceP)
                    values      (@ProductName,    @Units,    @Trademark,
@PriceR, @PriceP)
                select @ProductId = SCOPE_IDENTITY()
            end
        else
            begin
                update Products set ProductName = @ProductName,
Units = @Units, Trademark = @Trademark,
                    PriceR = @PriceR, PriceP = @PriceP
                where ProductId = @ProductId
            end
    end

END

```

Функція знищення працює таким чином: по ідентифікаційному номеру товару у базі даних знаходиться потрібний запис та видаляється. Лістинг процедури видалення зображений на лістингу 2.2.

Лістинг 2.2 – Функція видалення даних з таблиці продуктів

```

GO

create PROCEDURE [dbo].[ProductsDelete]
@ProductId int

AS
BEGIN
    delete from Products where ProductId = @ProductId
END

```

Функція `SelectAll` відбирає необхідні нам дані з бази даних по назві продукту.

Лістинг 2.3 – Функція відбору даних з таблиці продуктів

```

GO

create PROCEDURE [dbo].[ProductsSelectAll]

AS
BEGIN
    select * from Products order by ProductName

```

END

Загалом для довідника товарів було написано 3 збережені процедури.

2.3 Створення довідника клієнтів

Облік продажу товару не може супроводжуватися без вказаного клієнта (постачальника). Саме тому для програми була створена таблиця “Довідник клієнтів”.

Таблиця “Довідник Клієнтів” необхідна для накопичення карток постачальників з їхніми даними такими як: внутрішній код клієнта, назва клієнта, його адреса, а також податковий номер.

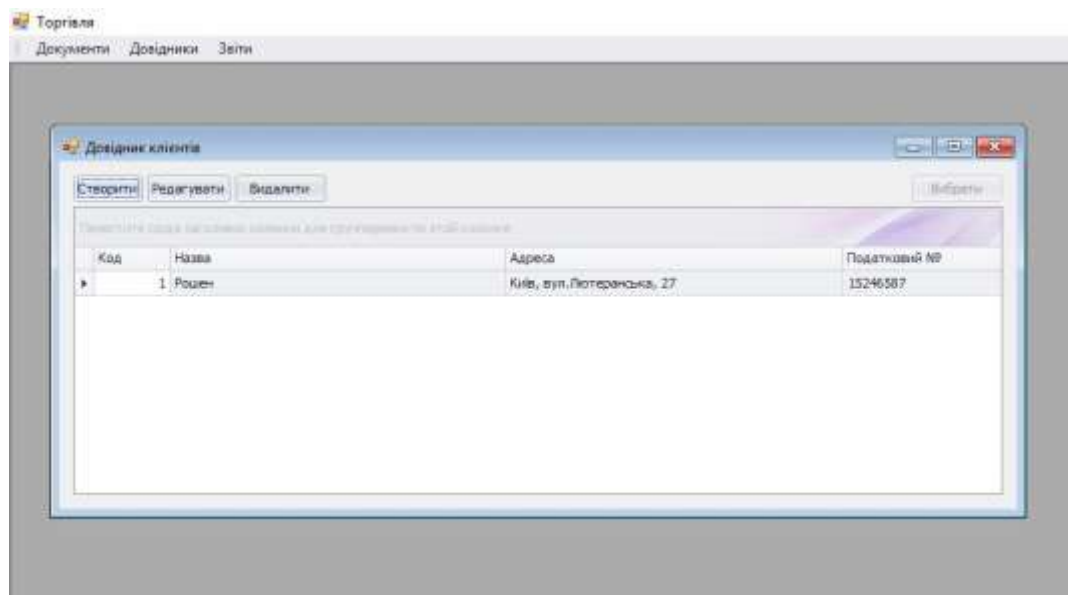


Рисунок 2.5 - Вікно довідника клієнтів

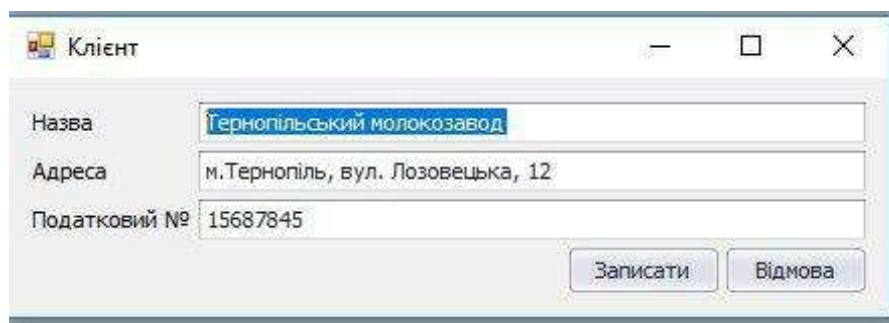


Рисунок 2.6 – Вікно картки клієнта

Основними функціями вікно, зображеного на рисунках 2.5 та 2.6 є створення нової картки клієнта, внесення змін у вже створену картку та видалення картки клієнта.

Для роботи з даними таблиці “Clients” було написано 3 збережені процедури:

- [dbo].[ClientsUpdate] – реалізована функція додавання Insert та оновлення Update даних;
- [dbo].[ClientsSelectAll] – реалізована функція відбору даних Select;
- [dbo].[ClientsDelete] – реалізована функція знищення Delete.

Як і у попередньому довіднику функція Insert описана в процедурі Update та виконана функцією If.

Лістинг 2.4 – Функція додавання та оновлення даних у таблиці клієнтів

```
BEGIN
  if @ClientId = 0 -- Створення нового клієнта
    begin
      insert into Clients (ClientName, TaxNumber,
ClientAddress) values (@ClientName, @TaxNumber, @ClientAddress)
      select @ClientId = SCOPE_IDENTITY()
    end
  else
    begin
      update Clients set ClientName = @ClientName,
TaxNumber = @TaxNumber, ClientAddress = @ClientAddress
      where ClientId = @ClientId
    end
END
```

Функція знищення працює таким чином: по ідентифікаційному номеру клієнта у базі даних знаходиться потрібний запис та видаляється.

Лістинг 2.5 – Функція видалення даних з таблиці клієнтів

```
GO
CREATE PROCEDURE [dbo].[ClientsDelete]
@ClientId int
AS
BEGIN
```



```
delete from Clients where ClientId = @ClientId
END
```

Функція `SelectAll` відбирає необхідні нам дані з бази даних по назві клієнта.

Лістинг 2.6 – Функція відбору даних з таблиці клієнтів

```
GO

create PROCEDURE [dbo].[ClientsSelectAll]

