

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

«На правах рукопису»
УДК 004.031.2

До захисту допущено:
В. о. завідувача кафедри
_____ Олександр РОЛІК
«__» _____ 2021 р.

Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Інформаційне забезпечення
робототехнічних систем»
зі спеціальності 126 «Інформаційні системи та технології»
на тему: «Система формування замовлень та прогнозування для автоматизації
логістичних процесів ритейлу з використанням методів аналізу даних»

Виконала:
студентка VI курсу, групи ІК-301мп
Миколук Валерія Ігорівна

Керівник:
асистент каф. ІСТ
Вітюк Альона Євгенівна

Рецензент:
зав. каф. СП ІПСА, д.т.н., проф.
Мухін Вадим Євгенович

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.
Студентка _____

Київ – 2021 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри

_____ Олександр РОЛІК

«___» _____ 2021 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Миколюк Валерії Ігорівні

1. Тема дисертації «Система формування замовлень та прогнозування для автоматизації логістичних процесів рітейлу з використанням методів аналізу даних», науковий керівник дисертації Вітюк Альона Євгенівна, затверджені наказом по університету від «25» 10 2021 р. № 3575-с

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження - процес прогнозування та формування замовлень на поставку товару.

4. Вихідні дані - принципи, підходи, методи та технології прогнозування та формування замовлень на поставку товару.

5. Перелік завдань, які потрібно розробити - аналіз предметної області та існуючих рішень, вибір та огляд технологій для розробки системи, проектування та розробка системи, маркетинговий аналіз стартап-проєкту.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу - 8 креслеників
7. Орієнтовний перелік публікацій - публікація статті у Міжнародному науковому електронному журналі «Наука Онлайн»
8. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	15.09.2021	
2	Огляд існуючих рішень	27.09.2021	
3	Вибір технологій реалізації	01.10.2021	
4	Проектування системи	10.10.2021	
5	Реалізація системи	01.11.2021	
6	Маркетинговий аналіз стартап - проєкту	10.11.2021	
7	Висновки	16.11.2021	

Студент

Науковий керівник

Валерія МИКОЛЮК

Альона ВІТЮК

АНОТАЦІЯ

Розмір пояснювальної записки – 102 аркуші, містить 82 ілюстрації, 23 таблиці, 10 додатків.

Ключові слова: замовлення на поставку товару, логістика, замовлення, поставка товару, рітейл, продажі, запаси, товар.

У даній роботі розглянуто особливості логістичних процесів галузі рітейлу, досліджено існуючі аналоги проектованої системи, визначено їх плюси і мінуси.

Для забезпечення безперебійного функціонування галузі торгівлі важливим є відлагодження, раціоналізація і забезпечення стабільності логістичних процесів. Саме вони забезпечують управління товарними потоками, допомагають організувати постачання швидко та з мінімумом витрат. Завдяки оптимізації транспортування на кожному етапі істотно знижується підсумкова собівартість продукції [1].

Метою магістерської дисертації є розробка системи для автоматизації основного процесу логістики – автоматичного формування замовлень на поставку товару. Аби цього досягти було виконано постановку завдань, які необхідно виконати та спроектовано принцип формування замовлень, виявлено показники, які впливають на даний процес та яким чином вони мають бути враховані, реалізована можливість ручного створення замовлень користувачем та редагування автоматично створених як самою системою на основі аналізу даних, так і клієнтом; враховано показники товарного запасу.

Було проведено дослідження та опубліковано статтю на тему «Аналіз показників необхідних для автоматичного формування замовлень на поставку товару у галузі рітейлу» у Міжнародному науковому електронному журналі «Наука Онлайн».

ABSTRACT

Explanatory note size – 102 pages, contains 82 illustrations, 23 tables, 10 applications.

Key words: purchase orders, logistics, order, delivery of goods, retail, sales, stocks, goods.

This paper considers the features of logistics processes in the retail industry, explores existing analogues of the designed system, identifies their advantages and disadvantages.

Debugging and rationalization logistics processes is important for the stable functioning of the trade sector. It helps to organize deliveries quickly and with a minimum costs. Due to the optimization of transportation at each stage the final cost of products is reduce.

The goal of the master's dissertation is to develop a system for automation the main logistics process - automatic forming purchase orders. The tasks were determined, the principle of order forming was designed, indicators that influence this process were identified. User can create orders manually and edit automatic orders.

A study was conducted and an article was published on the topic «Analysis of indicators for the automatic forming purchase orders in the retail» in the International scientific electronic journal "Science Online".

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Загальний огляд проблеми автоматичного формування замовлень на поставку товару	10
1.2 Огляд існуючих рішень	12
1.3 Постановка задачі.....	21
2 ВИБІР ТА ОБГРУНТУВАННЯ ЕЛЕМЕНТІВ ТА ТЕХНОЛОГІЙ	22
2.1 Мова програмування C# та IDE MS Visual Studio	22
2.2 Мова програмування T-SQL та СУБД MS SQL Server	24
2.3 SQL Server Integration Services.....	26
2.4 ADO .NET	27
2.5 Методи прогнозування	27
2.6 Методи аналізу даних	28
3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ	32
3.1 Принцип та особливості формування замовлень.....	32
3.2 Програмна реалізація формування замовлень	36
3.3 Автоматична корекція замовлення.....	48
3.4 Контроль товарного запасу	50
3.5 Прогнозування даних та створення замовлення	59
3.6 Отримання даних з інших серверів	61
3.7 Розробка інтерфейсу користувача	63
3.8 Робота користувачів з системою	69
4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ	82
4.1 Опис ідеї проєкту	82
4.2 Технічний аудит ідеї проєкту.....	84
4.3 Аналіз ринкових можливостей запуску стартап-проєкту.....	85
4.4 Розробка ринкової стратегії стартап-проєкту	92
4.5 Розробка маркетингової програми стартап-проєкту	96

ВИСНОВКИ	100
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	101
ДОДАТОК А Підготовка прогнозу	Ошибка! Закладка не определена.
ДОДАТОК Б Створення замовлень.....	Ошибка! Закладка не определена.
ДОДАТОК В Контекстна діаграма.....	Ошибка! Закладка не определена.
ДОДАТОК Г Декомпозиція першого рівня контекстної діаграми	Ошибка!
	Закладка не определена.
ДОДАТОК Д Діаграма вимог до системи	Ошибка! Закладка не определена.
ДОДАТОК Е Діаграма варіантів використання	Ошибка! Закладка не определена.
ДОДАТОК Ж Схема алгоритму автоматичної корекції	Ошибка! Закладка не определена.
ДОДАТОК И Автоматичний розрахунок товарного запасу	Ошибка! Закладка не определена.
ДОДАТОК К Використання модулів в залежності від типу замовлення ...	Ошибка!
	Закладка не определена.
ДОДАТОК Л SSIS пакет для отримання даних зі складської системи	Ошибка!
	Закладка не определена.

ВСТУП

У наш час роздрібна торгівля є важливою ланкою не тільки економіки, а і інших сфер суспільного життя - товар від виробника переходить до споживача, споживач отримує необхідний продукт для життєдіяльності, а постачальник оплати за його виробництво.

Роздрібна торгівля - це товарообмінний процес, спрямований на задоволення потреб людей шляхом вільного продажу товарів і послуг, що представляють для них цінність, для кінцевого некомерційного використання. Роздрібна торгівля може здійснюватися в будь-яких формах: це може бути торгівля в спеціально обладнаному приміщенні, особисті продажі, продаж за допомогою телебачення та інтернету.

Останнім часом дуже динамічно розвивається практика продажу товарів широкого вжитку, побутової техніки, продуктів харчування, будівельних матеріалів та інших груп в одному місці, тобто в супермаркеті. Все зроблено для зручності споживачів, щоб вони змогли закупитися всім необхідним у великому приміщенні, а не шукати маленькі точки по всьому місту [1]. Економіка диктує свої правила, тому власникам доводиться забезпечувати покупців якомога більшим асортиментом і надавати послуги, зручні для відвідувачів. Роздрібна реалізація продукції різних категорій кінцевому споживачу називається рітейлом.

Головна ознака рітейлу велика різноманітність товарів і послуг, які можна придбати в одному місці. Для покупців рітейл привабливий тим, що в таких магазинах можна знайти багато варіантів товару, зробити більш зважений вибір і придбати саме те, що потрібно.

Рітейл - це принцип організації торгівлі, згідно з яким супермаркети, гіпермаркети, інтернет-магазини закупають товар оптом, а потім продають його з націнкою в своїх торгових точках. Ціни на продукцію у великих рітейлерів значно нижчі, ніж в немережевих торгових точках, тому рітейл привертає більшу кількість споживачів.

У даній роботі розглядається продуктовий рітейл - торгівля різноманітними продуктами харчування на значній площі, наприклад, в супермаркетах і

гіпермаркетах. Це найстабільніша сфера ритейлу, яка працює навіть під час кризи, адже харчові продукти - це товари першої необхідності. Наприклад під час локдауна, пов'язаного з пандемією COVID-19, практично всі торгові точки були закриті, крім аптек і магазинів, які продають харчові продукти.

Варто зауважити, що наявність товару гарантує не тільки задоволення споживача [1] і функціонування економіки, а і пов'язана з питанням виживання тому дуже важливо не допускати помилок, пов'язаних з наповненістю складів, неправильного зберігання товарів, неповним асортиментом у супермаркеті або взагалі відсутністю товару «на полицях» тощо, всього цього можна уникнути при налагодженості логістичних процесів.

Логістика - один з найголовніших процесів будь-якого виробництва, підприємства тощо, який часто порівнюють з системою кровообігу організму людини. Покупець перш за все бачить магазин (розташування, функціонування, формат), назву, торговельні марки, результати маркетингу, асортимент, сервіс і саме за цими показниками оцінює ритейлер. Але за цим всім стоять інші важливі процеси, з яких саме логістичні є ваговою ланкою, адже завдяки їх стабільному і раціональному функціонуванню покупець в результаті може побачити товар на полицях.

Логістичні процеси охоплюють собою велику область бізнес-процесів: це і підтримка функціонування складів, контролю залишків товару у них, переміщення товару, налагодження зв'язків з постачальниками, контроль і налагодження транспортних перевезень, формування замовлень на поставку товару [1].

Тому питання розробки системи, яка надаватиме можливість автоматизувати один з основних процесів логістики ритейлу, а саме прогнозування та формування замовлень на поставку товару, і можливість її синхронізації з системами транспорту, складу є актуальним.

Метою розробки даної системи є автоматизація логістичних процесів ритейлу, а саме прогнозування і формування замовлень на поставку товару з використанням методів аналізу даних [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальний огляд проблеми автоматичного формування замовлень на поставку товару

Багато хто думає, що логістика в ритейлі це тільки зберігання товару на складах і організація перевезень, але насправді сюди входить набагато більше функцій і система (комплекс систем), яка покликана автоматизувати повний спектр логістичних процесів має надавати доступ до наступних розділів -

- управління замовленнями та рахунками завжди є першочерговим завданням для керівників операцій. Завжди бажано мати централізовані можливості управління замовленнями та виставленням рахунків, і якщо вони будуть більш гнучкими, це гарантуватиме, що рішення здатне відповідати всім вимогам;

- одним з найважливіших аспектів логістики та управління ланцюгами поставок є оптимізація управління запасами. Це допомагає забезпечити, щоб одночасно організація була здатна зберігати запаси товарів, сировини, матеріалів та була доступною, коли це потрібно. Якісне програмне забезпечення для управління складом дозволяє організаціям оптимізувати свої запаси, що дозволить підвищити точність та загальну ефективність, забезпечуючи своєчасне та економічно ефективне задоволення потреб;

- будь-яка галузь повинна мати можливість прогнозувати попит, щоб мати змогу так само планувати свої ресурси. Прогнозування дозволяє компаніям закуповувати необхідне, тим самим усуваючи необхідність перетарюватися сировиною або навіть зберігати готову продукцію;

- дані є важливою частиною будь-якого бізнесу, оскільки дають можливість аналізувати результати та оцінювати процеси і за необхідності оптимізовувати операції [2].

Якщо ми говоримо про формування замовлень на поставку товару то важливим є:

- врахування при його формуванні наявності вільного місця на складі, габарити транспортного засобу який буде здійснювати перевезення, фасування товару;
- вивчення попиту та пропозиції товарів, які оптимально відповідають за потреби покупців;
- планування поставок таким чином, щоб товар не псувався, не було його надлишку або навпаки, нестачі, тобто аналізувати минулі продажі і формувати прогноз продаж;
- аналізувати і відповідно мати доступ до інформації про потребу магазинів на певний вид товару;
- мати можливість формувати різні типи замовлень в залежності від їх особливості – короткий термін придатності товару (необхідний склад з певним обладнанням для зберігання цього товару), переміщення товару між постачальником і складом, переміщення товару між складами, між складом і магазином тощо;
- можливість задавати циклічність/повторюваність певних замовлень на певний період (формувати замовлень раз на місяць, два рази на місяць тощо);
- обмінюватися офіційною документацією, яка підтверджуватиме відправлення і отримання замовлення постачальником і складом/магазином відповідно [1].

Всі ці процеси впливають на швидкість доставки, якість і взагалі на можливість надходження товару на полиці супермаркетів.

Автоматизована система дозволить управляти всіма ланками і вільно взаємодіяти учасникам на кожному етапі логістичного процесу, забезпечуючи формування замовлень на поставку товару одночасно отримуючи інформацію зі складських систем, систем керування транспортом тощо; надаватиме можливість контролю та формування товарного запасу.

1.2 Огляд існуючих рішень

Наразі не існує програмного забезпечення (окремої повноцінної системи або комплексу систем), яке б повністю охоплювало весь спектр процесів необхідних для повного функціонування логістичних процесів ритейлу – складська система, система транспорту, система формування замовлень на поставку товару.

На даний момент всі схожі рішення не в змозі повністю покрити навіть одну з наведених вище систем. Далі розглядаються існуючі аналоги.

Система «BAS Управління торгівлею» виступає інструментом автоматизації ряду процесів галузі торгівлі – прогнозування і аналіз продаж, управління запасами складу, замовленнями, поставками, підтримка користувачів, управління цінами та аналіз витрат, аналіз торгівельної діяльності, формування звітів.

Відмінні риси даного продукту. Програма надає можливість формувати замовлення постачальнику на основі планів закупівель (рис. 1.1) та планів збирання.

← → Друк документа

Друк українською мовою Копій: 1

План купівель № 1 від 11 березня 2019 р.

Сценарій: Тиждень, Тиждень Вид плану: Основний
 Період з 01.04.2019 по 05.05.2019 Статус: У підготовці
 Підрозділ: Відділ закупівель

№	Товар	Постачальник	Оферта	Склад	Од. вим.	01.04 - 07.04	08.04 - 14.04
1	Телевізор "JVC"	Koenig Holding GmbH	Поставка побутової техніки	Склад побутової техніки	шт	1	1

Всього найменувань 1.

Відповідальний Бахшиєв П.І. Затвердив

Рисунок 1.1 – План купівель [4]

Прикладне рішення дозволяє визначити різні види цін на товари. Для кожного виду ціни визначаються свої правила розрахунку (спосіб задання ціни). Передбачена можливість призначити алгоритми розрахунку цін. Для створення формул розрахунку в системі передбачений «конструктор формул» (рис. 1.2).

Головне Ціни номенклатури Задачі

Записати та закрити Записати Створити на підставі

Використовувати спрощені можливості

Основе Уточнення по цінових групах Правила округлення цін Пороги спрацьовування

Найменування: Спеціальна

Ідентифікатор для формул: Спеціальная

Валюта ціни: грн ☐ Ціна включає ПДВ

Використовувати:

☐ При продажу клієнтам ☐ При випуску продукції, робіт і послуг за фіксованою

☐ При введенні на підставі документів поставки ☐ При передачі продукції давальця

☒ При передачі між організаціями

Спосіб завдання ціни: Довільна формула від інших видів цін

Формула: $\text{Макс}([\text{Себестоимость}] * 1.2, [\text{Мелкооптовая}])$...

Рисунок 1.2 – Конструктор формули ціни [4]

У системі передбачена можливість призначати знижки/націнки ручні і автоматичні (рис. 1.3).

Найменування	Умова надання
Умови надання знижок (націнок)	
Графік оплати: Передплата	За графік оплати
Кількість в документі не менше 10 од. по номенклатурі сегменту Кавоварки	За разовий об'єм продажів
Кількість в документі не менше 100 од. по номенклатурі сегменту Взуття	За разовий об'єм продажів
Кількість в документі не менше 2 од. по номенклатурі сегменту Кухонні комбайни	За разовий об'єм продажів
Кількість в документі не менше 2 од. по номенклатурі сегменту Соковижималки	За разовий об'єм продажів
Кількість в документі не менше 4 од. по номенклатурі сегменту Побутова техніка	За разовий об'єм продажів
Кількість в документі не менше 50 од. по номенклатурі сегменту Спортивне взуття	За разовий об'єм продажів
Кількість проданого товару по партнеру за минулий місяць не менше 100 од. по номенклатурі сегменту Взуття	За накопичений об'єм продажів
Кількість проданого товару по партнеру за минулий місяць не менше 20 од. по номенклатурі сегменту Меблі	За накопичений об'єм продажів
Кількість проданого товару по партнеру за минулий місяць не менше 50 од. по номенклатурі сегменту Побутова техніка	За накопичений об'єм продажів
Перший поплат партнеру	За накопичений об'єм продажів

Рисунок 1.3 – Умови надання знижок [4]

Відсоток і сума знижки можуть уточнюватися за ціновими групами, за певним сегментом номенклатури, на будь-яку номенклатуру зі списку (рис. 1.4).

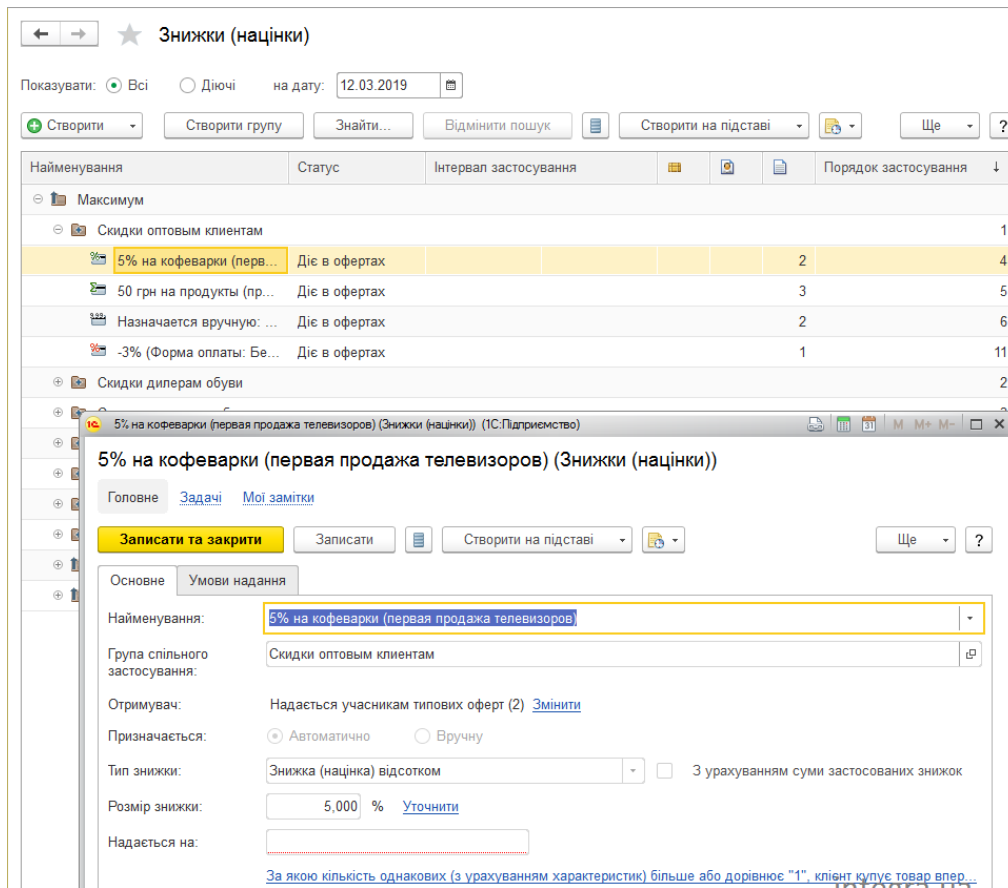


Рисунок 1.4 – Призначення процентної знижки [4]

Оформлення замовлень постачальникам. Для відображення попередньої домовленості про умови та строки поставки товарів, які купуються у постачальника, порядок і строки оплати, передбачено використання функціоналу замовлень постачальникам (рис. 1.5).

Замовлення постачальникам можуть формуватися автоматично на підставі розрахунку потреб в забезпеченні товарами відповідно до параметрів забезпечення (терміном поставок товарів від постачальника, середньоденного споживання тощо).

Також замовлення постачальнику може бути введено на підставі раніше оформлених замовлень покупців, угоди з клієнтом. При оформленні замовлення в рамках угоди автоматично заповнюються основні умови закупівель (валюта, вид ціни постачальника, параметри оплати, використання договорів тощо).

← → ☆ **Замовлення постачальнику ДОО0-000101 від 05.04.2012 14:17:22**

Головне [Погодження](#) [Файли](#) [Документообіг](#) [Задачі](#) [Мої замітки](#)

Провести і закрити [Іконки] [Друк] [Звіти]

Статус: **До надходження** ▼ Пріоритет: **Средний** ▼

Основне **То**

Не погоджено

Погоджений

Підтверджений

До надходження

Закритий

Номер:

Постачальник:

Контрагент:

Оферта:

Оплата:

Коментар:

Ерація: **Купівля у постачальника** ▼

Ганізація: **Добро** ▼ □

Говір: **На поставку кухонної техніки** ... □

Склад: **Центральний склад** ... □

Сплачено за замовленням: 20 760.00 грн 100%

Залік оплати

Оплата: [До оплати 06.04.2012 \(100%\)](#)

Рисунок 1.5 – Замовлення постачальнику [4]

Контроль виконання зобов'язань постачальником здійснюється за допомогою статусів і станів документа, планів (рис. 1.6). Статуси для документа присвоюються вручну [5][6].

← → ☆ **Формування замовлень постачальнику за планом**

Крок 1 з 3. Вибір планів

Створювати замовлення за:

Сценарій: **Тиждень** ▼

Період замовлення: **Цей тиждень** ... ×

☒ Без деталізації загальні плани по підприємству.

☐ По підрозділу **...** ▼

Відібрати плани за:

Постачальник: **База "Електроніка та побутова те** ▼ ×

Період надходження: **25.02.2019 - 31.03.2019** ... ×

Оферта: **Поставка кухонної техніки (База "** ▼ ×

Склад: **Склад побутової техніки** ▼ ×

Номенклатура: **Кавоварка BRAUN KF22R** ▼ ×

[Довільні відбори](#)

Рисунок 1.6 – Формування замовлень постачальникам по планам [4]

В програмі є функція формування замовлень та можливість обліку товарів в розрізі серій і термінів придатності (рис. 1.7).

Склад	Приміщення	Залишок	Днів до закінчення строку придатності, %
Артикул	Номенклатура, Характеристика	Од. вим.	Термін придатності
Придатний до			
Коса "Роздріб"			
Арт-7890000	Масло вершкове,	кг	«не указан»
10.04.12			60,000
10.04.12			60,000 не годен
Арт-7890000	Масло вершкове "Сільска",	пач	«не указан»
10.04.12			3,000
10.04.12			3,000 не годен
Арт-8790000	Молоко "Простоквашина" 1.5%,	цвг	«не указан»
10.04.12			30,000
10.04.12			30,000 не годен
Арт-8790000	Молоко "Простоквашина" 3.2%,	цвг	«не указан»
10.04.12			12,000
10.04.12			12,000 не годен
Магазин "Продукти"			
Арт-7890000	Масло вершкове,	кг	«не указан»
10.04.12			200,000
10.04.12			200,000 не годен
Арт-8000000	Масло вершкове "Екстра",	пач	«не указан»
14.04.12			100,000
14.04.12			5,000 не годен
15.04.12			95,000 не годен
Арт-7890000	Масло вершкове "Сільска",	пач	«не указан»
12.04.12			200,000
12.04.12			80,000 не годен
13.04.12			200,000 не годен
Арт-8790000	Молоко "Простоквашина" 1.5%,	цвг	«не указан»
07.04.12			180,000
07.04.12			80,000 не годен

Рисунок 1.7 – Залишки товарів за строками придатності [4]

Підставою для відвантаження товарів зі складу є документи реалізації. На їх підставі автоматично формуються видаткові ордери [4].

У цілому програмна система «BAS Управління торгівлею» надає багато інструментів для керування торгівлею, перевагою є можливість працювати з цінами і знижками напряму, зручний інтерфейс користувача, управління і запасами складу та можливість у цій же системі формувати замовлення на поставку, але все ж даний продукт більше направлений на взаємодію з користувачем – результат часто залежить від введення певних даних тут і зараз, більшість процесів не автоматичні, інформації, яку отримуємо на виході при формуванні замовлення, не достатньо для аналізу коректності створеного замовлення (на виході бачимо назву товару, склад для поставки та його ціну).

1С:Підприємство 8.0. Підсистема управління закупівлями визначає яку кількість товару і коли необхідно купити. Основними функціями є формування

замовлень постачальникам на основі замовлень покупців, формування та ведення графіків поставок замовлень та платіжного календаря.

Чітка і вчасно усвідомлена картина потреб дозволяє організувати взаємодію з постачальниками, оптимізувати вартість товарів, що закуповуються, й мінімізувати складські витрати. Інформація про закупівлі доступна з інших підсистем.

Формування замовлень постачальникам на підставі замовлень покупців. У системі управління замовленнями покупцями передбачено їх резервування і розміщення (у тому числі автоматичне) на складах і в оформлених замовленнях покупців. Замовлення на відсутні товари можна формувати автоматично. Також є можливість у звітах бачити дефіцит товару, просрочені замовлення (рис. 1.8).

Группировка	Показатели	Остаток на дату отчета	Итого	(10.11.2003)	(16.11.2003)	(14.12.2003)	(19.12.2003)	01.01.2004
Сардина ж/б 0,25	Поступление	-	-	-	-	-	-	-
	Отгрузка	-	50 000,000	-	20 000,000	20 000,000	10 000,000	-
	Остаток	88 400,000	-	88 400,000	88 400,000	48 400,000	38 400,000	38 400,000
Скумбрия ж/б 0,25	Поступление	-	-	-	-	-	-	-
	Отгрузка	-	20 000,000	-	10 000,000	-	10 000,000	-
	Остаток	22 000,000	-	22 000,000	12 000,000	12 000,000	2 000,000	2 000,000
Макрель ж/б 0,425	Поступление	-	-	-	-	-	-	-
	Отгрузка	-	84 000,000	-	20 000,000	4 000,000	40 000,000	-
	Остаток	80 000,000	-	80 000,000	60 000,000	56 000,000	16 000,000	16 000,000
Тунец ж/б 0,337	Поступление	-	-	-	-	-	-	-
	Отгрузка	-	-	-	-	-	-	-
	Остаток	20 000,000	-	20 000,000	20 000,000	20 000,000	20 000,000	20 000,000
Тунец ж/б 0,425	Поступление	-	12 000,000	12 000,000	-	-	-	-
	Отгрузка	-	21 800,000	-	1 000,000	8 800,000	12 000,000	-
	Остаток	-	12 000,000	11 000,000	2 200,000	(9 800,000)	(9 800,000)	-

Рисунок 1.8 – Товарний календар [7]

В оперативному товарному календарі відображається інформація про надходження, відвантаження і залишки номенклатури (причому як реальні, так і планові).

Звіт Аналіз замовлень покупців і внутрішніх замовлень визначає по замовленнях покупців потреби в товарах постачальників (рис. 1.9).

Номенклатура.Основной поставщик	Запланировано	Отгружено (в ед. хранения)	Осталось отгрузить	Осталось обеспечить	Заказ размещен	Заказ поставщику
База "Продукты"	35,000	35,000	35,000	35,000		
Никитаева-частное лицо	35,000	35,000	35,000	35,000		
Заказ покупателя ТД000000001 от 02.06.2014 12:50:23	35,000	35,000	35,000	35,000		
Причуда (вафли)	10,000	10,000	10,000	10,000		
Молоко "Останкинское" 1.5%	10,000	10,000	10,000	10,000		
Молоко "Домик в деревне" 1.5%	5,000	5,000	5,000	5,000		
Масло деревенское	10,000	10,000	10,000	10,000		
Итого	35,000	35,000	35,000	35,000		

Рисунок 1.9 – Звіт Аналіз замовлень покупців і внутрішніх замовлень [8]

На підставі даних звіту формуються замовлення постачальникам (рис. 1.10). Таблична частина автоматично заповнюється за зазначеними замовленнями (можна кількома) покупців тільки відсутніми товарами. При цьому вже враховуються і відвантажені, і зарезервовані, і розміщені товари.

№	Номенклатура	Количе...	Ед.	Цена	Сумма	%..	Сумм...	Всего
1	Крупа "Геркулес"	10,000	кг			18%		
2	Масло деревенское	1	шт			18%		
3	Молоко "Домик в деревне" 1.5%	1	шт			18%		
4	Молоко "Останкинское" 1.5%	1	шт			18%		
5	Причуда (вафли)	1	упак	0,35	3,50	18%	0,53	3,50

1USD = 25,9029руб., Тип цен: Цены поставщика "Базы продукты" Всего (USD): 3,50
 НДС (в т. ч.): 0,53

Рисунок 1.10 – Замовлення постачальнику [8]

Сформовані замовлення автоматично відправляються по електронній пошті постачальникам (рис. 1.11).

Исходящее, Рассмотрено, (HTML) : Электронное письмо 0000000009 от 02.06.2014 15:35:16

Действия: Отправить, Записать и закрыть, Формат текста (HTML), Печать

От кого: test@1C.ru <test@1C.ru>

Кому: info@baza-produktov.ru

Копии:

Тема: База "Продукты" Заказ поставщику №1 от 02.06.2014 на 03 июня 2014 (вторник) (Предварительный за)

☐ Не рассмотрено, рассмотреть после: . . .

Письмо, Дополнительно, Вложения

Заказ поставщику № 1 от 02 июня 2014 г.

Поставщик: АОЗТ База "Продукты", ИНН 1456784324

Покупатель: ЗАО "Торговый дом Комплексный", ИНН 7705260609, КПП 770501001, 117452, Москва г, Москва, Симферопольский б-р, дом № 78, корпус 1, тел.: (095) 1123456, доб. 789

№	Товары (работы, услуги)	Кол-во	Ед	Цена	Сумма
1	Крупа "Геркулес"	10	кг		
2	Масло деревенское	10	шт		
3	Молоко "Домик в деревне" 1.5%	5	шт		
4	Молоко "Останинское" 1.5%	10	шт		
5	Вафли "Пчелка"	10	упак	0,35	3,50
					3,50
					0,53

Всего наименований 5, на сумму 3,50 USD
Три доллара 50 центов

Руководит: /Ковальков Н. Ф./ Бухгалтер: /Белина А. Г./

Рисунок 1.11 – Приклад автоматического листа поставщику [8]

«1С:Управление торговлей 8.0» дозволяє задавати необмежену кількість аналітик за кожною номенклатурою й вести облік товару за кожною з характеристик (розмір, колір, повнота і т. д.). Характеристики можна встановити різні для кожної товарної групи або навіть товару (рис. 1.12). За кожною позицією номенклатури можна зберігати дані про серію, сертифікати, що має конкретний товар, і терміни придатності.

Справочник Номенклатура

Действия: Выбрать, Действия

Код	Артикул	Наименование	Услуга	Базовая	Е
00001		ТОВАРЫ	ло...		
00005		ПРОДУКТЫ ПИТАНИЯ	ло...		
00021		МОРЕ ПРОДУКТЫ	ло...		
00083		Сардина ж/б 0.25	ло...	шт.	

Список Характеристики номенклатуры (Сардина ж/б 0.25)

Действия: Действия

Наименование: Без головы, Большой, Тихий океан

Элемент Характеристики номенклатуры: Новый *

Действия: Действия

Наименование: Без головы, Средний, Индийский океан

Свойство	Значение
Разделка	Без головы
Размер	Средний
Район лова	Индийский океан

Рисунок 1.12 – Довідник «Номенклатура» [7]

Крім оперативного планування закупівель, система надає можливість контролювати графік поставчань і графік платежів за замовленнями [7][8][9].

На відміну від попереднього програмного забезпечення «1С Підприємство» має вже більше інформації у замовленні на поставку товару при його формуванні і оперує не просто заданою інформацією, а опирається на договори між постачальниками; замовлення після формування автоматично відправляються, окрім цього є можливість моніторингу дефіциту товару і розвинуті довідники, але як і в попередньому варіанті даних не достатньо, що говорить про неможливість повного переходу на тільки автоматичне формування замовлень, немає розподілу за типами замовлень та врахування розмірів замовлень, відсутня взагалі акційний модуль або акційні дані, які б надавали можливість створення акційних замовлень.

Наглядніше все це можна розглянути у порівняльній таблиці (табл 1.1).

Таблиця 1.1 – Порівняння існуючих систем

	BAS Управління торгівлею	1С Підприємство	Система, що розробляється
Формування замовлення з використанням договорів	-	+	+
Формування замовлення аналізуючи потребу магазинів	+	+	+
Робота з цінами	+	-	+
Робота з акціями	-	-	+
Поділ замовлень за типами	-	-	+
Робота з графіком постачань	-	+	+
Управління запасами складу	+	-	-
Робота з інформацією по запасам	+	-	+
Повнота замовлення	-	-	+

З таблиці 1.1 видно, що існуючі аналоги не в змозі покрити вище наведені функції, які є важливими для формування замовлень. Всі ці недоліки розглянутих систем підтверджують доцільність розробки нового продукту.

1.3 Постановка задачі

Для досягнення цілі розробки магістерської дисертації необхідно –

- проаналізувати особливості теми дослідження, наявність готових програмних рішень на ринку та відповідність їх потребам процесу логістики з формування замовлень;

- сформулювати функціональні та інші вимоги до програмного продукту;

- дослідити та обрати технології і методи за допомогою яких здійснюватиметься реалізація продукту;

- спроектувати систему;

- розробити схему бази даних;

- розробити інтерфейс користувача;

- реалізувати систему використовуючі дані з проведеного дослідження та проектування.

Об'єктом дослідження магістерської дисертації є процес прогнозування та формування замовлень на поставку товару.

Предметом дослідження - принципи, підходи, методи та технології прогнозування та формування замовлень на поставку товару.

Висновки до розділу. У даному розділі було здійснено огляд області дослідження, визначено основні функції системи, предмет та об'єкт дослідження, здійснено постановку задачі та виконаний аналіз існуючих рішень, який показав відсутність на даний момент на ринку продуктів, які б покривали всі необхідні функції та актуальність розробки подібної системи.

2 ВИБІР ТА ОБГРУНТУВАННЯ ЕЛЕМЕНТІВ ТА ТЕХНОЛОГІЙ

Система для формування замовлень та прогнозування логістичних процесів галузі ритейлу є десктопним додатком, який буде розроблено з використанням наступних методів та технологій – мови програмування T-SQL та C# (платформа .NET Core), СУБД MS Sql Server, IDE MS Visual Studio, ADO.Net, SQL Server Integration Services (SSIS), методи прогнозування та аналізу даних.