AS
BEGIN
    select * from Clients order by ClientName
END
```

Так як і для довідника товарів для довідника клієнтів нам потрібно було створити 3 збережені процедури в SQL.

2.4 Створення реєстру прихідних та розхідних документів

В даній таблиці зберігаються реєстр прихідних та розхідних документів. В полі `DocType` цієї таблиці прописується тип документу 1 – прихідні документи, 2 – розхідні документи.

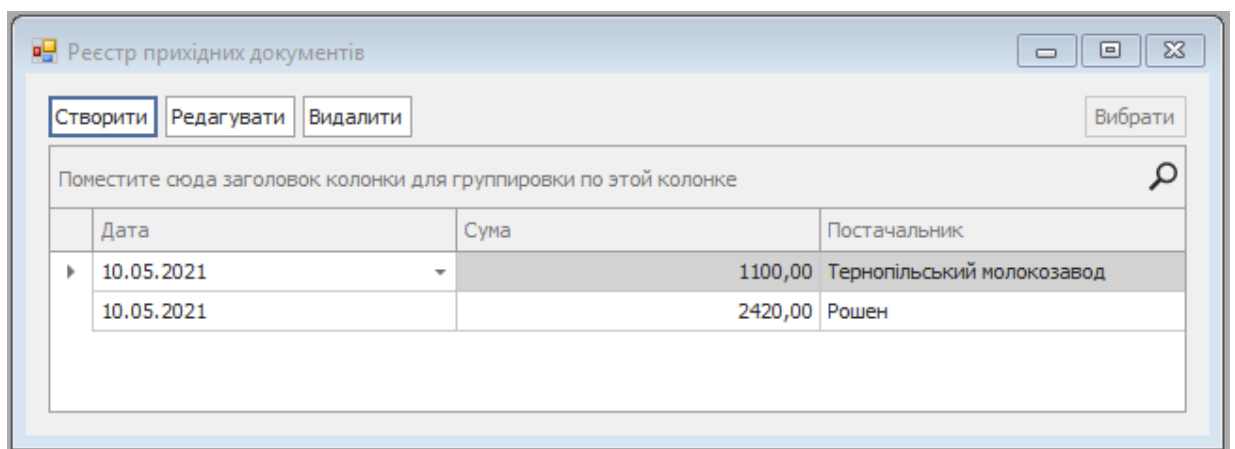


Рисунок 2.7 – Вікно реєстру прихідних документів

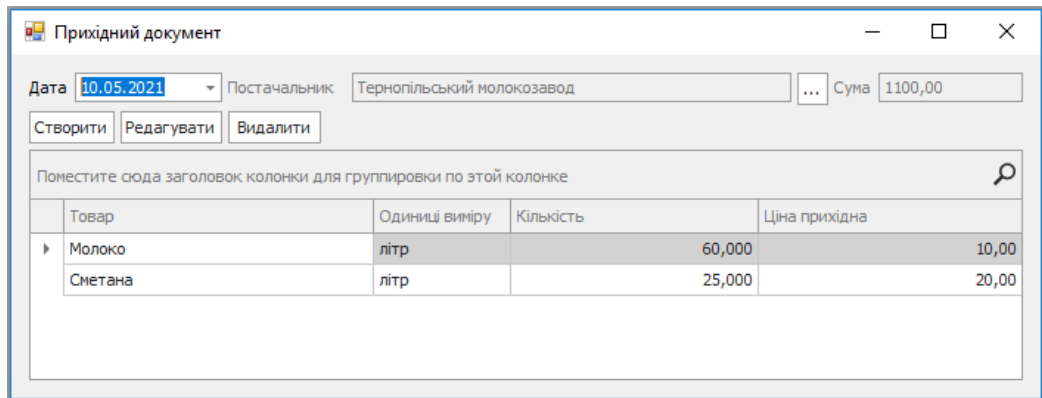


Рисунок 2.8 – Вікно прихідного документу

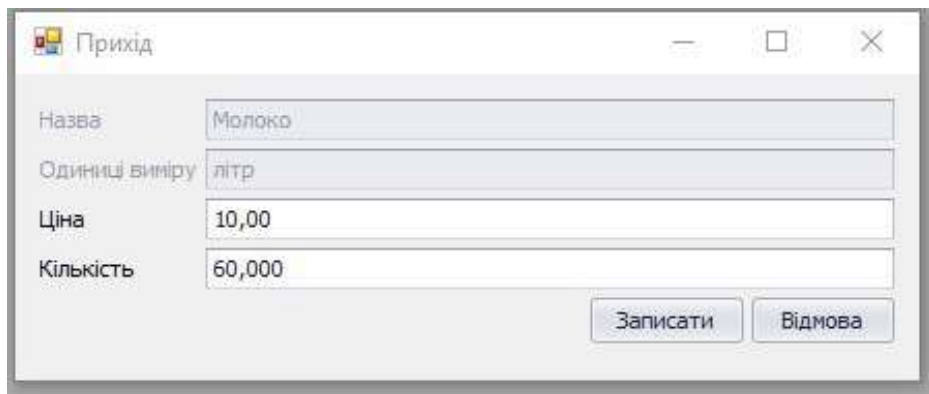


Рисунок 2.9 – Редагування позиції товару в прихідному документі

В даних вікнах зображених на рисунках 2.7 – 2.9 присутні таблиці та функціональні кнопки (створити документ, редагувати, видалити).

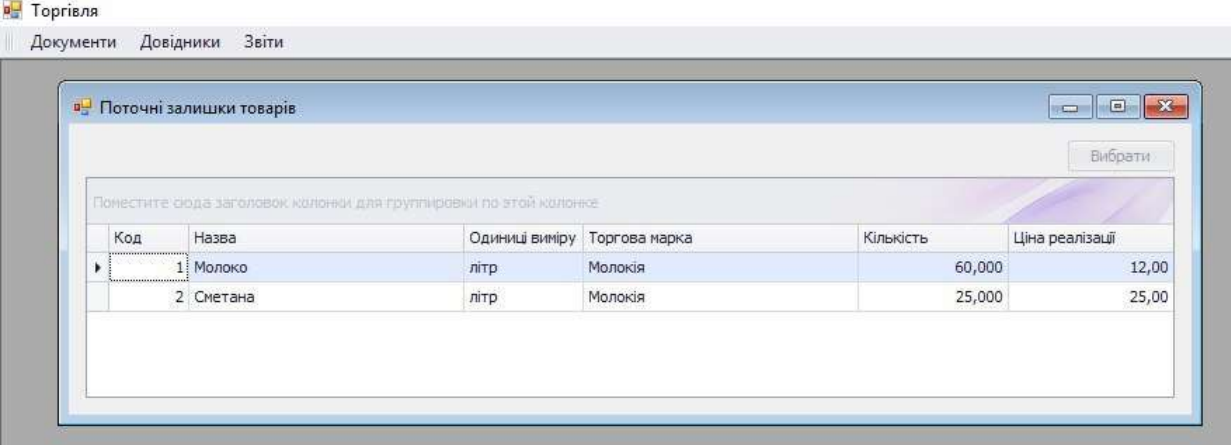
При створенні нового документу чи редагуванні поточного, відкривається вікно, в якому описано які саме товари, скільки, і по якій ціні було їх придбано чи продано – у випадку розхідного документу. При додаванні нового продукту до списку – появляється вікно “Довідник Товарів”, з якого ми можемо обрати потрібну продукцію. Натиснувши на кнопку “Вибрати” – ми потрапляєм в вікно редагування, у якому можна зазначити – скільки товару та за яку ціну було придбано.

При створенні нового об'єкта в “Журналі розхідних документів” – програма відкриває “Таблицю залишків”, щоб ми могли обрати товар, який на даний момент присутній на складі.

Формування прихідних накладних відбувається за такою послідовністю:

- Створення бланку накладної.
- Додавання одної або більше позицій товару по собівартості і з вказаною кількістю товару.
- Миттєвий розрахунок суми приходу товару.
- Вибрати постачальника з довідника партнерів, або зразу ж у даному вікні створити нову картку партнера.

Вікно “Поточні залишки товарів” – таблиця, яка генерується в реальному часі та показує скільки і якого товару залишилось в даний час на складі.



Код	Назва	Одиниці виміру	Торгова марка	Кількість	Ціна реалізації
1	Молоко	літр	Молюксія	60,000	12,00
2	Сметана	літр	Молюксія	25,000	25,00

Рисунок 2.10 – Вікно поточних залишків товару

Програма передбачає також можливість користувача спочатку здійснити продаж товару і тим самим утворити від’ємну кількість товару на залишках, а потім, при приході товару за допомогою прихідної накладної, кількість вирівнюється до фактичної кількості.

Для роботи з даними таблиці “Documents” було написано 3 збережені процедури:

- [dbo].[DocumentsUpdate] – реалізована функція додавання Insert та оновлення Update даних;
- [dbo].[DocumentsSelectAll] – реалізована функція відбору даних Select;
- [dbo].[DocumentsDelete] – реалізована функція знищення Delete.

Як і у попередніх таблицях функція Insert описана в процедурі Update, і виконана функцією If, так як це є найкращим варіантом для реалізації цієї

функції. При отриманні значення `id=0` – буде створюватися новий запис, при отриманні певного значення, наприклад, `id=1` – функція `Update` відредагує вже доступні значення в таблицях.

Лістинг 2.7 – Функція додавання та оновлення даних у таблиці документів