2.1 Мова програмування C# та IDE MS Visual Studio

C# є об'єктно-орієнтованою мовою і підтримує поліморфізм, успадкування, перевантаження операторів, статичну типізацію. Об'єктно-орієнтований підхід (ООП) дозволяє вирішити завдання з побудови великих, але в той же час гнучких, масштабованих і розширюваних додатків, основною ідеєю якого є об'єднання даних і дій в одне ціле.

Коли говорять про C#, нерідко мають на увазі технології платформи .NET (Windows Forms, WPF, ASP.NET, Xamarin). І, навпаки, коли говорять .NET, нерідко мають на увазі C#. Однак, хоча ці поняття пов'язані, ототожнювати їх невірно. Мова C# була створена спеціально для роботи з фреймворком .NET, проте саме поняття .NET дещо ширше.

Можна виділити наступні її основні риси:

Підтримка декількох мов - основою платформи є загальномовне середовище виконання Common Language Runtime (CLR), завдяки чому .NET підтримує кілька мов: окрім C# це також VB.NET, C++, F#, а також різні діалекти інших мов, прив'язані до .NET, наприклад, Delphi.NET. При компіляції код на будь-якому з цих мов компілюється в збірку спільною мовою CIL (Common Intermediate Language) - свого роду асемблер платформи .NET. Тому за певних умов є можливість зробити окремі модулі однієї програми на різних мовах.

Кросплатформеність. .NET є платформою доступною для перенесення (з деякими обмеженнями). Наприклад, остання версія платформи на даний момент - .NET 5 підтримується на більшості сучасних ОС Windows, MacOS, Linux. Тобто використовуючи різні технології на платформі .NET, можна розробляти програми на мові C # для Windows, MacOS, Linux, Android, iOS, Tizen.

Потужна бібліотека класів. .NET представляє єдину для всіх підтримуваних мов бібліотеку класів.

Різноманітність технологій. Загальномовне середовище виконання CLR і базова бібліотека класів є основою для цілого стека технологій, які розробники можуть задіяти при побудові тих чи інших додатків. Наприклад, для роботи з базами даних в цьому стеку технологій призначена технологія ADO.NET і Entity Framework Core. Для побудови графічних додатків з багатим насиченим інтерфейсом - технологія WPF і UWP, для створення більш простих графічних додатків - Windows Forms. Для розробки мобільних додатків – Xamarin і т.д.

Продуктивність. Згідно ряду тестів додатки на .NET 5 в ряді категорій сильно випереджають додатки, побудовані за допомогою інших технологій.

Також ще слід відзначити таку особливість мови C# і фреймворка .NET, як автоматичне збирання сміття. А це означає, що в більшості випадків не доведеться, на відміну для прикладу від C ++, піклуватися про звільнення пам'яті. CLR сама викличе збирач сміття і очистить пам'ять [10].

Для написання і відладки коду, а також подальшої публікації додатків варто використовувати IDE, яка являє собою комплекс з декількох інструментів, а саме: текстового редактора, компілятора або інтерпретатора, засобів автоматизації збирання і відладчика. Крім цього, IDE може містити інструменти для інтеграції з системами управління версіями і інші корисні утиліти. Є IDE, які призначені для роботи тільки з однією мовою програмування, проте більшість сучасних IDE дозволяє працювати відразу з декількома [11].

MS Visual Studio представляє собою повністю інтегровану середу розробки. Вона спроектована таким чином, щоб робити процес написання коду, його

налагодження та компіляції в збірку для поставки кінцевим споживачам якомога простішим.

Основні можливості Visual Studio - за допомогою текстового редактора можна готувати код на мові C # (Visual Basic, C ++ та ін.) в зручному режимі – між рядками створюються відступи, виконується перевірка на синтаксичні помилки, підкреслюються місця які будуть викликати помилки при компіляції, реалізовано засіб IntelliSense, який забезпечує автоматичне відображення імен класів, полів або методів під час їх введення і т.п; візуальний редактор форм дозволяє розміщувати бажані елементи управління для призначеного інтерфейсу користувача і доступу до даних в проєкті, а Visual Studio потім автоматично додає в вихідні файли код, який необхідний для створення екземплярів цих елементів в проєкті; доступ до цілого ряду інших утиліт, які дозволяють переглядати і змінювати різні аспекти комп'ютера або мережі, не покидаючи середовища розробки. Завдяки цим інструментам, можна переглядати виконувані служби та активні підключення з базами даних, заглядати в таблиці на сервері SQL Server тощо.

Також Visual Studio містить графічні редактори і конструктори XML, забезпечує підтримку розробки програм Windows, орієнтованих на мобільні пристрої, підтримку розробки програм Microsoft Office і Windows Workflow Foundation, містить вбудовану підтримку рефакторинга коду і інструменти візуального конструювання класів [12].

2.2 Мова програмування T-SQL та СУБД MS SQL Server

База даних (БД) – це набір взаємопов'язаних даних та взаємозв'язків між ними, який зберігається певним чином.

Для створення, ведення і використання бази даних використовується система управління базами даних (СУБД).

MS SQL Server є однією з найбільш популярних СУБД в світі. Дана СУБД підходить для самих різних проєктів: від невеликих додатків до великих високонавантажених.

SQL Server довгий час була винятково системою управління базами даних для Windows, проте починаючи з версії 16 ця система доступна і на Linux. Вона характеризується такими особливостями як продуктивність (працює дуже швидко), надійність і безпека (шифрування даних), простота.

Для організації баз даних MS SQL Server використовує реляційну модель, яка передбачає зберігання даних у вигляді таблиць, кожна з яких складається з рядків і стовпців. Кожен рядок зберігає окремий об'єкт, а в стовпчиках розміщуються атрибути цього об'єкта. В основі реляційної моделі лежить теорія множин, яка і має на увазі об'єднання різних об'єктів в одне ціле, під одним цілим в базі даних як раз і мається на увазі таблиця.

Для можливості роботи з базою даних використовується SQL Server Management Studio - утиліта для адміністрування та конфігурації компонентів Microsoft SQL Server. Утиліта включає скриптовий редактор та графічну програму, яка працює з налаштуваннями і об'єктами сервера.

Для взаємодії з базою даних застосовується мова SQL (Structured Query Language). Клієнт (наприклад, зовнішня програма) відправляє запит на мові SQL за допомогою спеціального API. СУБД належним чином інтерпретує і виконує запит, а потім надсилає клієнту результат виконання.

Мова SQL - це стандарт, він реалізований у всіх реляційних базах даних, але у кожної СУБД є розширення цього стандарту, так званий діалект мови SQL.

Виділяються два різновиди мови SQL: PL-SQL і T-SQL. PL-SQL використовується в таких СУБД як Oracle і MySQL. T-SQL (Transact-SQL) застосовується в SQL Server. Власне тому в рамках поточного проєкту буде розглядатися саме T-SQL.

T-SQL має всі можливості мови SQL, проте призначена вона для вирішення завдань програмування, не пов'язаних з реляційними даними. Іншими словами, на T-SQL можна повноцінно програмувати, використовуючи змінні, цикли, умовні конструкції та інші можливості. T-SQL дозволяє реалізовувати складні алгоритми бізнес-логіки у вигляді збережених процедур і функцій, тим самим створюючи так звані «програми» всередині бази даних.

Крім цього, T-SQL спрощує написання SQL запитів за рахунок величезної кількості вбудованих системних функцій, а також дозволяє адмініструвати SQL Server і отримувати системну інформацію за допомогою представлень, процедур і функцій [13][14].

2.3 SQL Server Integration Services

SQL Server Integration Services (SSIS) - інструмент для реалізації процесу перенесення даних з одного джерела в інший.

Якщо необхідно перенести дані з одного сервера на інший, першими думками про рішення даної задачі, як правило, є такі варіанти: експортувати дані з одного сервера в excel і скористатися майстром імпорту або скористатися лінком між серверами.

Однак лінку може і не бути, а завдання щодо перенесення даних може виявитися періодичним, тоді кожен раз доведеться повторювати вивантаження даних і їх імпорт. У таких випадках на допомогу якраз приходить SSIS.

SSIS - це набір контейнерів, що з'єднуються між собою. Програма складається з потоку управління (Control Flow), який може включати в себе потік даних (Data Flow) [15].

SSIS включає в себе графічні інструменти і майстри для створення та налагодження пакетів; завдання для виконання функцій робочого процесу, таких як операції FTP, виконання операторів SQL і відправлення повідомлень електронної пошти; джерела і призначення даних для вилучення і завантаження даних; перетворення для очищення, агрегування, об'єднання і копіювання даних; база даних управління SSISDB для адміністрування виконання і зберігання пакетів; і інтерфейси прикладного програмування (APIs) для програмування об'єктної моделі служб Integration Services [16].

2.4 ADO.NET

Технологія ADO.NET є посередником, що здійснює зв'язок між базою даних і додатком на мові C #.

ADO.NET представляє собою технологію роботи з даними, яка заснована на платформі .NET Framework. Ця технологія являє набір класів, через які ми можемо відправляти запити до баз даних, встановлювати підключення, отримувати відповідь від бази даних і виробляти ряд інших операцій.

Основу інтерфейсу взаємодії з базами даних в ADO.NET представляє обмежене коло об'єктів: Connection, Command, DataReader, DataSet і DataAdapter. За допомогою об'єкта Connection відбувається установка підключення до джерела даних. Об'єкт Command дозволяє виконувати операції з даними з БД. Об'єкт DataReader зчитує отримані в результаті запиту дані. Об'єкт DataSet призначений для зберігання даних з БД і дозволяє працювати з ними незалежно від БД. І об'єкт DataAdapter є посередником між DataSet і джерелом даних. Головним чином, через ці об'єкти і буде йти робота з базою даних.

Однак щоб використовувати один і той же набір об'єктів для різних джерел даних, необхідний відповідний провайдер даних. Власне через провайдер даних в ADO.NET і здійснюється взаємодія з базою даних. Причому для кожного джерела даних в ADO.NET може бути свій провайдер (для MS SQL Server, для OLE DB тощо), який власне і визначає конкретну реалізацію вищевказаних класів [17].

2.5 Методи прогнозування

Прогнозування - це метод, в якому використовується як накопичений у минулому досвід, так і поточні припущення щодо майбутнього з метою його визначення. Якщо прогнозування виконано якісно, результатом буде картина майбутнього, яку можна застосовувати як основу для планування.

Розрізняють такі методи прогнозування:

- неформальні. Керівництво покладається на різні джерела письмової та усної інформації як на допоміжний засіб з метою прогнозування та вироблення цілей;
- якісні. Якщо якість інформації недостатня, або керівництво не розуміє складних методів, або коли кількісна модель є надто дорогою, керівництво може використовувати якісні моделі прогнозування. При цьому майбутнє прогнозують експерти, до яких звертаються за допомогою;
- кількісні, що застосовуються для прогнозування, якщо є підстави вважати, що діяльність у минулому мала певну тенденцію, яку можна продовжити в майбутньому, і коли наявної інформації достатньо для виявлення статистично достовірних тенденцій або залежностей;

Вирізняють такі типові методи кількісного прогнозування:

- аналіз часових рядів, що ґрунтується на дослідженні подій, які відбулися в минулому, є основою для планування. Його можна провести за допомогою таблиці або графіка шляхом нанесення на координатну сітку точок, що відповідають подіям минулого. Цей метод часто використовується з метою оцінювання попиту на товари й послуги, визначення потреб у запасах і прогнозування структури збуту;
- каузальне (причинно-наслідкове) моделювання — спроба спрогнозувати те, що відбудеться у подібних ситуаціях, шляхом дослідження статистичної залежності між фактором, що розглядається, й іншими змінними. Ця залежність називається кореляцією. Чим тісніша кореляція, тим більша придатність моделі для прогнозування [18].

У даній системі використовуються кількісні методи прогнозування - аналіз часових рядів та причинно-наслідкове моделювання.

2.6 Методи аналізу даних

Регресійний аналіз. Основною метою регресійного аналізу є визначення наявності та характеру зв'язку між змінними. У регресійному аналізі моделюється взаємозв'язок однієї випадкової змінної від однієї чи кількох інших випадкових змінних.

Цілі регресійного аналізу

- передбачення залежної змінної за допомогою незалежних змінних;
- визначення вкладу окремих незалежних змінних у варіацію залежної;
- регресійний аналіз не можна використовувати для визначення наявності

зв'язку між змінними, оскільки наявність такого зв'язку і є передумовою застосування цього виду аналізу.

Факторний аналіз. Принцип факторного аналізу полягає у тому, щоб маючи велику кількість параметрів, виділити з них малу кількість загальних батьківських параметрів та аналізувати вже не кожен параметр окремо, а групи параметрів як єдиний комплексний параметр. За допомогою цього є можливим оптимізувати структуру даних для аналізу. Факторний аналіз дозволяє скорити число змінних та класифікувати дані.

Приклад застосування даного методу - покупець оцінює супермаркет за великою кількістю параметрів. За допомогою факторного аналізу їх можна розподілити за певними чинниками у групи – «робота персоналу», «якість товару», «взаємодія з покупцем» тощо.

Специфікою цього методу є те, що при об'єднанні параметрів в групи кожен фактор акумулює у собі загальні закономірності, що переважають у всіх параметрах, відкидаючи особливості кожного параметра окремо.

Кластерний аналіз. Метою кластерного аналізу є виділення порівняно невеликої кількості груп об'єктів, якнайбільше схожих між собою всередині групи, і якнайбільше відмінних від цих об'єктів у інших групах.

Для прикладу провівши аналіз покупок клієнтів у супермаркетах, можна виявити яка вікова група, з яким доходом надає перевагу тому чи іншому товару.

У факторному аналізі групуються стовбці, тобто мета – аналіз структури множини ознак та виявлення узагальнених факторів.

У кластерному аналізі – групуються рядки, тобто мета – аналіз структури безлічі об'єктів.

Кореляційний аналіз. Кореляційний аналіз дозволяє судити наскільки схоже поведуться різні змінні. У найзагальнішому вигляді прийняття гіпотези про наявність

кореляції означає, що зміна значення змінної А відбудеться одночасно з пропорційною зміною значення Б: якщо обидві змінні зростають, то позитивна кореляція; якщо одна змінна зростає, а друга зменшується – негативна кореляція.

При вивченні кореляцій намагаються встановити, чи існує якийсь зв'язок між двома показниками в одній вибірці (наприклад, між зростанням та вагою дітей або між рівнем IQ та шкільною успішністю) або між двома різними вибірками (наприклад, при порівнянні пар близнюків), і якщо цей зв'язок існує, то чи супроводжується збільшення одного показника зростанням (позитивна кореляція) або зменшенням (негативна кореляція) іншого [19].

Дерева рішень. Дерева рішень - це спосіб подання правил в ієрархічній послідовній логічній структурі, що дозволяє співвіднести об'єкт або ситуацію на вході з одним або декількома вихідними (термінальними) вузлами [20].

Під правилом розуміється логічна конструкція представлена у вигляді «якщо... то».

Метод дерев рішень може допомогти у прийнятті складного рішення, на яке впливають десятки параметрів.

Нейронні мережі. Нейронна мережа – це послідовність нейронів, з'єднаних між собою синапсами. Структура нейронної мережі прийшла у світ програмування прямо з біології. Завдяки такій структурі машина знаходить здатність аналізувати і навіть запам'ятовувати різну інформацію.

Нейронні мережі використовуються для вирішення складних завдань, які вимагають аналітичних обчислень подібних до тих, що робить людський мозок. Найпоширенішими застосуваннями нейронних мереж є: класифікація образів - розпізнавання літер, мови тощо; кластеризація / категоризація - застосовується для отримання знань, стиснення даних та дослідження властивостей даних; апроксимація функцій - шумопоглинання прийому сигналу різної природи; прогноз - прогноз курсу валют, прогноз погоди; оптимізація - призначення штату працівників із низки умінь і чинників є класичним прикладом; управління - оптимальне керування двигуном, кермо на кораблях, літаках [21].

Основним мінусом нейронних мереж є те, що процес навчання нейронної мережі та процес прийняття рішень абсолютно неконтрольовані. Що робить у собі нейронна мережа, зрозуміти неможливо, оскільки аналізуються тестові дані (відбувається навчання нейронної мережі), і одночасно з цим система намагається мінімізувати помилку, автоматично змінюючи внутрішні параметри.

Для аналізу даних необхідних для формування замовлень використовувалась дерева рішень, регресійний та кореляційний аналіз.

Висновки до розділу. У даному розділі було здійснено огляд методів і технологій, які використовувались при розробці системи для формування замовлень та обгрунтовано їх вибір. Було обрано для реалізації системи використовувати мову програмування C# та IDE MS Visual Studio, базу даних і алгоритми роботи з нею - T-SQL та СУБД MS SQL Server, отримання даних з інших серверів за допомогою SSIS; взаємодія інтерфейсу з БД реалізовано ADO.Net; для роботи з даними було обрано дерева рішень, регресійний і кореляційний аналіз та аналіз часових рядів та причинно-наслідкове моделювання для прогнозу.

3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1 Принцип та особливості формування замовлень

У даному розділі розглядається головний процес, а саме формування замовлень.

Основні типи замовлень -

- стелажні замовлення;
- замовлення по дистрибуції;
- замовлення на виробництво;
- замовлення на імпорт;
- замовлення по надлишкам;
- роздрібні замовлення.

По типу контрагента -

- на зовнішнього постачальника;
- на внутрішнього постачальника;
- від зовнішнього клієнта.