```
GO

CREATE PROCEDURE [dbo].[DocumentsUpdate]
@DocId int output, @ClientId int, @Suma numeric (18,2), @DocDate
datetime, @DocType smallint

AS
BEGIN
    if @DocId = 0 -- Створення нового документа
        begin
            insert into Documents (ClientId, Suma, DocDate,
DocType)
                values (@ClientId, @Suma, @DocDate, @DocType)
            select @DocId = SCOPE_IDENTITY()
        end
    else
        begin
            update Documents set ClientId = @ClientId, Suma =
@Suma, DocDate = @DocDate, DocType = @DocType
                where DocId = @DocId
        end
END
```

Функція знищення працює таким чином: по ідентифікаційному номеру документа у базі даних знаходиться потрібний запис та видаляється з двох таблиць: документів та прихідних та розхідних.

Лістинг 2.8 – Функція видалення даних з таблиці документів

```
GO

CREATE PROCEDURE [dbo].[DocumentsDelete]
@DocId int

AS
BEGIN
    delete from DocDetails where DocId = @DocId
    delete from Documents where DocId = @DocId
```

END

Функція відбору даних з таблиці документів є складнішою за інші, так як дані тут беруться вже з різних таблиць, та добавляються за допомогою left join.

Лістинг 2.9 – Функція відбору даних з таблиці документів

```
GO

CREATE PROCEDURE [dbo].[DocumentsSelectAll]
@DocType smallint

AS
BEGIN
    select t1.*, ClientName = isnull(t2.ClientName, '')
    from Documents t1 left join Clients t2 on t1.ClientId =
t2.ClientId
    where t1.DocType = @DocType
    order by t1.DocDate
END
```

Позицій прихідних та розхідних документів зберігаються в таблиці “DocDetails”. Для роботи з даними таблиці “DocDetails” було написано 3 збережені процедури:

- [dbo].[DocDetailsUpdate] – реалізована функція добавлення Insert та оновлення даних Update;
- [dbo].[DocDetailsSelectbyDocId] – реалізована функція відбору даних Select по ідентифікаційному номеру документу;
- [dbo].[DocDetailsDelete] – реалізована функція знищення Delete.

Реалізація функції добавляння та оновлення даних знаходиться у лістингу 2.10.

Лістинг 2.10 – Функція добавляння та оновлення даних у таблиці прихідних та розхідних документів

```
GO

CREATE PROCEDURE [dbo].[DocDetailsUpdate]
@Id int output, @DocId int, @ProductId int, @Quantity numeric
(18,3), @PriceR numeric (9,2), @PriceP numeric (9,2)
```

```

AS
BEGIN
    if @Id = 0 -- Створення нової позиції товару в документі
        begin
            insert into DocDetails (DocId, ProductId, Quantity,
PriceR, PriceP)
                values (@DocId, @ProductId, @Quantity, @PriceR,
@PriceP)
            select @Id = SCOPE_IDENTITY()
        end
    else
        begin
            update DocDetails set DocId = @DocId, ProductId =
@ProductId, Quantity = @Quantity, PriceR = @PriceR, PriceP =
@PriceP
                where Id = @Id
        end
    end
END

```

Функція знищення працює таким чином: по ідентифікаційному номеру у базі даних знаходиться потрібний запис та видаляється.

Лістинг 2.11 – Функція видалення даних з таблиці прихідних та розхідних документів

```

GO

CREATE PROCEDURE [dbo].[DocDetailsDelete]
@Id int

AS
BEGIN
    delete from DocDetails where Id = @Id
END

```

Функція SelectAll відбирає необхідні нам дані з різних таблиць та складає їх в одну.

Лістинг 2.12 – Функція відбору даних з таблиці прихідних та розхідних документів

```

GO

CREATE PROCEDURE [dbo].[DocDetailsSelectByDocId]

```

```

@DocId int

AS
BEGIN
    select t1.*, ProductName = isnull(t2.ProductName, ''), Units =
isnull(t2.Units, '')
        from DocDetails t1 left join Products t2 on t1.ProductId
= t2.ProductId
        where t1.DocId = @DocId

        order by t1.Id
END

```

Для документів, прихідних та розхідних документів в SQL було створено в загальному 6 збережених процедур.

2.5 Створення звіту про реалізацію товарів

Як результат написання даної програми – це формування звіту, котрий показує аналіз продажу товару за потрібний період. Вікно “Звіт продаж” складається з таблиці та кнопок вибору дати для отримання обліку руху товару за певний період часу.

В процедурі [dbo.SelectZvitRealiz] описана функція Select для звіту по продажі товару за певний період часу. Реалізія даної функції показана в лістингу 2.13.

Код	Назва	Зал.Поч.Ксть	Зал.Поч.Сума	Прих.Ксть	Прих.Сума	Розх.Ксть	Розх.Сума	Зал.Пот.Ксть	Зал.Пот.Сума
3	Яблука	20,000	40,00	0,000	0,00	1,000	2,00	19,000	38,00
5	Картопля	10,000	20,00	5,000	10,00	15,000	30,00	0,000	0,00
4	Груші	20,000	100,00	0,000	0,00	10,000	50,00	10,000	50,00
6	Полуниця	10,000	100,00	0,000	0,00	0,000	0,00	10,000	100,00
1	Молоко	0,000	0,00	20,000	200,00	0,000	0,00	20,000	200,00
2	хліб	0,000	0,00	20,000	100,00	2,000	10,00	18,000	90,00

Рисунок 2.11 - Вікно сформованого звіту про продаж товару за період часу

Лістингу 2.13 – Функція звіту про реалізацію товарів

GO