Кожен тип замовлення формується у відповідний час, даний процес можливий за допомогою використання джобів у MS SQL Management Studio – дій які запускають виконання певних процесів у зазначений час.

Алгоритм врахування факторів під час формування замовлення виглядає наступним чином –

- дані договорів (Dogovor, ScheduleDogovor, ServiceDogovor) ;
- дані про артикулу (Lager, LagerAssortiment) ;
- прогноз продажів (PrognozSales, PrognozSalesAction) ;
- товар в дорозі (GoodsInWay) ;
- залишки на філіалі (FilialStores).

Детальніше етапи алгоритму представлено у Додатку Б.

Перш за все про необхідність формування того чи іншого замовлення система розуміє аналізуючи дані договорів – своєрідних об'єктів, які одночасно є угодою з

постачальниками по певному переліку послуг/товарів і які вказують як часто і в який день тижня необхідна поставка того чи іншого товару на певний або склад, або магазин [1].

У ScheduleDogovor (рис. 3.1) відображається частота створення замовлення на конкретний склад.

	DogovorID	FillID	day1	day2	day3	day4	day5	day6	day7
1	468351	2082	0	1	0	1	1	0	0
2	468351	2102	0	0	0	0	1	0	0

Рисунок 3.1 – Дані з ScheduleDogovor

У даному випадку за договором можуть формуватися замовлення на два склади, для першого поставки будуть 3 рази на тиждень (вівторок, четвер і п'ятниця), для другого раз на тиждень (п'ятниця).

По даному договору на дані склади може бути доставлено тільки певний асортимент товару (рис. 3.2).

	DogovorID	LagerID	Rasf_	PricePurch
	468351	24137	768.000	18.11000
	468351	24138	1200.000	12.85000
	468351	24139	1200.000	14.31000
	468351	27940	1050.000	10.11000
	468351	36046	900.000	10.11000
	468351	93796	900.000	10.11000
	468351	135298	400.000	24.45000
	468351	135300	400.000	24.45000
	468351	135302	400.000	24.45000
)	468351	135304	400.000	24.45000
	468351	135306	400.000	24.45000
)	468351	135402	400.000	24.45000

Рисунок 3.2 – Асортимент товару по договору

Також додатково можна отримати доповнену інформацію по артикулу, наприклад його розфасовка, ціна доставки тощо.

У відповідній таблиці є змога переглядати інформацію про постачальника, що працює по даному договору і супутню інформацію - контактні дані, тип товару, термін дії договору і т.д (рис. 3.3).

ID	PostavshikID	DateBegin	DateFinish
468351	10784473	2021-03-31 00:00:00.000	2022-12-18 00:00:00.000

Рисунок 3.3 – Дані з таблиці Dogovor

Окрім цього існує системна таблиця у якій є можливість задавати параметри для певного договору, це може бути ідентифікація типу замовлення, відмітка чи треба зважати при формуванні на різні фактори (продажі, товар у дорозі тощо).

Наступним показником, який система відбирає для формування замовлення є артикули - аналогічно до договорів аналізується інформація, який артикул може бути замовлено на певний склад (асортимент) або якщо ми говоримо про магазин, то окрім цього ще залишок на самому магазині та на полицях [1] (рис. 3.4).

LagerId	Filld	ZapasPolka
596357	2016	7.26
596357	2017	6.13
596357	2018	4.83
596357	2019	3.23
596357	2020	5.36
596357	2021	6.78
596357	2022	8.76
596357	2023	8.68
596357	2024	11.87
596357	2025	6.39
596357	2026	7.46
596357	2027	3.50
596357	2028	5.91
596357	2029	3.17

Рисунок 3.4 – Дані з LagerAssortiment

Наступним етапом є аналіз продажів, а саме визначення середніх продаж, прогнозу майбутніх продаж.

Основні типи прогнозів для формування замовлень по філіалам:

- 4-х тижневий прогноз (стелажні замовлення);
- 14-ти денний короткий прогноз;
- 4-х тижневий середній прогноз (з урахуванням нульових продажів);
- нульовий прогноз - елітні товари наприклад екзотичні фрукти;
- сезонна модель.

Детальніше взаємозалежності між прогнозами зображено у Додатку А.

Розрахунок продаж по артикулу\філіалу відбувається за формулою 3.1.

$$\text{Середньозважені середньоденні (і (розрах))} = \frac{\sum \text{середньоденні продажі} * \text{коефіцієнт}}{\sum \text{коефіцієнт}} \quad (3.1)$$

де $\sum$ - сума показників.

Розрахункові відрізняються тим, що з періодів відкидаються ті дні, де кількість продажів = 0 (тобто якщо період дорівнює трьом дням і в один з днів продажі = 0, то при визначенні середньоденних продажів за період, відбувається ділення не на 3 дні, а на 2).

Для товару по групі А якщо продажі за період = 0, то коефіцієнт періоду прирівнюємо до 0.

Продажі беруться не акційні (акційні продажі визначаються за наявністю по даному артикулу філіалу в даний період запису в таблиці акцій. Є запис незалежно від коефіцієнта, продажі виключаються). Продаж зі знижкою 25%/50% враховується коефіцієнтом 0.25.

Розрахунок стандартного середнього відхилення прогнозу продаж від реальних продаж відбувається за формулою 3.2.

$$\bar{\sigma}_{FE} = \sqrt{\frac{\sum_{i=1}^n (FS - RS)^2}{n}} \quad (3.2)$$

де FS – (Forecast sales), прогноз продажів. Прогноз продажів за минулий період(N). Період для розрахунку має складати не менше 2 місяців. $N > 60$;

RS – (Real sales), фактичні продажі. Продажі за період N;

$\bar{\sigma}_{FE}$ – Стандартне середнє відхилення FS від RS в абсолютному вимірі. Тобто фактична кількість продажів товару, яку ми не точно прогнозуємо як в більшу так і в меншу сторону.

Де для розрахунку беруться не акційні дні (акція відсутня, або коефіцієнт дорівнює 1) і залишки на початок дня і на кінець дня більші ніж 0 ($St > 0$ and $isnull(As,1) = 1$). В результаті отримуємо кількість продажів, яку в середньому ми не очікували отримати, або які не виправдали наш прогноз.

St – (Stores). Залишки на філіалі. Сума залишків філіалу за період N.

As – (Actions), коефіцієнт акції. Наявність маркетингової активності за період N.

Якщо йти за алгоритмом далі відбувається робота з наявністю залишків на складі або магазині та товару у дорозі, все це визначається безпосередньо алгоритмом формування замовлення про що йтиме мова далі [1].

3.2 Програмна реалізація формування замовлень

Формування замовлення на зовнішнього постачальника. Джоб, який відповідає за даний процес, у певний час запускає процедуру створення замовлень - `proc_AutoOrder_RC`, обмеження для створення задаються за допомогою вхідних параметрів (рис. 3.5).

```
ALTER PROCEDURE [dbo].[proc_AutoOrder_RC]
@FilID          int,
@PostavshikID   int = 0,
@TimeByMacro    int = 0,
@StatusZakaz    int = 0,
@DogovorID      int = -1,
@DaysOff        int = 0,
@TypeZakazStatus int = 0
```

Рисунок 3.5 – Вхідні параметри для формування замовлень на внутрішнього постачальника

Де @filid це склад або магазин.

@PostavshikID - постачальник.

@TimeByMacro - час відпрацювання замовлення (за замовченням 10, якщо встановити 0 - всі графіки).

@StatusZakaz - статус замовлення після створення.

@DogovorID - номер договору (за замовченням -1 всі договори). Договори у собі містять інформацію по необхідному графіку переміщень замовлень, який постачальник на який філіал який вид товару переміщає і як часто.

@DaysOff - тип графіка, встановлюється >0 якщо необхідно перенести створення замовлення через вихідні/святкові дні.

@TypeZakazStatus - визначення типу замовлення, наприклад стелажні замовлення на внутрішнього постачальника також пожують розподілятися за типами - 0 - звичайні замовлення, 1 - акційні, 2 замовлення по формулі сирної кімнати (коли є артикул, який потім видозмінюється, наприклад маємо головку сиру, яка на складі нарізається шматками, одне замовлення перетворюється в інше) тощо.

І також в самій процедурі формується @DateLast - дата створення замовлень, getdate().

Якщо запуск відбувається без вказівки параметрів то будуть створені замовлення по всіх філіалах, постачальниках та договорах.

Приклад запуску по філіалу у джобі зображено на рисунку 3.6.

The screenshot shows a configuration window for a job step. The 'Step name' is 'Create Zakaz_FilID_2080'. The 'Type' is 'Operating system (CmdExec)'. The 'Run as' user is 'CSSISDBProxy'. The 'Command' field contains the SQL command: 'osql -E -S KVCEN14-SQLS004\LITE004 -b -Q "exec AUTOORDER.dbo.proc_AutoOrder_RC 2080, 0, 0, 21"'. There are 'Open...' and 'Select All' buttons below the command field.

Step name:	Create Zakaz_FilID_2080
Type:	Operating system (CmdExec)
Run as:	CSSISDBProxy
Command:	osql -E -S KVCEN14-SQLS004\LITE004 -b -Q "exec AUTOORDER.dbo.proc_AutoOrder_RC 2080, 0, 0, 21"

Рисунок 3.6 – Запуск формування замовлень на внутрішнього постачальника

У свою чергу процедура виконує запуск SSIS пакету RunCreateOrdersRC_Pilot.dtsx (рис. 3.7).

```
EXEC CSSISDB.SSISDB.ssis.proc_Package@Execute@Generic
    @ExecFolderName = N'Logistic Integration Catalog',
    @ExecPackageName = N'RunCreateOrdersRC_Pilot.dtsx',
    @ExecProjectName = N'AutoorderShortage',
    @ExecEnvironmentName = NULL,
    @Exec32BitRuntime = true,
    @Parameters = @ParametersXML
```

Рисунок 3.7 – Запуск SSIS пакету

Структура ssis-пакету зображена на рисунку 3.8.

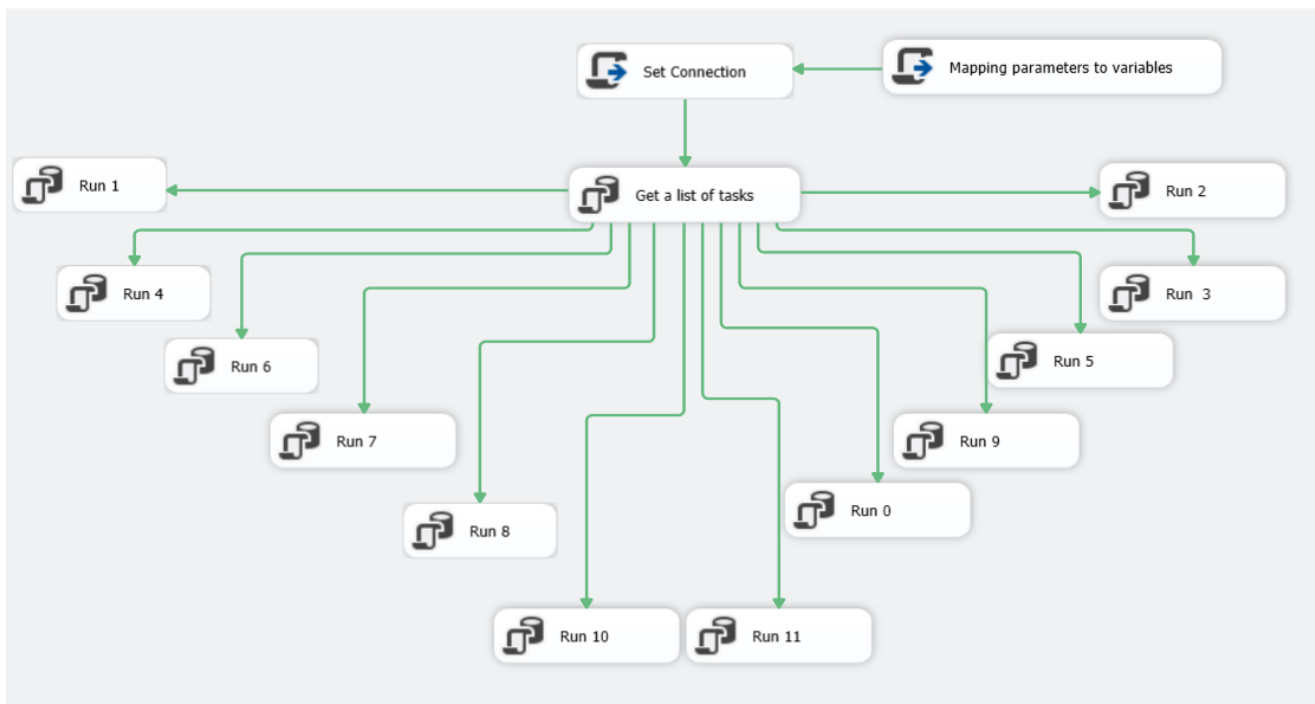


Рисунок 3.8 – Структура RunCreateOrdersRC_Pilot.dtsx

Наприклад у запуску рис. 3.6 першим кроком необхідно визначити всі договори для даного філіалу, з яких буде зрозуміло які замовлення формуватимуться, який буде постачальник товару і асортимент товару. Для цього викликається процедура `proc_AutoOrder_RC@GetDogovor` (рис. 3.9).

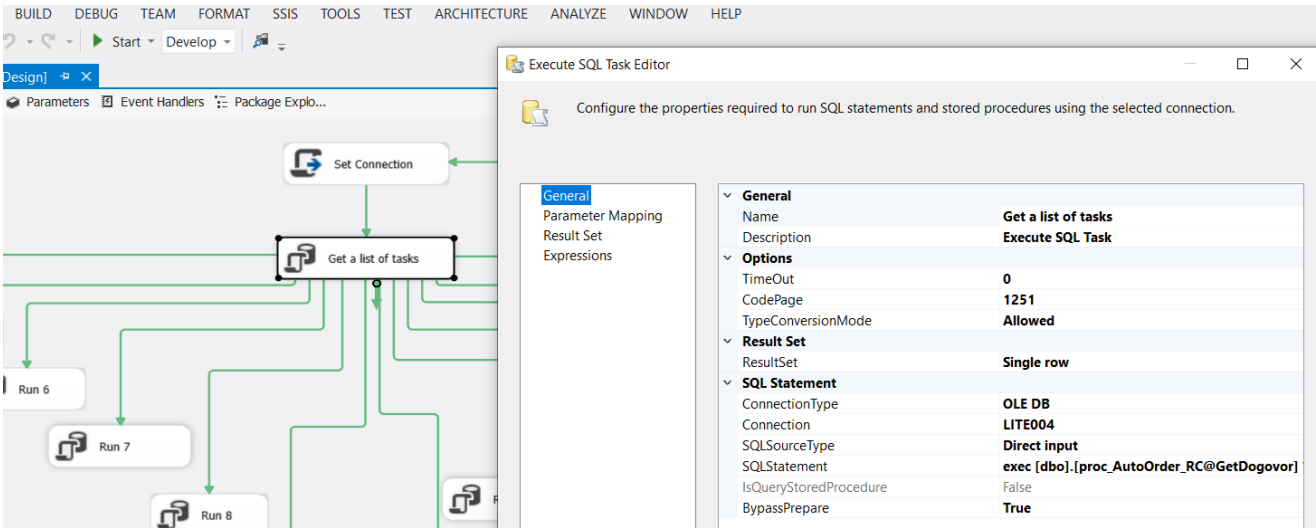


Рисунок 3.9 – Виклик proc_AutoOrder_RC@GetDogovor

Окрім цього GetDogovor формус завдання на виконання для подальших процесів, кожне завдання відповідає за створення певного замовлення.

Також у GetDogovor визначається наявність дочірніх замовлень за допомогою параметрів у договорі і одразу запускається формування на них завдань процедурою proc_AutoOrder_RC@CreateZakazChild

Після отримання необхідної інформації створюється черга формування замовлень процедура proc_AutoOrder_RC@RunQueue (рис. 3.10).

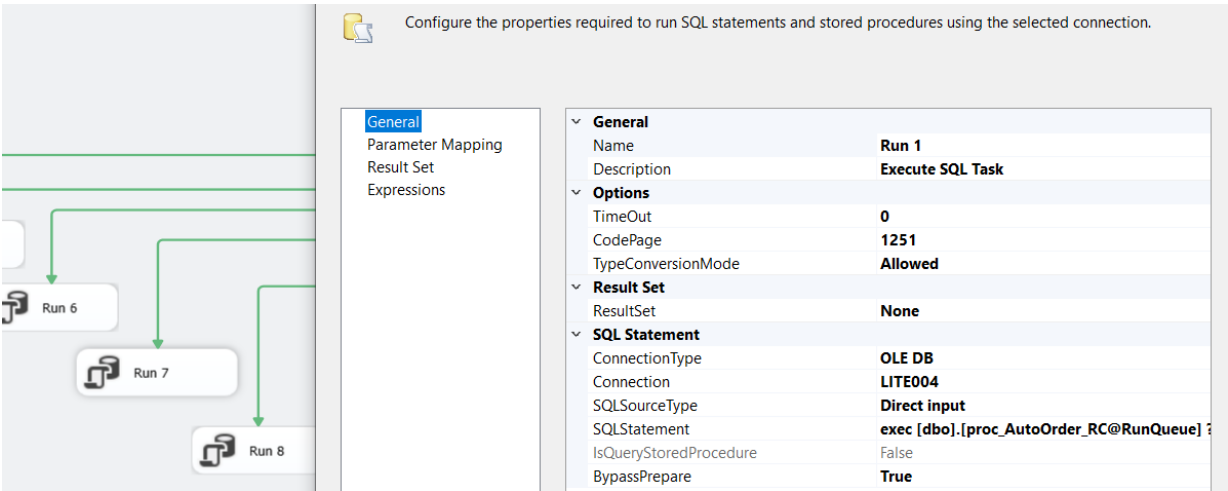


Рисунок 3.10 – Запуск створення черги

Таких запусків у SSIS пакеті 12, одночасно запускається на виконання одразу декілька процесів у паралельному режимі. Тобто на минулій ітерації створено завдання на виконання, чергу, і на даному етапі будуть паралельно підхоплюватися 12 завдань і на їх основі формуватимуться самі замовлення. Якщо один з процесів раніше завершив формування замовлення він може підхопити наступне вільне завдання.

У процедурі RunQueue відбувається безпосередньо запуск процедури на створення замовлень - proc_AutoOrder_RC@CreateZakaz.

Для формування замовлень на початку відбувається відбір даних для обробки, а саме все що стосується договорів та артикулів, необхідного асортименту товару. Далі проводиться аналіз наявних залишків зі складу та магазинів по відповідному товару, що можливе завдяки синхронізації зі складськими системами.

Важливим фактором, що впливає на кількість товару у замовленні є період який він має покривати. Даний проміжок розраховується наступним чином – кількість днів від дати формування замовлення до дати поставки + кількість днів від дати поставки до дати наступної поставки товару. Тобто під час розрахунку кількості варто брати запас і враховувати, коли очікувати наступне замовлення [1].

Прогнозування. Прогнози формуються на основі минулих замовлень, у системі фіксуються продажі товару, а саме як швидко це відбувалося і яка періодичність в залежності від дня тижня тощо. Використовуючи дану інформацію отримується прогноз на наступний обраний період – 7-30 днів (Додаток А)

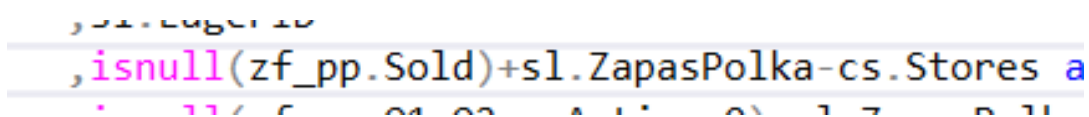
Оскільки кожен постачальник працює по відповідним договорам, замовлення по товару формується зі сталою частотою на основі даних по прогнозам, залишкам і т.д. Внести якісь глобальні зміни у замовлення система не може, а бувають ситуації коли це потрібно.

Якщо говорити наприклад про затримку доставки товару на магазини, відсутність продажів, кількість товару не змінилась, у даних ситуаціях при формуванні нового замовлення завдяки показникам про товар у дорозі і залишки система визначить, що наразі нове замовлення не потрібне і не створить його, у цьому випадку все відпрацює коректно.

Але якщо не має ніяких криз, а є ситуація, коли для прикладу на певний товар діє акція у магазині на певний період, це несе за собою ризик недостачі товару, оскільки на акційний товар більший попит у споживачів.

Аби забезпечити можливість автоматично покривати дану потребу наступним етапом при формуванні замовлення є наявність акцій по певному товару. Даний об'єкт містить у собі інформацію як довго проходитиме акційний період, прогноз саме акційних продажів, необхідна додаткова кількість до основного замовлення відповідно до середнього прогнозу акційних продажів, щоб все покрити, і додатково страховий запас на випадок успіху товару і перевищення попиту за середній прогноз продаж [1].

Відповідно сама кількість товару формується за допомогою виразу зображеного на рисунку 3.11.



```
,isnull(zf_pp.Sold)+sl.ZapasPolka-cs.Stores a
```

Рисунок 3.11 – Визначення кількості при формуванні замовлення

Де Sold - це прогноз продажів саме для цього замовлення по-артикульно, ZapasPolka – це розмір полиць у супермаркеті (скільки товару можливо виставити на вітрину) і Stores – це наявні залишки (сюди також входить товар у дорозі).

Якщо ZapasPolka це задане сталє значення, Stores було розглянуте вище, то на Sold варто зупинитися. Дана змінна, яка найбільше відповідає за кінцеву кількість у замовленні формується з використанням наступної функції (рис. 3.12).

```

func_FormulaZakaz ( dl.TypeFormulaID,
  case when (dl.ZeroStore = 1) then 0
        else dl.GoodsInWay
  end,
  ( case when ((isnull(r1.KolvoZ,0) - isnull(r1.KolvoN,0)) < 0) then 0
      else (isnull(r1.KolvoZ,0) - isnull(r1.KolvoN,0))
    end
  ),
  case when (dl.TypeFormulaID = 32) then dl.Rasf_
        else case when (dl.ZapasPolka < 0) then 0
                  else dl.ZapasPolka + case when isnull(CaID.Minss, 0) >
      ( case when (isnull(sal.SRSal_Act, 0) >= ISNULL(caID.primaryOrder,0)) then 0.000
        else isnull(caID.primaryOrder, 0) - isnull(sal.SRSal_Act, 0)
      end
      ) then ISNULL(caID.Minss, 0)
        else ( case when isnull(sal.SRSal_Act, 0) >= isnull(caID.primaryOrder, 0) then 0.000
                  else isnull(caID.primaryOrder, 0) - isnull(sal.SRSal_Act, 0)
                end
              )
        end
  end
end,
end,

```

Рисунок 3.12 – Алгоритм розрахунку Sold

В залежності від типу замовлення відрізняється спосіб формування його кількості, для прикладу вплив прогнозу продажів у відсотковому співвідношенні тощо (рис. 3.13).

```

IF @TypeFormula in (1,11,26)      -- Стелажка/Торти
  IF @InWay = 0
    SET @ResultVar = @ZapasPolka + @Sales - @Stores
  ELSE SET @ResultVar = @ZapasPolka + @Sales - (@Stores+@KolvoGoodsInWay)

IF @TypeFormula = 2              -- Скоропорт
  IF @InWay = 0
    SET @ResultVar = @ZapasPolka + @Sales - @Stores
  ELSE SET @ResultVar = @ZapasPolka + @Sales - (@Stores+@KolvoGoodsInWay)

IF @TypeFormula = 3              -- МинМакс
  IF @InWay = 0
    SET @ResultVar = @ZapasPolka - @Stores
  ELSE SET @ResultVar = @ZapasPolka - (@Stores+@KolvoGoodsInWay)

IF @TypeFormula in (4,22)        -- Стелажка
  IF @InWay = 0
    SET @ResultVar = @ZapasPolka + round((@Sales*@Coef),3) - @Stores
  ELSE SET @ResultVar = @ZapasPolka + round((@Sales*@Coef),3) - (@Stores+@KolvoGoodsInWay)

IF @TypeFormula in (5,12)        -- Стелажка (коэффициенты 0, 0, 0.5, 0.5)
  IF @InWay = 0
    SET @ResultVar = @ZapasPolka + round((@Sales*@Coef),3) - @Stores
  ELSE SET @ResultVar = @ZapasPolka + round((@Sales*@Coef),3) - (@Stores+@KolvoGoodsInWay)

IF @TypeFormula = 6              -- Скоропорт (коэффициенты 0, 0, 0.5, 0.5)
  IF @InWay = 0
    SET @ResultVar = @ZapasPolka + round((@Sales*@Coef),3) - @Stores
  ELSE SET @ResultVar = @ZapasPolka + round((@Sales*@Coef),3) - (@Stores+@KolvoGoodsInWay)

-- 04.07.2011
-- IF @InWay = 0
--   SET @ResultVar = @ZapasPolka + round((@Sales*@Coef),3)
--   ELSE SET @ResultVar = @ZapasPolka + round((@Sales*@Coef),3) - (@KolvoGoodsInWay)

-- 13.09.2011
IF @TypeFormula = 9              -- Стелажка ТЕСТ
  IF @InWay = 0
    SET @ResultVar = @ZapasPolka + @Sales - @Stores
  ELSE SET @ResultVar = @ZapasPolka + @Sales - (@Stores+@KolvoGoodsInWay)

IF @TypeFormula in (10,14,21)    -- ОВОЩИ
  IF @InWay = 0
    SET @ResultVar = @ZapasPolka + @Sales - case when @Stores<0 then 0 else @Stores end
  ELSE SET @ResultVar = @ZapasPolka + @Sales - (case when (@Stores+@KolvoGoodsInWay)<0 then 0 else (@Stores+@KolvoGoodsInWay) end)

```

Рисунок 3.13 – Види розрахунку кількості залежно від типу замовлення

А після відбується округлення отриманої кількості до розфасовки (рис. 3.14).

```

-- case when @TypeFormyla in (1001,1003) SET @ResultVar = case when @Rasf*
dbo.func_RoundRasf_LU_NEW (
    z_o.Rasf,
    z_o.KolvoPeriod,
    z_o.Store,
    z_o.TypeOrder,
    null,
    null,
    null,
    null,
    null ) end as Sold,
--

```

Рисунок 3.14 – Передача параметрів у функцію округлення до розфасовки

У функцію передаються значення фасування певного виду товару (наприклад вода фасується 6 пляшок у пак відповідно Rasf=6), KolvoPeriod – сформована кількість з використанням прогнозів, договорів, асортименту, інформації по періодам, Store – залишки, TypeOrder – тип замовлення.

І в залежності від типу замовлення використовується відповідний спосіб округлення замовлення (рис. 3.15).

```

SET @ResultVar = case when (@Rasf*(round(@ZakKolvoFirst/@Rasf,0) +
-- OAB 20160719 case when ( round(@ZakKolvoFirst/@Rasf,0)=0 and (@StoreKolvo< @Rasf*dbo.func_PercentRasf(@Rasf)) ) then 1 else 0 end)<0
    case when ( round(@ZakKolvoFirst/@Rasf,0)=0 and (@StoreKolvo< @Rasf*dbo.func_PercentRasf(@Rasf, 0)) ) then 1 else 0 end)<0
    then 0
    else (@Rasf*(round(@ZakKolvoFirst/@Rasf,0) +
-- OAB 20160719 case when ( round(@ZakKolvoFirst/@Rasf,0)=0 and (@StoreKolvo< @Rasf*dbo.func_PercentRasf(@Rasf)) ) then 1 else 0 end))
    case when ( round(@ZakKolvoFirst/@Rasf,0)=0 and (@StoreKolvo< @Rasf*dbo.func_PercentRasf(@Rasf, 0)) ) then 1 else 0 end))
    end
- IF @ResultVar=0 and @ZakKolvoFirst>0 and @TypeFormyla=4 SET @ResultVar=@Rasf
IF /*@ZakKolvoFirst>0 and*/ @TypeFormyla in (4,5,22/*,6*/ ,24) SET @ResultVar=case when (@Rasf*Round(Round(Round(Round((@ZakKolvoFirst/@Rasf)+0.0004,3)+0.004,2)+0.04,1)+0.4,0))<0
    then 0
    else @Rasf*Round(Round(Round(Round((@ZakKolvoFirst/@Rasf)+0.0004,3)+0.004,2)+0.04,1)+0.4,0)
    end

IF @TypeFormyla in (25) SET @ResultVar=case when @Rasf*(round(@ZakKolvoFirst/@Rasf,0)+case when (@ZakKolvoFirst - (@Rasf*round(@ZakKolvoFirst/@Rasf,0))) > @Rasf*0.3 then 1 else 0 end)<0
    then 0
    else @Rasf*(round(@ZakKolvoFirst/@Rasf,0)+case when (@ZakKolvoFirst - (@Rasf*round(@ZakKolvoFirst/@Rasf,0))) > @Rasf*0.3 then 1 else 0 end) end

IF @TypeFormyla in (27) SET @ResultVar=0.0

IF @TypeFormyla in (1/*,26*/) SET @ResultVar=case when (@Rasf*(round(@ZakKolvoFirst/@Rasf,0) + case when (@StoreKolvo+(@Rasf*round(@ZakKolvoFirst/@Rasf,0)))<0 then 1 else 0 end)<0
    then 0
    else (@Rasf*(round(@ZakKolvoFirst/@Rasf,0) + case when (@StoreKolvo+(@Rasf*round(@ZakKolvoFirst/@Rasf,0)))<0 then 1 else 0 end)) end

IF @TypeFormyla in (32) SET @ResultVar = case when @StoreKolvo< @Rasf*dbo.func_PercentRasf(@Rasf, 1) then @Rasf
    else 0 end
--новое округление для заказов PA,PI,PH
IF @TypeFormyla in (1001,1003) SET @ResultVar = case when @Rasf*(CEILING(@ZakKolvoFirst/@Rasf)) <=0
    then 0
    else @Rasf*(CEILING(@ZakKolvoFirst/@Rasf))
    end

```

Рисунок 3.15 – Округлення кількості у замовленні

Поетапне проходження по всьому процесу створення замовлень. Виконується запуск створення замовлення по певному складу та договору (рис. 3.16).

```
exec AUTOORDERTEST.dbo.proc_AutoOrder_RC 2725, 0, 0, 21,311929
```

Рисунок 3.16 – Створення замовлення на філіалі 2725 по договору 311929

У черзі сформувалося три завдання на створення замовлень (рис. 3.17), що говорить про те, що окрім основного будуть ще дочірні. Це можна перевірити за допомогою id завдання, idparent відповідає id батьківського замовлення.

DogovorID	FillID	DeltaForDelivery	DateToPost	DPeriod	DateLast	Queue	SapCounter	Created	Modified	Worker	TypeZakazStatus	ID	IDParent	IDQueue_Level
311929	2725	2	2021-10-23 00:00:00.000	7	2021-10-21 00:00:00.000	NULL	NULL	2021-10-21 13:31:24.737	2021-10-21 13:31:24.737	NULL	0	132592	NULL	85058
348506	3021	2	2021-10-22 00:00:00.000	10	2021-10-21 00:00:00.000	1	NULL	2021-10-21 13:31:24.720	2021-10-21 13:31:24.923	7	0	132590	132592	85059
348506	3021	2	2021-10-24 00:00:00.000	17	2021-10-21 00:00:00.000	2	NULL	2021-10-21 13:31:24.720	2021-10-21 13:31:24.923	4	0	132591	132592	85060

Рисунок 3.17 – Завдання на створення замовлень

Створення замовлень відбувається одночасно і завершується коли з'являється SapCounter – номер замовлення (рис. 3.18).

DogovorID	FillID	DeltaForDelivery	DateToPost	DPeriod	DateLast	Queue	SapCounter	Created	Modified	Worker	TypeZakazStatus	ID	IDParent	IDQueue_Level
311929	2725	2	2021-10-23 00:00:00.000	7	2021-10-21 00:00:00.000	3	P0000BIUK7	2021-10-21 13:31:24.737	2021-10-21 13:34:18.870	7	0	132592	NULL	85058
348506	3021	2	2021-10-22 00:00:00.000	10	2021-10-21 00:00:00.000	1	P0000BIUK6	2021-10-21 13:31:24.720	2021-10-21 13:32:04.350	7	0	132590	132592	85059
348506	3021	2	2021-10-24 00:00:00.000	17	2021-10-21 00:00:00.000	2	P0000BIUK5	2021-10-21 13:31:24.720	2021-10-21 13:32:04.333	4	0	132591	132592	85060

Рисунок 3.18 – Створені замовлення

Отже для чого потрібна така структура, перш за все це одночасний запуск створення кількох замовлень за рахунок черги, формування замовлень різних типів та можливість створення вкладених (дочірніх замовлень), що забезпечують можливість одразу формувати замовлення на подальше переміщення після надходження на основний склад. Тобто одразу формується ланцюг переміщень від постачальника на склад, після цього зі складу на наступний склад.

Детальніше на рисунку 3.19 зображено яку інформацію містить у собі сформоване замовлення.

Results Messages									
id	FilId	PostavshikID	Created	DateToPost	Modified	SapCounter	DogovorID	Summa	
18386595	2725	2482	2021-10-21 13:34:04.413	2021-10-23 00:00:00.000	2021-10-21 13:34:15.023	PO000BIUK7	311929	68434.200	

Рисунок 3.19 – Приклад сформованого замовлення

На рис. 3.19 можна побачити основні дані по замовленню – звідки і куди прямує замовлення (постачальник, філіал), дата створення та модифікації, дата поставки, договір, номер замовлення та розрахована вартість товару.

У детальній інформації (рис. 3.20) можна побачити товар, який містить у собі замовлення, його початкову сформовану кількість (kolvofirst) до того як було застосовано округлення до розфасовки (поле Rasf) і застосована автоматична корекція - перевірити її відпрацювання можна по полям Warning (замовлення без акційних товарів) і Marketingprog (замовлення з акційними товарами); і фінальну кількість – Kolvo.