```
CREATE PROCEDURE [dbo].[SelectZalProducts]
```

```

AS
BEGIN

    SET NOCOUNT ON;

    SELECT ProductId, Quantity = sum(Quantity)
        into #temp
        from DocDetails
        where DocId in (select DocId from Documents where DocType
= 1) -- Прихід товару
        group by ProductId

    insert into #temp (ProductId, Quantity)
        SELECT ProductId, Quantity = -1*sum(Quantity)
        from DocDetails
        where DocId in (select DocId from Documents where
DocType = 2) -- Реалізація товару
        group by ProductId

    SELECT ProductId, Quantity = sum(Quantity)
        into #temp2 -- Залишки товару
        from #temp
        group by ProductId

    SELECT t1.ProductId, t1.Quantity,
        t2.ProductName, t2.Units, t2.Trademark, t2.PriceR --
додаткові характеристики з довідника товарів
        from #temp2 t1 inner join Products t2 on t1.ProductId =
t2.ProductId
        where t1.Quantity != 0

END
GO

```

Дана частина програми забезпечує реалізацію звіту про реалізацію товару, таким чином компанія, яка користуватиметься даною програмою отримуватиме інформацію, який товар доступний та в якій кількості.

2.6 Реалізовані методи класів C#

В даній кваліфікаційній роботі всі основні дії, операції, процедури реалізовані за допомогою SQL, методи класів в Visual Studio, створені на мові програмування C# служать в більшості для зв'язку програми та бази даних

звідки береться вся потрібна інформація. Діаграма класів C# зображена на рисунку 2.12.

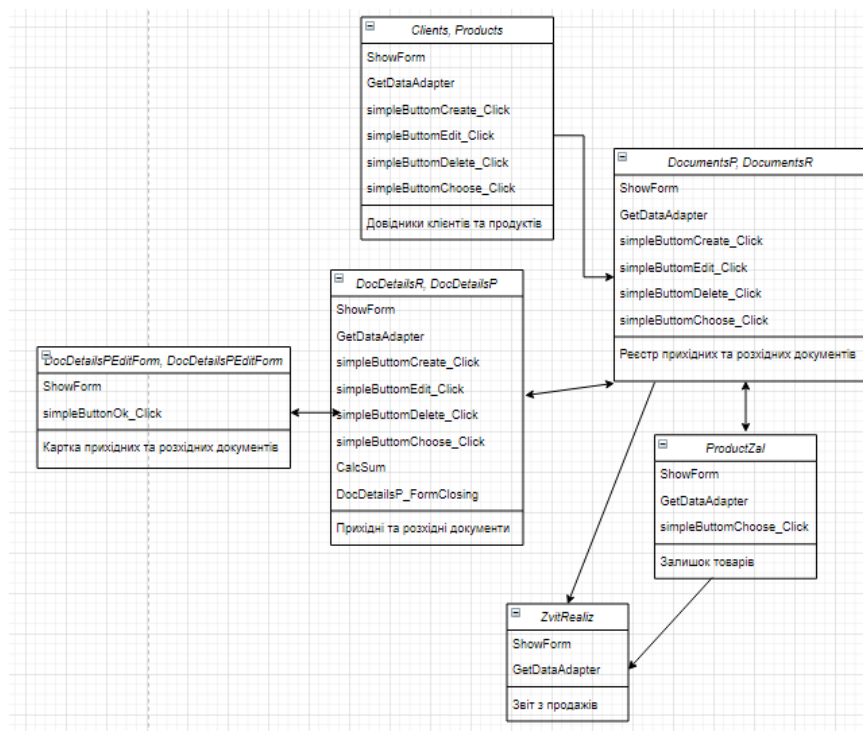


Рисунок 2.12 – Діаграма класів

Класи довідників “Clients” та “Products” – містять такі методи:

- ShowForm(bool modal) – метод для відображення форми на екрані в модальному або немодальному режимі.
- GetDataAdapter – метод, що повертає sqlDataAdapter.
- simpleButtonCreate_Click – метод, що викликається при натисненні кнопки Створити.
- simpleButtonEdit_Click – метод, що викликається при натисненні кнопки Редагувати.
- simpleButtonDelete_Click – метод, що викликається при натисненні кнопки Видалити.
- simpleButtonChoose_Click – метод, що викликається при натисненні кнопки Вибрати.

Класи реєстрів прихідних та розхідних документів “DocumentsP” та “DocumentsR” – містять такі методи :

- ShowForm(bool modal) – метод для відображення форми на екрані в модальному або немодальному режимі.

- GetDataAdapter – метод, що повертає sqlDataAdapter.

- simpleButtonCreate_Click – метод, що викликається при натисненні кнопки Створити.

- simpleButtonEdit_Click – метод, що викликається при натисненні кнопки Редагувати.

- simpleButtonDelete_Click – метод, що викликається при натисненні кнопки Видалити.

- simpleButtonChoose_Click – метод, що викликається при натисненні кнопки Вибрати.

Класи прихідних та розхідних документів “ DocDetailsP ” та “ DocDetailsP” – містять такі методи :

- ShowForm(bool modal) – метод для відображення форми на екрані в модальному або немодальному режимі.

- GetDataAdapter – метод, що повертає sqlDataAdapter.

- simpleButtonCreate_Click – метод, що викликається при натисненні кнопки Створити.

- simpleButtonEdit_Click – метод, що викликається при натисненні кнопки Редагувати.

- simpleButtonDelete_Click – метод, що викликається при натисненні кнопки Видалити.

- simpleButtonChoose_Click– метод, що викликається при натисненні кнопки Вибрати.

- simpleButtonClient_Click – метод, що викликається при натисненні кнопки вибору клієнта.

- CalcSum – метод, який підраховує загальну суму документу.

- DocDetailsP_FormClosing – метод, який викликається при закритті форми.

Класи карток прихідних та розхідних документів “DocDetailsPEditForm” та “ DocDetailsREditForm” – містять такі методи:

- ShowForm() – метод для відображення форми на екрані в модальному режимі.

- simpleButtonOk_Click – метод, що викликається при натисненні кнопки Записати.

Клас залишків “ProductsZal” – містить такі методи:

- ShowForm(bool modal) – метод для відображення форми на екрані в модальному або немодальному режимі.

- GetDataAdapter – метод, що повертає sqlDataAdapter.

- simpleButtonChoose_Click – метод, що викликається при натисненні кнопки Вибрати.

Клас звіту з продаж “ZvitRealiz” – містить такі методи:

- ShowForm() – метод для відображення форми на екрані в немодальному режимі.

- GetDataAdapter – метод, що повертає sqlDataAdapter.

2.7 Висновок до другого розділу

В другому розділі кваліфікаційної роботи було реалізовано за допомогою Microsoft Visual Studio 2017, SQL Management Studio 2012 та описано всі необхідні операції та функції, які були потрібні для програми в даній кваліфікаційній роботі. В даній було створено таблицю, котра містить такі значення: код товару, назва товару, кількісний залишок на початок вказаного періоду, значення суми початкового залишку, прихід кількість того ж товару, та його прихідна сума на товар. Продаж товару в розхідній накладній містить кількість і суму по роздрібним цінам. Відповідно до цього, автоматично розраховується значення залишків на кінець вказаного періоду: кількість і сума по прихідних цінах.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

3.1 Працездатність людини-оператора

Програмне забезпечення для автоматизації роздрібної торгівлі не може функціонувати без оператора, людини яка вноситиме всі необхідні дані в систему. Щоб роздрібна торгівля була ефективною, потрібно, щоб оператор був уважний та працездатним.

Під працездатністю людини розуміють можливість її виконувати роботу з необхідною якістю та в установлений час. Працездатність людини залежить як від зовнішніх чинників, так і від внутрішнього стану (внутрішні чинники).

До зовнішніх чинників належать:

- кількість та форма отриманої інформації,
- характер взаємостосунків в колективі,
- зручність робочого місця,
- вплив чинників середовища існування.

До внутрішніх чинників належать:

- рівень підготовки,
- тренованість людини
- емоційна стійкість.

У процесі роботи людина переживає різні функціональні стани, які зумовлюють різні рівні її працездатності.

Виділяють 4 фази працездатності: пристосування до праці, стійкої працездатності, субкомпенсації, втоми. Тривалість усіх фаз та усього циклу роботи залежить від рівня підготовки людини до роботи.

Фаза пристосування до праці - це час, протягом якого людина адаптується до майбутніх умов праці. Основний показник поступово досягає свого встановленого значення. Тривалість періоду пристосування організму до умов праці залежить від багатьох чинників, серед яких основними є інтенсивність

роботи (чим інтенсивніша робота, тим цей період коротший) та рівень готовності людини до майбутньої роботи.

Значного скорочення фази пристосування до праці можна досягти за рахунок попередньої підготовки людини до роботи та шляхом посиленого навчального навантаження. Суть останнього полягає в тому, що оператор перед початком роботи проводить короткочасне тренування щодо розв'язання однієї чи кількох задач підвищеної складності.

Фаза стійкої працездатності характеризується найвищою якістю праці при оптимальних рівнях функціонування фізіологічних систем організму. Тривалість цього періоду залежить від інтенсивності роботи. Чим інтенсивніша праця, тим коротший цей період. Найоптимальніша динамічна робота, коли цей період може бути в десятки разів довшим, ніж при статичній діяльності.

На процес стійкої працездатності великий вплив справляють емоції. Негативні, такі як, страх, невпевненість, поганий настрій – знижують працездатність. Позитивні: впевненість, спокій, бадьорий настрій – значно продовжують період стійкої працездатності.

Продовження періоду стійкої працездатності можна забезпечити:

- оптимальним рівнем напруги психофізіологічних функцій;
- комфортними умовами праці;
- правильним поєднанням режимів праці та відпочинку;
- емоційним розвантаженням;
- використанням тонізуючих напоїв (кава, чай), фармакологічних засобів, зокрема препаратів рослинного походження (вітаміни, препарати, які впливають на енергетичні та метаболічні процеси);

- інформуванням людини про наслідки її діяльності, наглядом та контролем її роботи.

Практичний досвід свідчить, що вживання легких стимуляторів допомагає знизити сонливість, сприяє підвищенню працездатності на короткий період. Однак активні стимулятори на відповідальних видах робіт здатні викликати негативний ефект - погіршується самопочуття, знижується

рухливість та швидкість реакцій. Поширене серед населення вживання транквілізаторів, викликаючи заспокоєння та запобігаючи розвитку неврозів, може знизити психічну активність, сповільнити реакції, спричинити апатію та сонливість.

Фаза субкомпенсації розглядається як початок розвитку втоми. В цей період якість праці ще зберігається на високому рівні, але тільки за рахунок перенапруги и відповідних функцій організму.

Фаза втоми характеризується чітко вираженим зниженням якості роботи при подальшому погіршенні функціонального стану людини. Об'єктивними показниками втоми є зміна частоти пульсу, дихання, зорової та слухової чутливості.

Наступною фазою життєдіяльності людини повинна бути фаза відновлення працездатності або відпочинку, яка може тривати від кількох хвилин до декількох діб.

3.2 Загальні вимоги безпеки з охорони праці для користувачів ПК

ПК є необхідністю при роботі з програмним забезпеченням, але при неправильному поводженні з ним можуть виникнути проблеми, через це існують загальноприйняті загальні вимоги безпеки з охорони праці для користувачів ПК.

Інструкції щодо вимог безпеки з охорони праці для користувачів ПК розроблена відповідно до Положення про розробку інструкцій з охорони праці, затвердженого наказом Держнаглядохоронпраці від 29.01.1998 № 9, Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці від 26.01.2005 № 15, Правил охорони праці під час експлуатації електронно-обчислювальних машин, затверджених наказом Державного комітету України з промислової безпеки, охорони праці та гірничого нагляду від 26.03.2010 № 65, Державних санітарних правил і норм роботи з візуальними дисплейними терміналами

електронно-обчислювальних машин ДСанПіН 3.3.2.007-98, затверджених постановою Головного державного санітарного лікаря України від 10.12.1998 № 7, Загальних вимог стосовно забезпечення роботодавцями охорони праці працівників, затверджених наказом Міністерства надзвичайних ситуацій України від 25.01.2012 № 67 (НПАОП 0.00-7.11-12).

Згідно цих наказів робоче місце для працюючих з відео терміналами необхідно розташувати таким чином, щоб до поля зору працюючого не потрапляли вікна, освітлювальні прилади, поверхні які мають властивість віддзеркалювання. Поверхня робочого столу не повинна бути полірованою. Для попередження відблисків на екрані відео моніторів, особливо влітку та у сонячні дні, екран відео монітора розміщується так, щоб світло від вікна падало збоку, бажано зліва.

Екран відео монітору ПК повинен знаходитись від очей користувача (надалі оператора) на відстані не менше 500 – 700 мм. Кут зору в межах від 10° до 40°. Найбільш раціональним є розташування екрану перпендикулярно до лінії зору оператора.

ПК розташовується на відстані не ближче 1 метра від джерела тепла.

Клавіатура розміщується на поверхні столу або спеціальній підставці на відстані 100-300 мм від краю, повернутого до користувача. Кут нахилу панелі клавіатури до горизонтальної поверхні повинен бути в межах від 5° до 15°. Висота робочої поверхні столу повинна бути в межах 680-800 мм.

Крісло має забезпечувати операторові зручні умови праці та фізіологічну раціональну робочу позу в процесі праці. Також крісло має забезпечувати можливість регулювання висоти поверхні сидіння, кут нахилу спинки та висоту спинки.

Для захисту від прямих сонячних променів, які створюють відблиски на екрані відео монітора на вікнах встановлюються сонцезахисні пристрої. Екран відео монітора розміщується так, щоб світло від вікна падало на робоче місце збоку, бажано зліва.

Як джерело штучного освітлення в приміщеннях, де встановлено ПК, рекомендовано використовуються люмінесцентні лампи. Можливе також застосування ламп розжарювання в світильниках місцевого освітлення. Освітленість робочого місця у горизонтальній площині на висоті 0,8 м від рівня підлоги не менш 400 лк. Вертикальна освітленість у площині екрану не більше 200 лк. Для зменшення напруженості зору необхідно забезпечити достатньо рівномірне розподілення яскравості робочої поверхні відео монітора та навколишнього простору.

У приміщеннях для роботи ПК необхідно проводити щоденне вологе прибирання та регулярне провітрювання протягом робочого дня. Видалення пилу з екрану необхідно проводити не рідше 1 разу на день.

Для захисту оператора від електромагнітних випромінювань і електростатичних полів, які створює відео монітор необхідно використовувати захисні екрани.

3.3 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи описано питання з безпеки життєдіяльності “Працездатність людини – оператора”, програмне забезпечення для автоматизації роздрібної торгівлі вимагає оператора комп’ютерного набору для вводу даних. Так як обсяг інформації, яку потрібно внести при роботі є великим, важливо, щоб оператор був працездатним, так як це також впливає на його уважність та якість виконання роботи. З охорони праці були описані “Загальні вимоги безпеки з охорони праці для користувачів ПК”, так як при неправильному поводженні з ПК можуть виникнути проблеми, від проблем зі здоров’ям у користувача до пожеж в приміщенні, тому важливо знати та дотримуватись цих правил при роботі з комп’ютером.