LagerID	Kolvo	KolvoFirst	PricePurch	Rasf	StoreKolvo	SoldKolvo	Polka	GoodsInWay	Warning	MarketingProg	Action_id	ABC
777149	0.000	0.000	6.84000	12.000	29.000	16.000	5.040	0.000	NULL	Акционная автокоррекция	50297	NULL
777586	6.000	1.499	47.46000	6.000	11.000	8.109	4.390	0.000	NULL	NULL	NULL	B
777587	114.000	33.255	50.04000	6.000	6.000	111.923	5.460	0.000	Auto1 увеличение до минимума Auto4 > до срока годн	NULL	NULL	B
782787	0.000	0.000	41.70000	9.000	18.000	12.026	7.810	0.000	NULL	NULL	NULL	B
794121	0.000	0.000	50.40000	9.000	20.000	12.443	5.260	0.000	NULL	NULL	NULL	B
794122	4.000	0.536	22.68000	4.000	7.000	8.196	3.340	0.000	NULL	NULL	NULL	B
830219	20.000	5.730	12.66000	20.000	23.000	22.160	6.570	0.000	NULL	Акционная автокоррекция	50297	NULL
830220	0.000	0.000	12.66000	20.000	20.000	30.800	8.470	0.000	NULL	Акционная автокоррекция	50297	NULL
830221	8.000	5.260	27.00000	8.000	10.000	22.000	1.260	0.000	NULL	Акционная автокоррекция	25675	NULL
830222	8.000	1.015	27.00000	8.000	14.000	20.625	2.390	0.000	NULL	Акционная автокоррекция	25675	B
830225	12.000	10.923	29.64000	6.000	0.000	26.182	4.220	0.000	NULL	NULL	NULL	B
830226	0.000	0.000	26.83000	12.000	25.000	23.440	8.760	0.000	NULL	Акционная автокоррекция	50297	NULL
830227	0.000	0.000	25.14000	12.000	36.000	32.840	8.100	0.000	NULL	Акционная автокоррекция	50297	NULL
830228	12.000	4.620	24.83000	12.000	16.000	15.200	5.420	0.000	NULL	Акционная автокоррекция	50297	NULL
836195	12.000	6.730	15.48000	12.000	14.000	34.400	10.330	0.000	NULL	Акционная автокоррекция	50297	NULL
836196	0.000	0.000	11.64000	12.000	3.000	47.960	6.000	0.000	NULL	Акционная автокоррекция	49883	NULL
836540	0.000	0.000	13.02000	20.000	17.000	12.930	22.310	0.000	NULL	NULL	NULL	B
836541	0.000	0.000	12.36000	20.000	28.000	13.022	22.390	0.000	NULL	NULL	NULL	B
842306	0.000	0.000	6.84000	12.000	3.000	17.200	6.490	0.000	NULL	Акционная автокоррекция	50297	NULL
842307	12.000	0.000	6.84000	12.000	9.000	24.800	7.800	0.000	NULL	Акционная автокоррекция	50297	NULL
844049	0.000	0.000	15.12000	8.000	11.000	6.887	2.860	0.000	NULL	Акционная автокоррекция	NULL	NULL
844050	0.000	1.511	15.12000	8.000	6.000	2.991	4.520	0.000	Auto2 снижение до нормы	NULL	NULL	NULL
844051	0.000	0.000	24.30000	8.000	9.000	4.267	1.640	0.000	NULL	NULL	NULL	B

Рисунок 3.20 – Детальна інформація по замовленню

Для кожного товару відома ціна (PricePurch), прогноз продажів (SoldKolvo), залишки від минулих поставок (StoreKolvo), товар у дорозі (GoodsInWay) та залишки безпосередньо «на полицях» (Polka). Action_id вказує на те чи є акційний товар у замовленні. Також це ідентифікатор того, до якої саме акції він відноситься, детальніше це можна переглянути у відповідних таблицях (рис. 3.21).

activityId	activityName	dateFrom	dateTo
50297	ДА 27.10.21-22.11.21 молочка	2021-10-27	2021-11-22

Рисунок 3.21 – Приклад акції

Різні типи товарів потребують різних методів поповнення запасів, використовуючи ABC-аналіз (поле ABC), можна відокремити товари з великим оборотом (група А), які слід поповнювати частіше, і товари з низьким рівнем товарообігу (група В,С) [1].

Також важливою особливістю зовнішніх замовлень є те, що у них одразу розписано на які магазини поїде товар після отримання його складом від постачальника.

Наприклад замовлення PO000BIUK7 містить у собі певну загальну кількість по товарам (рис. 3.22).

```

27
28 select SapCounter,lagerid,kolvo from zakazauto z inner join zakazlinesauto za on z.id=za.zakazid where sapcounter='PO000BIUK7'
29
30
31

```

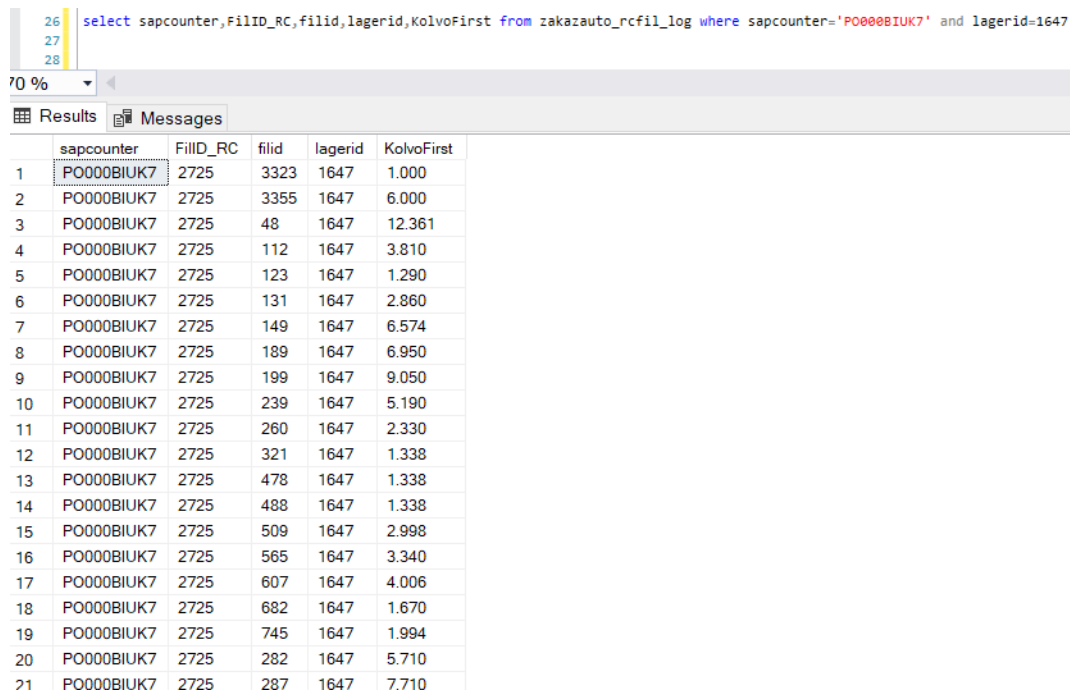
70 %

Results Messages

	SapCounter	lagerid	kolvo
1	PO000BIUK7	1647	228.000
2	PO000BIUK7	6084	168.000
3	PO000BIUK7	6300	0.000
4	PO000BIUK7	6809	552.000
5	PO000BIUK7	10836	60.000
6	PO000BIUK7	24759	264.000
7	PO000BIUK7	27484	0.000
8	PO000BIUK7	28336	0.000
9	PO000BIUK7	28741	0.000
10	PO000BIUK7	35993	0.000
11	PO000BIUK7	47302	0.000
12	PO000BIUK7	48717	0.000
13	PO000BIUK7	267295	0.000
14	PO000BIUK7	519276	360.000
15	PO000BIUK7	521335	48.000
16	PO000BIUK7	532792	36.000
17	PO000BIUK7	532801	60.000
18	PO000BIUK7	599428	0.000
19	PO000BIUK7	775693	0.000
20	PO000BIUK7	820001	12.000
21	PO000BIUK7	838888	96.000

Рисунок 3.22 – Кількість у замовленні по артикулам

І по певному артикулу можна продивитись, яка його кількість на який магазин (filid) поїде (рис. 3.23).



```

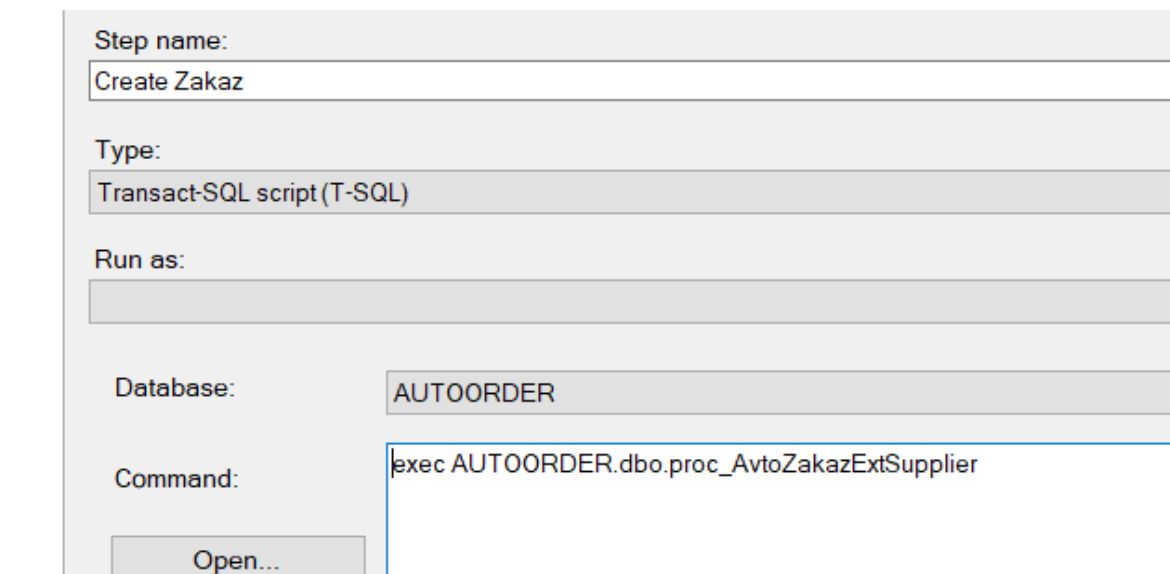
26 select sapcounter, FilID_RC, filid, lagerid, KolvoFirst from zakazauto_rcfil_log where sapcounter='PO000BIUK7' and lagerid=1647
27
28

```

	sapcounter	FilID_RC	filid	lagerid	KolvoFirst
1	PO000BIUK7	2725	3323	1647	1.000
2	PO000BIUK7	2725	3355	1647	6.000
3	PO000BIUK7	2725	48	1647	12.361
4	PO000BIUK7	2725	112	1647	3.810
5	PO000BIUK7	2725	123	1647	1.290
6	PO000BIUK7	2725	131	1647	2.860
7	PO000BIUK7	2725	149	1647	6.574
8	PO000BIUK7	2725	189	1647	6.950
9	PO000BIUK7	2725	199	1647	9.050
10	PO000BIUK7	2725	239	1647	5.190
11	PO000BIUK7	2725	260	1647	2.330
12	PO000BIUK7	2725	321	1647	1.338
13	PO000BIUK7	2725	478	1647	1.338
14	PO000BIUK7	2725	488	1647	1.338
15	PO000BIUK7	2725	509	1647	2.998
16	PO000BIUK7	2725	565	1647	3.340
17	PO000BIUK7	2725	607	1647	4.006
18	PO000BIUK7	2725	682	1647	1.670
19	PO000BIUK7	2725	745	1647	1.994
20	PO000BIUK7	2725	282	1647	5.710
21	PO000BIUK7	2725	287	1647	7.710

Рисунок 3.23 – Кількість у замовленні по артикулам на магазини

Якщо мова йде про замовлення на зовнішнього постачальника, тут все простіше, є відповідна процедура запуск якої відповідає за створення замовлень даного типу зі складу на постачальника - proc_AvtoZakazExtSupplier (рис. 3.24).



Step name:

Create Zakaz

Type:

Transact-SQL script (T-SQL)

Run as:

Database:

AUTOORDER

Command:

exec AUTOORDER.dbo.proc_AvtoZakazExtSupplier

Open...

Рисунок 3.24 – Запуск формування замовлень на зовнішнього постачальника

Так само як і у попередньому типі замовлення можна задавати параметри створення (рис. 3.25).

```
ALTER PROCEDURE [dbo].[proc_AvtoZakazExtSupplier]
@FilID int = 0 ,
@PostID int = 0,
@StatusZakaz int = 0,
@TypeFormula int = 0,
@DaysOff int = 0
```

Рисунок 3.25 – Вхідні параметри для створення замовлень на зовнішнього постачальника

А далі у цілому процес створення подібний до внутрішніх замовлень тільки без складної структури з використанням черг, завдань тощо.

3.3 Автоматична корекція замовлення

Важливим етапом формування замовлень є перевірка коректності його кількості на основі минулих продажів, прогнозу продажів і т.д., за це відповідає автоматична корекція. Призначення - автоматично збільшувати та зменшувати замовлення відповідно до норм оборотності [1].

Оборотність товару показує коли повернуться вкладені гроші з урахуванням прибутку, загалом знаючи швидкість оборотності, буде зрозуміло, коли потрібно замовляти топові позиції, які варто збувати за лояльними цінами, а за якими на підставі річної оборотності можна планувати закупівлі на наступний період.

Є наступні види правил для зміни кількості замовлення –

- Auto1 збільшення кількості до мінімуму. Підвищення до норми мінімальної оборотності: за певними типами договорів (молоко, м'ясо) виявляємо рядки замовлень, які менші за мінімальну норму оборотності та збільшуємо кількість до мінімальної оборотності (із округленням до цілої розфасовки);

- Auto2 зменшення кількості до норми. Видаємо попередження і не обрізаємо, якщо це товари власного виробництва та акції; якщо кількість менше

норми, а якщо додати одну розфасовку то буде більше норми; якщо продажів не було, якщо продажі за сьогодні набагато більше середніх продажів.

Обрізаємо якщо кількість більше максимальної оборотності і не потрапляє попередні пункти;

- Auto3 велика розфасовка - в залежності від минулих продажів товару збільшення або зменшення до норми;

- Auto4 збільшення замовлення, якщо дозволяє термін придатності товару. При оборотності замовлення менше норми оборотності - піднімати замовлення до мінімальної норми за умови, що нова норма оборотності менша за термін придатності.

Норма оборотності (НО) = $(\text{DayBTWDeliv} * (0.5 + K_ABC + K_Prov))$

Якщо $(\text{DayDeliv} * (0.5 + K_ABC + K_Prov)) > K_ABC$, то $\text{НО} = \text{MaxНО}$.

Де DayBTWDeliv – середня кількість днів між поставками,

K_ABC – коефіцієнт ABC по-артикульно,

K_Prov – коефіцієнт постачальника,

MaxНО – максимальна норма оборотності.

$\text{DayBTWDeliv} = 7 / \text{DayDeliv} * \text{Freq}$

Де DayDeliv – кількість днів між поставками,

Freq – частота замовлень.

$\text{MaxНО} = 0,5 * (\text{ExpDate}) - (7 / \text{OrderPerWeek}) * (\text{Freq})$.

Де ExpDate - термін придатності,

OrderPerWeek - кількість замовлень в тиждень.

Мінімальна Норма оборотності – базується на денній нормі оборотності по макрогрупі. Розрахунок мінімальної норми оборотності = Кількість днів від дати постачання до наступного постачання + мінімальна норма по макрогрупі та ABC.

Максимальна норма оборотності - = Кількість днів від дати постачання до наступного постачання + максимальна норма по макрогрупі та ABC [1].

Зміни, які застосувала до замовлень корекція можна переглянути у відповідній таблиці (рис. 3.26).

	ZakazID	LagerId	kolvoFil	kolvoAutoCor	remark
28	13869007	125793	18.000	36.000	Auto1 увеличение до нормы
29	13869007	144184	0.000	5.000	Auto1 увеличение до нормы
30	13869007	177553	0.000	10.000	Auto1 увеличение до нормы
31	13869007	339898	0.000	10.000	Auto1 увеличение до нормы
32	13869007	459190	0.000	18.000	Auto1 увеличение до нормы
33	13869007	504057	44.000	66.000	Auto1 увеличение до нормы
34	13869007	540217	0.000	20.000	Auto1 увеличение до нормы
35	13869008	32485	555.000	407.000	Auto2 уменьшение до нормы
36	13869008	32549	48.000	36.000	Auto2 уменьшение до нормы
37	13869008	35971	30.000	15.000	Auto2 уменьшение до нормы
38	13869008	97146	340.000	370.000	Auto1 увеличение до нормы
39	13869008	137851	0.000	2.000	Auto1 увеличение до нормы
40	13869008	142342	130.000	156.000	Auto1 увеличение до нормы
41	13869008	471849	0.000	20.000	Auto1 увеличение до нормы
42	13869009	335	30.000	15.000	Auto2 уменьшение до нормы
43	13869009	32490	5.000	10.000	Auto1 увеличение до нормы

Рисунок 3.26 – Зміна кількості у замовленні автоматичною корекцією

Якщо замовлення корегувалося у ньому про це також буде зберігатися відповідна відмітка (рис. 3.27).

	sapcounter	kolvo	kolvofirst	lagerid	warning
505	PO0023HRB3	10.000	10.850	819882	Auto3 большая расфасовка
506	PO0023HRB3	0.000	2.514	855533	Auto2 снижение до нормы
507	PO0023HRB3	0.000	2.290	856065	Auto2 снижение до нормы
508	PO0023HRB3	3.840	4.416	858667	Auto3 большая расфасовка
509	PO0023HRB9	6.000	0.903	795627	Auto1 увеличение до минимума
510	PO0023HRLB	0.000	11.137	658795	Auto2 снижение до нормы
511	PO0023HRLB	27.000	12.830	731390	Auto3 большая расфасовка
512	PO0023HRLB	0.000	12.057	798828	Auto3 большая расфасовка
513	PO0023HRLG	0.000	10.570	798831	Auto2 снижение до нормы
514	PO0023HRLH	81.000	93.560	541677	Auto2 снижение до нормы
515	PO0023HRLI	0.000	1.572	541679	Auto2 снижение до нормы
516	PO0023HRL0	0.000	2.270	662349	Auto2 снижение до нормы
517	PO0023HRL0	12.000	17.057	682859	Auto2 снижение до нормы

Рисунок 3.27 – Відмітка у замовленні про застосування корекції

Процес корекції замовлень автоматично запускається відповідним джобом кожного дня після формування замовлень всіх типів.

3.4 Контроль товарного запасу

Важливою складовою замовлення є дані про стан і кількість товару наявного безпосередньо на полицях та складі супермаркетів і яка кількість загалом туди може поміститися. За все це відповідає алгоритм контролю товарного запасу (КТЗ).