ВИСНОВКИ

Основним завданням кваліфікаційної роботи було створення програми для обліку продажу товарів. В даному проекті є таблиця, котра містить такі значення: код товару, назва товару, кількісний залишок на початок вказаного періоду, значення суми початкового залишку, прихід кількість того ж товару, та його прихідна сума на товар. Продаж товару в розхідній накладній містить кількість і суму по роздрібним цінам. Відповідно до цього, автоматично розраховується значення залишків на кінець вказаного періоду: кількість і сума по прихідних цінах.

При написанні кваліфікаційної роботи було створено базу даних, в якій вводилася інформація про товари, його продаж і надходження на підприємстві.

В першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр» подано інформацію про ринок роздрібною торгівлі в Україна та описано вибір програмного забезпечення для створення.

В другому розділі кваліфікаційної роботи було розроблено програмний продукт для автоматизації роздрібною торгівлі, було описано, які дії відбуваються при використанні тої чи іншої функції, кожну функцію було протестовано.

У розділі «Безпека життєдіяльності, основи охорони праці» висвітлено важливе питання для даної роботи, це “Працездатність людини–оператора”, так як саме оператор вносить всю необхідну інформацію в програмний продукт, тому важливо, щоб робітник був працездатний, адже об’єм інформації з якою прийдеться працювати в програмі є досить великий. Також було описано загальні вимоги безпеки з охорони праці для користувачів ПК.

ПЕРЕЛІК ДЖЕРЕЛ

1. Мешкова В. В. Статистичний аналіз ринку роздрібної торгівлі України / В. В. Мешкова, Н. С. Пащенко. – Львів: Наш формат, 2018. – 157 с.
2. Силкіна Ю. О. Проблеми та перспективи розвитку ринку роздрібних торговельних мереж в Україні / Ю. О. Силкіна. – Київ: Ранок, 2017. – 152-156 с. – (Економіка і менеджмент).
3. Завгородній О. Ю. Системи та програми автоматизації на підприємствах торгівлі: вимоги до функціоналу та їх реалізація / О. Ю. Завгородній. – Харків: Теза, 2021. – 210 с.
4. Дмитрів О. Р. "Цифровізація економіки та автоматизація виробництва: проблеми та шляхи їх вирішення." Тези доповідей міжнародної науково-практичної конференції „Цифрова економіка як фактор інноваційного розвитку суспільства / О. Р. Дмитрів, В. В. Семеґен. – Київ: Фоліо, 2020. – 16 с.
5. Карнаушенко А. С. Особливості автоматизації обліку торговельних підприємств / Алла Сергіївна Карнаушенко. – Дніпро: Піраміда, 2019. – 89 с.
6. Демянова Ю. О. Автоматизовані системи управління як невід’ємний елемент розвитку глобальних торговельно-роздрібних мереж / Ю. О. Демянова. – Київ: Човен, 2018. – 170-174 с.
7. Безус А. М. Перспективи інноваційного розвитку роздрібної торгівлі в Україні / А. М. Безус, Б. М. Шевчун, П. І. Безус. – Полтава: Астра, 2019. – 24-27 с.
8. Бойчук А. А. Проблеми організації автоматизації обліку на підприємствах торгівлі / А. А. Бойчук, В. О. Бойчук, Л. О. Моцна. – Київ: Фабула, 2017. – 56-60 с.
9. Карнаушенко А. С. Проблеми автоматизації обліку торговельних підприємств / Алла Сергіївна Карнаушенко. – Харків: Комора, 2019. – 220 с.
10. Mukherjee S. Indexes in Microsoft SQL Server / Sourav Mukherjee. – New-York: HarperCollins, 2019. – 64 с.

11. Mukherjee S. Popular SQL server database encryption choices / Sourav Mukherjee. – New-York: HarperCollins, 2019. – 44 c.
12. Mukherjee S. SQL Server Development Best Practices / Sourav Mukherjee. – New-York: HarperCollins, 2019. – 40 c.
13. Dejan S. SQL Server 2016 Developer's Guide / S. Dejan, M. Radivojevic, W. Durkin. – Zagreb: Cornelsen, 2017. – 68 c.
14. William A. SQL server 2017 administration inside out / Assaf William. – Washington: RELX Group, 2018. – 96 c.
15. Chmel M. SQL Server 2017 Administrator's Guide: One stop solution for DBAs to monitor, manage, and maintain enterprise databases / M. Chmel, V. Muzny. – Warsaw: Klett, 2017. – 64 c.
16. Efficiency test of Microsoft SQL Server 2016 / T.Balla, T. Radvanyi, S. Kiraly, R. Kiraly. –Amsterdam: Wolters Kluwer, 2018. – 231 c.
17. Husni M. Improved Information Retrieval Performance on SQL Database Using Data Adapter / M. Husni. – Madrid: Grupo Planeta, 2018. – 135 c. – (Materials Science and Engineering).
18. Noaman A. eCloudDB: A Unified API for Secure SQL and NoSQL Cloud Databases. / A. Noaman, E. Noaman, A. Noaman. – Washington: Wiley, 2019. – 134 c. – (Proceedings of the 2019 3rd International Conference on Cloud and Big Data Computing).
19. Snell P. Microsoft Visual Studio 2015 / P. Snell, L. Snell, M. Snell. – New-York: HarperCollins, 2019. – 120 c.
20. Kunal C. Mastering Visual Studio 2017 / Clowdhury Kunal. – Los-Angeles: RELX Group, 2017. – 74 c.
21. Sharp J. Microsoft Visual C# 2013 Step by Step. Pearson Education / John Sharp. – Boston: McGraw, 2017. – 83 c.
22. Buananno E. Functional programming in C#. Manning / Enrico Buananno. – San-Jose: HarperCollins, 2018. – 104 c.

23. Schrotenboer S. Illustrated C# 7: The C# Language Presented Clearly, Concisely, and Visually / S. Schrotenboer, D. Schrotenboer, C. Schrotenboer. – New-York: Apress, 2018. – 107 c.

24. Watts G. Developing a Declarative Analysis Language: LINQToROOT. / Gordon Watts. – Washington: Apress, 2019. – 23 c. – (EPJ Web of Conferences).

25. Strauss D. Working with Visual Studio 2019. Getting Started with Visual Studio 2019. / Dirk Strauss. – New-York: Apress, 2020. – 61 c.

26. Development of Dissolved Gas Analysis Analyzing Program using Visual Studio Program. / P.Kunagonniyomrattana, P. Kunagonniyomrattana, T. Kunagonniyomrattana, C. Supakit. – Tokyo: Shuesisha, 2019. – 235 c.

27. Johnson B. Using Git in Visual Studio 2019 / Bruce Johnson. – New-York: Apress, 2020. – 153 c.

28. Guerin B. ASP. NET con C# en Visual Studio 2017: diseño y desarrollo de aplicaciones Web / Brice-Arnaud Guerin. – Madrid: Grupo Santillana, 2018. – 65 c.

29. Bhandari P. Consuming Microsoft Cognitive APIs. / P. Bhandari, N. Bhandari, A. Bhandari. – Berkeley: Californial, 2018. – 125 c.

30. Alian M. Ncap: Network-driven, packet context-aware power management for client-server architecture / Mohammad Alian. – Washington: Scholastic, 2017. – 98 c.

ДОДАТКИ

Скрипт таблиць бази даних