Першим етапом іде розрахунок середніх продажів. Далі Розрахунок Асортименту, розрахунок страхового запасу (SS) та розрахунок мінімальної викладки на полиці (МСП).

Кількість товару на полиці = Страховий запас (обов'язкова найменша кількість товару на складі) + ТЗ (товарний запас).

Розрахунки МСП (рис. 3.28) становлять алгоритм отримання значення МСП.

ID	Название расчета	Описание	Активность
1	ПАП новый	Последняя версия планогрaмм	☒
2	ПАП расчетный	Последняя версия планогрaмм №2	☐
3	ПАП СТМ	Сумируется к остальным расчетам	☒
4	ПАП Ручной	Пап Ручной	☒
5	МСК	Расчет на основании стандартной формулы McKinsey	☒
6	Прикасса	Суммируется к любому расчету, суммарное значение MSP по всей прикассовой зоне	☒
7	MacroMicroMSP	Автоматический коммерческий расчет MSP на базе продаж	☒
8	MSP 0	Нулевой MSP для макро по которым не нужен авторасчет	☒

Рисунок 3.28 – Розрахунки МСП

Методи МСП - це набори розрахунків у необхідному пріоритеті вибору, тобто коли буде обиратися необхідний МСП для філіалу та артикулу візьметься розрахунок у якого найвищий пріоритет у методі.

Для створення та керування методами передбачено відповідний інтерфейс (рис. 3.29).

The screenshot displays a complex interface for managing calculation methods (МСП). It includes several sections:

- Методы (Methods):** A table listing calculation methods with columns for ID, Name, Description, List of calculations, Activity, and Default. Methods include 'Стандарт' (Standard), 'McK' (McKinsey), and 'Ручной' (Manual).
- Расчеты в выбранном методе 1 - Расчеты МСП (Calculations in the selected method 1 - MSP calculations):** A table showing specific calculations for the selected method, including 'МСК' (McK), 'ПАП новый' (New PАП), 'ПАП расчетный' (Calculated PАП), and 'ПАП Ручной' (Manual PАП).
- Все существующие расчеты (All existing calculations):** A table listing all available calculations, including 'ПАП новый', 'ПАП СТМ', 'ПАП Ручной', and 'МСК'.
- 3. Разрешенные связи (3. Allowed connections):** A section for defining relationships between calculations.
- Определить добавляемый расчет в методе. 2. Расчет которые плюсятся (Define the calculation to be added to the method. 2. Calculations that are added):** A table for selecting which calculations to add to a specific method.
- Доступные добавляемые расчеты (Available calculations to be added):** A table listing calculations that can be added to the method.

Рисунок 3.29 – Інтерфейс для керування методами МСП

Сам метод визначається:

- розрахунком МСП у методі. У методи користувач додає розрахунок(и) (за допомогою інтерфейсу рис. 3.29). Якщо, наприклад, у методі МсК не порахується МСП, то з даного розрахунку, буде обрано розрахунок наступний за пріоритетом (у даному випадку ПаП новий). Користувач сам визначає пріоритет, виставляючи його у вікні «Розрахунок у вибраному методі»;

- розрахунки, які додаються. У цьому функціоналі закладено додавання розрахунків між собою, тобто якщо необхідно до Розрахунку МсК додати ПаП ручний і ПаП СТМ (як показано на прикладі) -

Метод МсК = Розрахунок МсК + ПаП Ручний + ПаП СТМ;

- дозволені зв'язки розрахунок + розрахунок. Для того щоб була можливість додавати розрахунок до розрахунку необхідно створити зв'язку (реалізовано для того, щоб заздалегідь визначати які розрахунки можна додавати до розрахунків, а які ні).

Методи SS - визначають значення страхового запасу (рис. 3.30).

ID ▲	Название SS	Описание	Активность	Тип SS	Дата последнего запуска	Дата создания
0	Empty SS	Будет всегда 0	☐	all	05.06.2020	05.06.2020 0:00:00
1	Shelf default	Существующий п...	☒	all	05.06.2020	05.06.2020 0:00:00
2	McK default	Существующий п...	☒	all	05.06.2020	05.06.2020 0:00:00
3	Нулевой SS	SS всегда будет р...	☒	all	01.01.1900	18.11.2020 0:00:00
4	SS без SSOFF	Способ расчета S...	☒	all	01.01.1900	08.12.2020 0:00:00

Рисунок 3.30 – Методи SS

Ознаки Ассортименту (рис. 3.31) - це запит, сутність яка визначає брати за цією ознакою дані з джерела чи ні, чи брати лише деякі.

Наприклад, ознака це - «наявність графіка замовлення 1 або 0», тобто якщо в правилі стоїть ознака в якому значення «1» то обрані артикули для цього правила повинні бути в джерелі з активним графіком замовлення.

ID	Название признака	Описание	Активность	Дата создания
1	Macr	Макро Группа	☒	10.02.2020
2	SAPstatus	САП статус	☒	12.02.2020
3	Bussines	Бизнес	☒	12.02.2020
4	McK	Маккинзи статус	☒	14.02.2020
5	Тип артикула	Типы всех Артикулов	☒	20.02.2020

Рисунок 3.31 – Ознаки Ассортименту

Правила Ассортименту. Для выбора необходимого ассортимента заостосовуються правила. Правила містять у собі ознаки, на підставі яких буде відібрано філіал та артикул.

У правилах визначається за якими ознаками буде набраний асортимент для певного правила, а також за якими саме ознаками, тобто наприклад, правило №1 в ознаках асортименту 5, значення ознаки встановлено «1» це означає, що будуть взяті тільки Артикули які потрапляють під категорію Звичайний товар (рис 3.32)

ИД правила	Наименование правила	Активность	McK Статус	Описание	дата начала	дата
0	Default_base	☒	☐	Default_base	09.04.2020	09.04.2020
1	Default_McK	☒	☒	default rull	10.04.2020	09.04.2020
2	Default_PaP	☒	☒	default papl	04.05.2020	09.04.2020

Выбранные признаки		Существующие признаки	
ИД признака	Название признака	ИД признака	Название признака
3	Bussines	1	Macr
4	McK	2	SAPstatus
5	Тип артикула	3	Bussines
6	Не сезонные товары	4	McK

Значение признаков			
Выбрать	Описание	Значение	Название
☒	я.с	я.с	я.с
☐		9	Расчет производство
☐		3	Сырье
☐		12	кас.лента
☐		6	услуга
☐		7	материалы
☒		1	Обычный товар
☐		10	Покеййд

Рисунок 3.32 – Работа с правилами ассортимента

Структура таблиці у якій створюються правила зображена на рисунку 3.33.

ИД правила	Наименование правила	Активность	McK Статус	Описание	дата начала	дата окончания
0	Default_base	☒	☐	Default_base	09.04.2020	09.04.2049
1	Default_McK	☒	☒	default rull	10.04.2020	09.04.2049
2	Default PaP	☒	☒	default oadl	04.05.2020	09.04.2049

Рисунок 3.33 – Правила асортименту

У таблиці (рис. 3.34) знаходяться всі ознаки, які працюють у правилі - Набір всіх правил + розрахунок.

ИД правила	Наименование правила	Активность
0	Default_base	☒
1	Default_McK	☒

Выбранные признаки	
ИД признака	Название признака
2	SAPstatus
8	Тип формулы заказа

Рисунок 3.34 – Набір ознак для правила з розрахунком

Набір всіх правил + розрахунок + встановлена ознака розрахунку відображена на рисунку 3.35.

ЕКТ	
☒	11 Продовольственные товары
☒	12 Непродовольст. товары
☐	6 Услуги

Рисунок 3.35 – Ознаки розрахунку

Комбінації ТЗ. Усі раніше описані сутності Правила, Розрахунки, Методи МСП, Методи SS складаються в одну комбінацію. На підставі складеної комбінації (а саме її дати зв'язку) визначається метод розрахунку, що належить комбінації (рис. 3.36)

Правила Ассортимента					Методы МСП		Методы SS		
ИД правила	Название ...	Описание	Количеств...	Выбрать	ИД метода	Название метод	ИД SS	Название SS	Тип SS
0	Default_base	Default_base	3601898	☐	1	Стандарт	1	Shelf default	all
1	Default_McK	default rull	1914168	☐	2	McK	2	McK default	all
2	Default_PaP	default papl	143752	☐	3	Ручной	3	Нулевой SS	all
3	Default_M...	МСК расче...	98855	☐	4	Только ПАП	4	SS без SSOFF	all

Созданные комбинации								
ИД связи	Описание связи	ИД правила	Название правила	ИД метода	Название метода	Дата связи	ИД метода SS	Названи
11	MSK без SSOFF (эл...	11	MSK_elit_4_макро	1	Стандарт	08.12.2020	4	SS без S
10	МСП = 0	10	МСП 0	12	мсп = 0	17.11.2020	1	Shelf def
9	ТРАШ	3	Default_McK_ТРАШ	3	Ручной	17.11.2020	2	McK defa
2	Новинка	2	Default_PaP	4	Только ПАП	22.04.2020	1	Shelf def

Рисунок 3.36 – Комбінації товарного запасу

Структура таблиці Комбінацій –

ID_Combo (int) - ІД комбінації

ID_Rull (int) - ІД правила

ID_Method (int) - ІД методу

ID_SS (int) - ІД страхового запасу

ComboName (nvarchar 100) - Ім'я комбінації

Created (datetime) - Дата час створення

Modified (datetime) - Дата час зміни

UserName (nvarchar 100) - Ім'я користувача

isDefault (bit) - Комбінація за замовчуванням, створена на основі існуючого принципу розрахунку

isActive (bit) - Тригер, показує активність поля. Якщо 1 статус активний, якщо 0 неактивний.

Автоматичний розрахунок товарного запасу -

- виконується розрахунок середніх продажів, із раніше зібраних джерел збираються та підраховуються дані;
- виклик процедури розрахунку асортименту - на даному етапі викликається з параметром 'all' в результаті будуть взяті всі артикули;
- виклик процедури розрахунку страхового запасу - на даному етапі викликається з параметром 'Shelf', в результаті будуть взяті всі артикули стелажної продукції;

- виклик процедури розрахунку МСП - на даному етапі викликається з параметром 'all' в результаті будуть взяті всі артикули;
- розрахунок параметра addplace.

Далі на підставі артикулів з першого етапу «Розрахунку середніх продажів» беремо набрані результати (філіал, артикул, МСП, страховий запас) та за формулою 3.3 розраховуємо значення МСП.

$$(SS) + \text{МСП} + (\text{addNorm} * \text{WeightedAVGDailySales}) + \text{addplace} + D3 \quad (3.3)$$

Де SS – страховий запас;

МСП – мінімальна викладка на полиці;

addNorm, D3 – дані з середніх продажів;

WeightedAVGDailySales – середні денні продажі.

Розрахунок Асортименту. Основний набір Асортименту, за яким необхідно проводити розрахунок, береться з даних за договорами, далі запускається цикл за правилами МСП, за умови що правила та зв'язки активні і дата закінчення правила більша за сьогоднішню дату.

У кожній ітерації циклу йде логування – записується результат з'єднань ознак у правилі, тобто один артикул і філіал повинні потрапляти у всі отримані множини кожної ознаки, яка перебуває в правилі, ті артикули яких не буде в одному з джерел не потрапить у дане правило.

Відбираємо унікальний асортимент (пріоритет: дата початку правила). З попереднього кроку беруться дані і з них вибираємо на підставі максимальної дати (для одного артикула) філіал та артикул.

І набір Артикул/Філіал/Правило відібраний на основі максимальної дати правила.

Далі приєднується отриманий унікальний асортимент/Правило і на підставі комбінацій та наявного асортименту виконується вибір методу розрахунку, який проводиться на підставі максимальної дати комбінації, в яку входить даний метод.

Процедура вибору ознаки для правила отримує загальний набір збігів філіал/артикул у ознаках отриманого правила, повертає сформований набір у вигляді таблиці. План виконання:

- вибір активних ознак із правила, за якими проходитиме цикл;
- набір всіх значень всіх ознак в одну таблицю;
- далі циклом вибирається значення ознаки і пишеться в тимчасову таблицю. Кожна наступна ознака повинна містити зв'язку Філіал/Артикул, якщо така зв'язка не знайдена вона не буде братися після проходження за всіма ознаками в правилі. Залишається набір який і буде повернутий з процедури.

Розрахунок страхового запасу (SS) (рис. 3.37), перед викликом самої процедури розрахунку страхового запасу йде збереження передрахованих даних. План виконання:

- відбираються активні ІД страхового запасу, після чого циклом викликається процедура отримання значення SS; отриманий результат записується в тимчасову таблицю;
- значення для страхового запасу відбираються на основі існуючого Асортименту та Комбінацій, а саме береться перший ID_SS з максимальною датою комбінації.

```

-- #MSP_PrepareSSGroup s
inner join (
    select a.filId ,a.LagerId ,c.ID_SS
    ,ROW_NUMBER() OVER (PARTITION BY filId ,LagerId ORDER BY c.Created desc) AS rn
    from [dbo].[MSP_Filial_Assortiment] a
    left join [dbo].[MSP_CombinationsCTZ] c on a.ID_Rull = c.ID_Rull
    and a.ID_Method = c.ID_Method and c.isActive = 1
) f on s.filId = f.filId and s.LagerId = f.LagerId and s.ID_SS = f.ID_SS

```

Рисунок 3.37 – Розрахунок страхового запасу

Ті значення SS, які не будуть знайдені будуть за замовчуванням дорівнювати 0.

Підготовлений результат записуємо значеннями SS. З таблиці беруться дані SS до розрахунку МСП в основній процедурі розрахунку МСП.

Процедура отримання SS повертає отриманий набір страхового запасу. У разі потреби можна додавати необмежену кількість джерел. Процедура отримує ІД і по ньому йде в певний запит, звідки вибирає Страховий запас/Філіал/Артикул і записує їх у тимчасову таблицю.

Розрахунок МСП вибирає значення МСП із переднабору даних чи процедури.

План виконання:

- набір необхідного асортименту з раніше підрахованої таблиці;
- далі маючи список джерел у кожному наборі функція визначає активний список;
- ІД_розрахунку (рис. 3.38).

```
Select ID_Calc FROM dbo.MSP_CalculationMethodCatalog
where isActive = 1 and Source in (@MethodSource, 'Source')
```

Рисунок 3.38 – Визначення активного списку ІД розрахунку

З отриманого результату формуємо тимчасову таблицю з унікальними Філіал/Артикул/ІД_МСП/МСП виходячи з таблиці комбінації методів беремо найвищий пріоритет з отриманої вибірки тобто з 5 припущених 3 пораховано, то беремо найпріоритетніший з порахованих.

Далі до отриманого результату приєднується запит із значеннями МСП, які необхідно додати до розрахунку (рис. 3.39).

```
FROM #MSP_PrepareCalculated as p
left join [dbo].[MSP_AddCalcMethodCatalog] as ad on p.ID_Calc = ad.ID_AddCalc
left join [dbo].[MSP_AddCalcAvailable] as av on p.ID_Calc = av.ID_AddCalc and av.ID_Calc = ad.ID_Calc
```

Рисунок 3.39 – Приєднання значень МСП

Після чого результат запишеться в таблицю MSP_PrepareCalculatedTotal, де збережені всі результати МСП.

3.5 Прогнозування даних та створення замовлення

Для можливості прогнозувати будь-який тип/набір даних у модуль прогнозування відповідно необхідно подати на вхід датасет на основі якого буде можливість аналізувати дані, знаходити закономірності та розраховувати майбутній прогноз продаж.

Приклад такого датасету зображено на рисунку 3.40.

YEAR	MONTH	BAKERY	CONFECTIONERY	ICE CREAM	BEVERAGES	CANNED, BOTTLED, FROZEN FOODS	MEAT	Clients	DELIVERS
2019	10	71 390	43 457	10 280	7 347	5 419	23 590	2 566	138
2019	11	56 101	33 658	7 264	6 845	4 760	18 333	1 066	223
2019	12	50 969	24 515	7 037	7 134	3 094	20 806	1 643	331
2020	1	59 505	35 391	8 682	5 192	4 387	24 121	1 923	285
2020	2	55 671	33 313	8 653	4 133	4 271	21 129	1 767	145
2020	3	61 621	37 256	12 047	4 223	6 392	28 251	1 750	190
2020	4	57 338	38 657	11 481	3 991	5 915	28 037	1 757	254
2020	5	58 264	36 658	11 036	3 516	6 733	27 505	1 972	1 414
2020	6	64 913	35 991	12 886	4 448	7 425	27 851	2 026	1 247
2020	7	56 212	39 022	11 874	6 168	11 083	24 016	1 662	783
2020	8	60 357	43 859	11 343	4 966	12 644	25 617	1 915	164
2020	9	71 380	45 248	10 856	6 025	11 764	26 876	1 559	142
2020	10	66 993	36 417	9 601	6 596	5 832	26 710	2 541	236
2020	11	57 963	31 547	8 116	6 023	5 244	21 874	2 267	124
2020	12	54 708	28 971	7 886	6 418	4 527	22 677	2 177	78
2021	1	58 718	34 712	9 179	4 255	5 271	27 005	2 189	97
2021	2	57 613	34 496	9 348	3 039	4 902	29 867	1 590	165
2021	3	68 970	39 869	13 015	4 430	7 047	37 729	2 116	137
2021	4	58 080	34 725	11 540	4 576	5 933	26 766	1 829	154
2021	5	53 790	35 848	13 291	3 513	6 712	23 691	2 407	206
2021	6	66 676	36 484	13 422	5 503	8 961	28 449	2 223	712
2021	7	63 019	37 018	11 618	5 370	9 882	25 254	1 999	183
2021	8	66 530	45 709	11 407	4 526	13 785	23 244	2 513	706
2021	9	83 682	44 060	10 634	4 896	14 233	21 277	3 702	736
2021	10	60 604	47 898	9 348	6 155	7 286	22 751	2 491	888

Рисунок 3.40 – Приклад датасету

У зображеному наборі даних міститься інформація за останні два роки (YEAR), по місяцям (MONTH) стосовно здійснених продажів певної продукції, у даному прикладі це – хлібобулочні вироби (BAKERY), кондитерські вироби (CONFECTIONERY), морозиво (ICE CREAM), напої (BEVERAGES), консерви, заморожені продукти (CANNED, BOTTLED, FROZEN FOODS), м'ясо (MEAT).