```
USE [torg]
GO
```

```
CREATE TABLE [dbo].[Clients](
    [ClientId] [int] IDENTITY(1,1) NOT NULL,
    [ClientName] [varchar](100) NOT NULL,
    [TaxNumber] [varchar](12) NOT NULL,
    [ClientAddress] [varchar](200) NOT NULL,
    CONSTRAINT [PK_Clients] PRIMARY KEY CLUSTERED
(
    [ClientId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[DocDetails](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [DocId] [int] NOT NULL,
    [ProductId] [int] NOT NULL,
    [Quantity] [numeric](18, 3) NOT NULL,
    [PriceR] [numeric](9, 2) NOT NULL,
    [PriceP] [numeric](9, 2) NOT NULL,
    CONSTRAINT [PK_DocDetails] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Documents](
    [DocId] [int] IDENTITY(1,1) NOT NULL,
    [ClientId] [int] NOT NULL,
    [Suma] [numeric](18, 2) NOT NULL,
    [DocDate] [datetime] NULL,
    [DocType] [smallint] NOT NULL,
    CONSTRAINT [PK_Documents] PRIMARY KEY CLUSTERED
(
    [DocId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
```

```

CREATE TABLE [dbo].[Products](
    [ProductId] [int] IDENTITY(1,1) NOT NULL,
    [ProductName] [varchar](100) NOT NULL,
    [Units] [varchar](20) NOT NULL,
    [Trademark] [varchar](100) NOT NULL,
    [PriceR] [numeric](9, 2) NOT NULL,
    [PriceP] [numeric](9, 2) NOT NULL,
    CONSTRAINT [PK_Products] PRIMARY KEY CLUSTERED
(
    [ProductId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON)
ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Clients] ADD CONSTRAINT
[DF_Clients_ClientName] DEFAULT (') FOR [ClientName]
GO
ALTER TABLE [dbo].[Clients] ADD CONSTRAINT [DF_Clients_TaxNumber]
DEFAULT (') FOR [TaxNumber]
GO
ALTER TABLE [dbo].[Clients] ADD CONSTRAINT
[DF_Clients_ClientAddress] DEFAULT (') FOR [ClientAddress]
GO
ALTER TABLE [dbo].[DocDetails] ADD CONSTRAINT
[DF_DocDetails_DocId] DEFAULT ((0)) FOR [DocId]
GO
ALTER TABLE [dbo].[DocDetails] ADD CONSTRAINT
[DF_DocDetails_ProductId] DEFAULT ((0)) FOR [ProductId]
GO
ALTER TABLE [dbo].[DocDetails] ADD CONSTRAINT
[DF_DocDetails_Quantity] DEFAULT ((0)) FOR [Quantity]
GO
ALTER TABLE [dbo].[DocDetails] ADD CONSTRAINT
[DF_DocDetails_CenaR] DEFAULT ((0)) FOR [PriceR]
GO
ALTER TABLE [dbo].[DocDetails] ADD CONSTRAINT
[DF_DocDetails_PriceP] DEFAULT ((0)) FOR [PriceP]
GO
ALTER TABLE [dbo].[Documents] ADD CONSTRAINT
[DF_Documents_ClientId] DEFAULT ((0)) FOR [ClientId]
GO
ALTER TABLE [dbo].[Documents] ADD CONSTRAINT [DF_Documents_Suma]
DEFAULT ((0)) FOR [Suma]
GO
ALTER TABLE [dbo].[Documents] ADD CONSTRAINT
[DF_Documents_DocType] DEFAULT ((0)) FOR [DocType]
GO
ALTER TABLE [dbo].[Products] ADD CONSTRAINT
[DF_Products_ProductName] DEFAULT (') FOR [ProductName]
GO
ALTER TABLE [dbo].[Products] ADD CONSTRAINT [DF_Products_Units]
DEFAULT (') FOR [Units]

```

```
GO
ALTER TABLE [dbo].[Products] ADD CONSTRAINT
[DF_Products_Trademark] DEFAULT (') FOR [Trademark]
GO
ALTER TABLE [dbo].[Products] ADD CONSTRAINT [DF_Products_PriceR]
DEFAULT ((0)) FOR [PriceR]
GO
ALTER TABLE [dbo].[Products] ADD CONSTRAINT [DF_Products_PriceP]
DEFAULT ((0)) FOR [PriceP]
GO
ALTER TABLE [dbo].[DocDetails] WITH CHECK ADD CONSTRAINT
[FK_DocDetails_DocDetails] FOREIGN KEY([DocId])
REFERENCES [dbo].[Documents] ([DocId])
GO
ALTER TABLE [dbo].[DocDetails] CHECK CONSTRAINT
[FK_DocDetails_DocDetails]
GO
ALTER TABLE [dbo].[DocDetails] WITH CHECK ADD CONSTRAINT
[FK_DocDetails_Products] FOREIGN KEY([ProductId])
REFERENCES [dbo].[Products] ([ProductId])
GO
ALTER TABLE [dbo].[DocDetails] CHECK CONSTRAINT
[FK_DocDetails_Products]
GO
```


Скрипт збережених процедур бази даних

```
USE [torg]
GO
CREATE PROCEDURE [dbo].[ClientsDelete]
@ClientId int

AS
BEGIN
    delete from Clients where ClientId = @ClientId
END
GO

create PROCEDURE [dbo].[ClientsSelectAll]

AS
BEGIN
    select * from Clients order by ClientName
END
GO

CREATE PROCEDURE [dbo].[ClientsUpdate]
@ClientId int output, @ClientName varchar(100), @TaxNumber
varchar(12), @ClientAddress varchar(200)

AS
BEGIN
    if @ClientId = 0 -- Створення нового контрагента
        begin
            insert into Clients (ClientName, TaxNumber,
ClientAddress) values (@ClientName, @TaxNumber, @ClientAddress)
            select @ClientId = SCOPE_IDENTITY()
        end
    else
        begin
            update Clients set ClientName = @ClientName,
TaxNumber = @TaxNumber, ClientAddress = @ClientAddress
            where ClientId = @ClientId
        end
END
GO

CREATE PROCEDURE [dbo].[DocDetailsDelete]
@Id int

AS
BEGIN
    delete from DocDetails where Id = @Id
END
```

GO

```
CREATE PROCEDURE [dbo].[DocDetailsSelectByDocId]
@DocId int
```

AS

BEGIN

```
    select t1.*, ProductName = isnull(t2.ProductName, ''), Units =
isnull(t2.Units, '')
    from DocDetails t1 left join Products t2 on t1.ProductId
= t2.ProductId
    where t1.DocId = @DocId
    order by t1.Id
```

END

GO

```
CREATE PROCEDURE [dbo].[DocDetailsUpdate]
```

```
@Id int output, @DocId int, @ProductId int, @Quantity numeric
(18,3), @PriceR numeric (9,2), @PriceP numeric (9,2)
```

AS

BEGIN

```
    if @Id = 0 -- Створення нової позиції товару в документі
        begin
            insert into DocDetails (DocId, ProductId, Quantity,
PriceR, PriceP)
                values (@DocId, @ProductId, @Quantity, @PriceR,
@PriceP)
            select @Id = SCOPE_IDENTITY()
        end
    else
        begin
            update DocDetails set DocId = @DocId, ProductId =
@ProductId, Quantity = @Quantity, PriceR = @PriceR, PriceP =
@PriceP
                where Id = @Id
        end
```

END

GO

```
CREATE PROCEDURE [dbo].[DocumentsDelete]
```

```
@DocId int
```

AS

BEGIN

```
    delete from DocDetails where DocId = @DocId
    delete from Documents where DocId = @DocId
```

END

GO

```
CREATE PROCEDURE [dbo].[DocumentsSelectAll]
```

```
@DocType smallint
```

```
AS
BEGIN
    select t1.*, ClientName = isnull(t2.ClientName, '')
        from Documents t1 left join Clients t2 on t1.ClientId =
t2.ClientId
        where t1.DocType = @DocType
        order by t1.DocDate
END
GO

CREATE PROCEDURE [dbo].[DocumentsUpdate]
@DocId int output, @ClientId int, @Suma numeric (18,2), @DocDate
datetime, @DocType smallint

AS
BEGIN
    if @DocId = 0 -- Створення нового документа
        begin
            insert into Documents (ClientId, Suma, DocDate,
DocType)
                values (@ClientId, @Suma, @DocDate, @DocType)
            select @DocId = SCOPE_IDENTITY()
        end
    else
        begin
            update Documents set ClientId = @ClientId, Suma =
@Suma, DocDate = @DocDate, DocType = @DocType
                where DocId = @DocId
        end
END
GO

create PROCEDURE [dbo].[ProductsDelete]
@ProductId int

AS
BEGIN
    delete from Products where ProductId = @ProductId
END
GO

create PROCEDURE [dbo].[ProductsSelectAll]

AS
BEGIN
    select * from Products order by ProductName
END
GO

CREATE PROCEDURE [dbo].[ProductsUpdate]
```

```

@ProductId int output, @ProductName varchar(100), @Units
varchar(20), @Trademark varchar(100),
@PriceR numeric (9,2), @PriceP numeric (9,2)

AS
BEGIN
    if @ProductId = 0 -- Створення нового товару
        begin
            insert into Products (ProductName, Units, Trademark,
PriceR, PriceP)
                values (@ProductName, @Units, @Trademark,
@PriceR, @PriceP)
            select @ProductId = SCOPE_IDENTITY()
        end
    else
        begin
            update Products set ProductName = @ProductName,
Units = @Units, Trademark = @Trademark,
                PriceR = @PriceR, PriceP = @PriceP
            where ProductId = @ProductId
        end

END
GO

CREATE PROCEDURE [dbo].[SelectZalProducts]
AS
BEGIN
    SET NOCOUNT ON;

    SELECT ProductId, Quantity = sum(Quantity)
        into #temp
        from DocDetails
        where DocId in (select DocId from Documents where DocType
= 1) -- Прихід товару
        group by ProductId

    insert into #temp (ProductId, Quantity)
        SELECT ProductId, Quantity = -1*sum(Quantity)
        from DocDetails
        where DocId in (select DocId from Documents where
DocType = 2) -- Реалізація товару
        group by ProductId

    SELECT ProductId, Quantity = sum(Quantity)
        into #temp2 -- Залишки товару
        from #temp
        group by ProductId

    SELECT t1.ProductId, t1.Quantity,
        t2.ProductName, t2.Units, t2.Trademark, t2.PriceR --
додаткові характеристики з довідника товарів

```

```
        from #temp2 t1 inner join Products t2 on t1.ProductId =  
t2.ProductId  
        where t1.Quantity != 0
```

```
END
```

```
GO
```