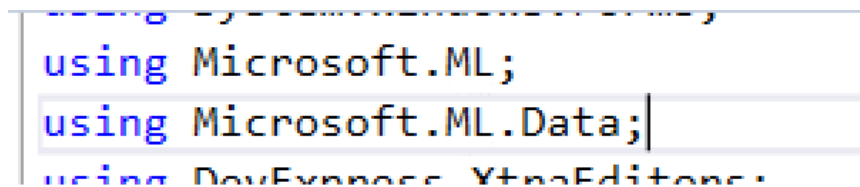
Окрім даних по продукції у датасеті міститься кількість покупців (Clients) та здійснених доставок замовлень (DELIVERS) за певний місяць.

У цілому можна сказати, що для можливості прогнозування необхідно надати дані для навчання по періоду на основі якого відбудеться аналіз даних, кількісні показники які братимуться за основу (за прикладом це дані по продажу продукції) та

у свою чергу, на які аспекти повпливали основні чинники (у прикладі продукція охопила відповідну кількість покупців та спричинила певну кількість замовлень на поставку товару). Також у результаті навчання є змога отримати прогнозні дані на те, яка кількість замовлень на поставку товару необхідна буде у майбутніх періодах.

Для реалізації функції прогнозування було використано ML.NET - бібліотека машинного навчання для мови програмування C# і не тільки. ML.NET надає можливість аналізу та прогнозування за допомогою машинного навчання з використанням моделей.

Аби мати можливість користуватися бібліотекою ML.NET необхідно підключити у коді певні бібліотеки (рис. 3.41).

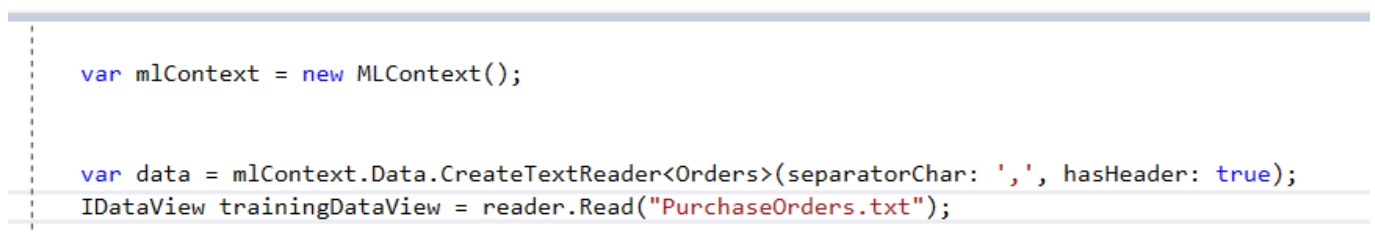


```
using Microsoft.ML;
using Microsoft.ML.Data;
using DevExpress.YtraEditors;
```

Рисунок 3.41 – Підключення бібліотек ML

Принципи роботи з даною бібліотекою. Для роботи з датасетом у коді необхідно визначити структуру даних, яка відповідатиме початковому набору даних.

Наступним кроком буде створення середовища для роботи та отримання даних з датасету (рис. 3.42).



```
var mlContext = new MLContext();

var data = mlContext.Data.CreateTextReader<Orders>(separatorChar: ',', hasHeader: true);
IDataView trainingDataView = reader.Read("PurchaseOrders.txt");
```

Рисунок 3.42 – Налаштування середовища та отримання даних

Подальшими кроками будуть приведення даних до відповідного типу, присвоєння значень з датасету раніше створеній структурі даних та вибір методу навчання. Для навчання використовується відповідна команда (рис. 3.43).

```
var model = learning.Fit(trainingDataView);
```

Рисунок 3.43 – Навчання моделі на основі набору даних

Варто зауважити, що існує декілька видів навчання -

- Sentiment Analysis – класифікує дані у двох категоріях (позитивна/негативна), використовується, якщо необхідно передбачити бінарний результат;

- Issue Classification – якщо результат передбачення не бінарний, а може бути у більше ніж двох категоріях;

- Price Prediction - даний сценарій передбачає числове значення ґрунтуючись на заданих даних (наприклад датасет продажу продукції (рис.3.40)) ;

- Custom scenario - дозволяє створювати користувацькі моделі. Даний сценарій підійде, якщо перші три сценарії не можуть вирішити поставлене завдання.

У системі формування замовлення та прогнозування використовується 3 вид навчання - Price Prediction.

У подальшому у навчену модель можна передавати певні набори інформації і на основі знань отриманих з навчання вона робитиме те або інше прогнозування.

3.6 Отримання даних з інших серверів

Існує декілька способів отримання даних з інших серверів, наприклад за допомогою використання реплікацій - процесів підтримки двох (чи більше) наборів даних у узгодженому стані; процес зміни одного набору даних (репліка), у відповідь на зміни іншого основного набору даних, але зовсім не обов'язково в результаті набори будуть однаковими оскільки в залежності від об'єму, навантаження на сервери тощо процес оновлення даних може розтягуватися на години, що часто є не прийнятним. Окрім цього реплікації є неконтрольованим процесом. Альтернативами можуть бути linked servers (зв'язані сервери) або ssis-пакети. Оскільки зв'язані сервери мають такі недоліки як залежність продуктивності і стабільності роботи від

віддаленості серверів було прийнято рішення зупинитися на використанні ssis-пакетів при налаштуванні отримання даних з інших систем.

Наприклад для оновлення інформації по двом типам залишків товару на складі (дві таблиці) зі складської системи джоб кожного дня запускає об 9, 11 та 15 ssis пакет TransitGold_IN_OUT.CopyVegetShortageStores, який реалізовує вивантаження даних у таблиці бази даних по замовленням (рис. 3.44).

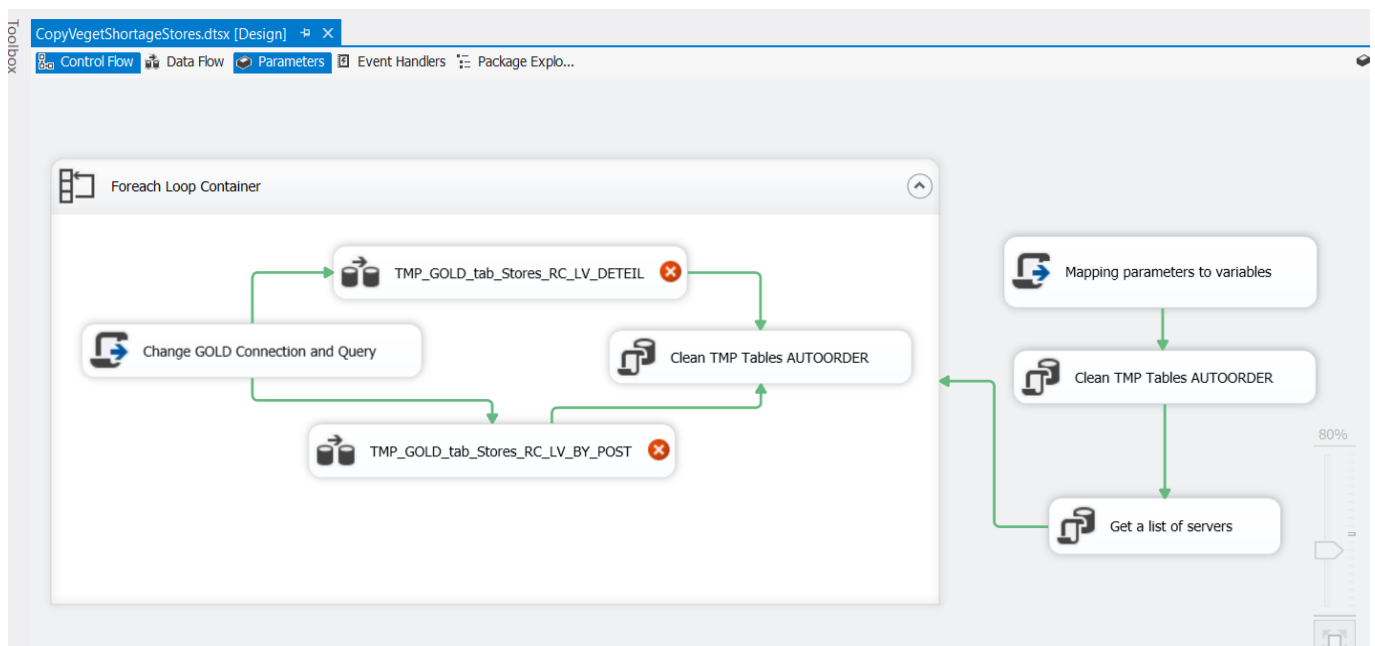


Рисунок 3.44 – Пакет CopyVegetShortageStores

У даному пакеті спочатку організовано підготовчий етап - першим кроком прописаний скрипт, який використовується для забору даних і в якому визначено, які поля необхідно отримати. Далі відбувається очищення темпових таблиць у кінцевій базі даних (процес отримання даних працює спочатку з проміжними таблицями, а потім вже дані потряпляють у кінцеві) та отримання адрес серверів з яких і буде отримання даних.

Отримавши базу даних з серверу отримання даних, кроком TMP_GOLD_tab_Stores_RC_LV_DETEIL відбувається підключення і виконується запит на отримання даних, які далі конвертуються у потрібний формат і записуються у проміжну таблицю (рис. 3.45).

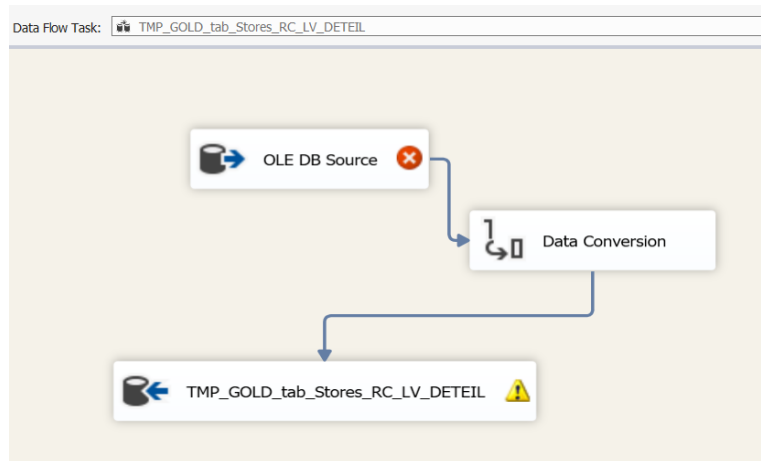


Рисунок 3.45 – Збереження даних у проміжну таблицю

І далі кроком Clean TMP Tables AUTOORDER викликається процедура `proc_CopyVegetShortageStores`, яка переносить дані з проміжних таблиць до основних, очищаючи при цьому перші.

Наповнення другої таблиці відбувається по такому ж принципу.

3.7 Розробка інтерфейсу користувача

Інтерфейс користувача розроблений з використанням .Net Core та WinForms. Проєкт у MS Visual Studio має структуру зображену на рисунку 3.46.

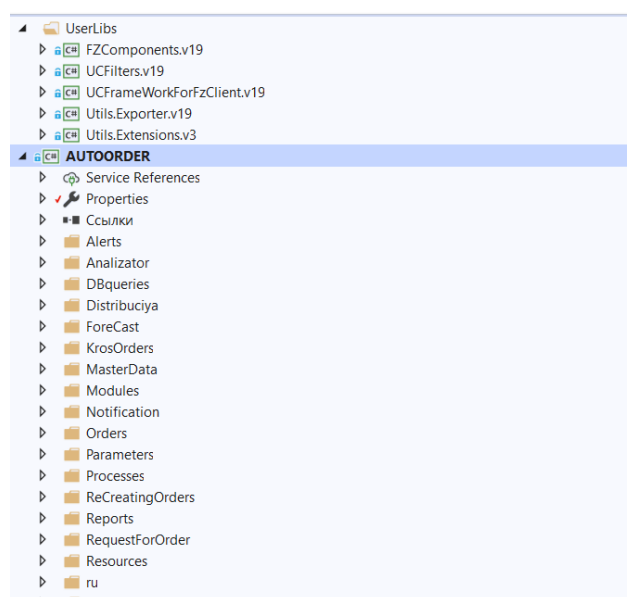


Рисунок 3.46 – Структура проєкту у MS Visual Studio

До нього входять форми і власні бібліотеки, у яких для прикладу закладено параметри для створення фільтру для форм (рис. 3.47).

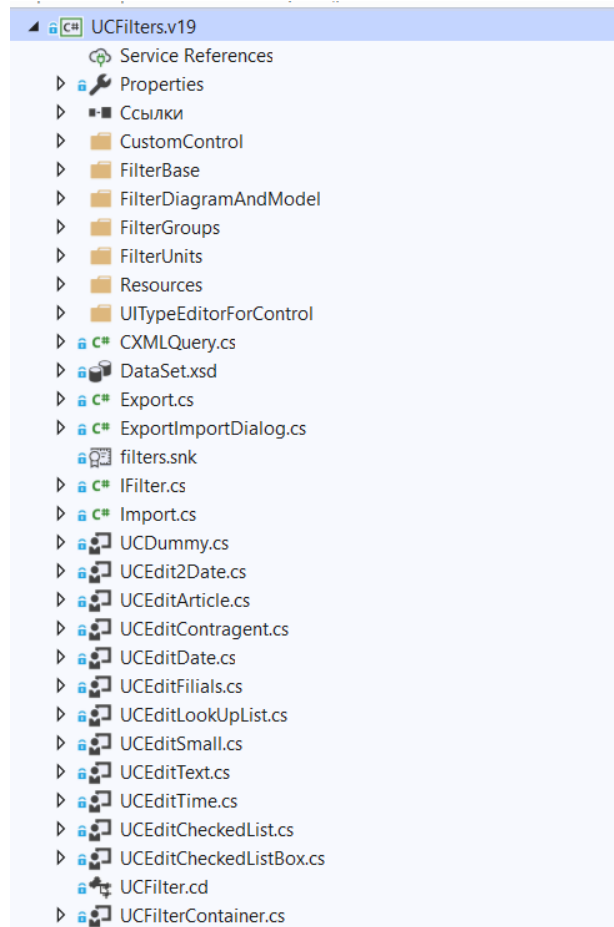


Рисунок 3.47 – Структура користувацьких бібліотек

Є бібліотеки що відповідають за обмеження прав, формують об'єкти для форм, реалізують імпорт та експорт даних тощо.

У цілому кожен звіт має власну форму (рис. 3.48), але схожий функціонал (наприклад перегляд замовлень зовнішніх постачальників, внутрішніх та зовнішніх клієнтів) реалізований через одну.

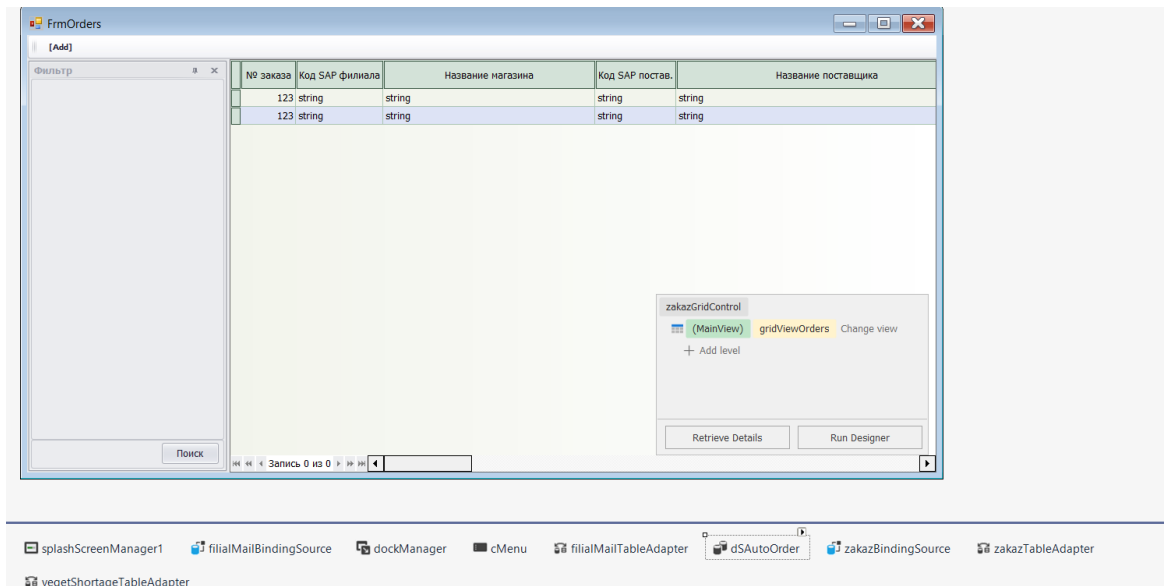


Рисунок 3.48 – Приклад форми з компонентами

Для наповнення форми використовуються TableAdapter (рис. 3.49), завдяки якому можна підключатися до бази даних і отримувати звітні дані та використовувати вбудовані запити на видалення, оновлення та зміну даних.

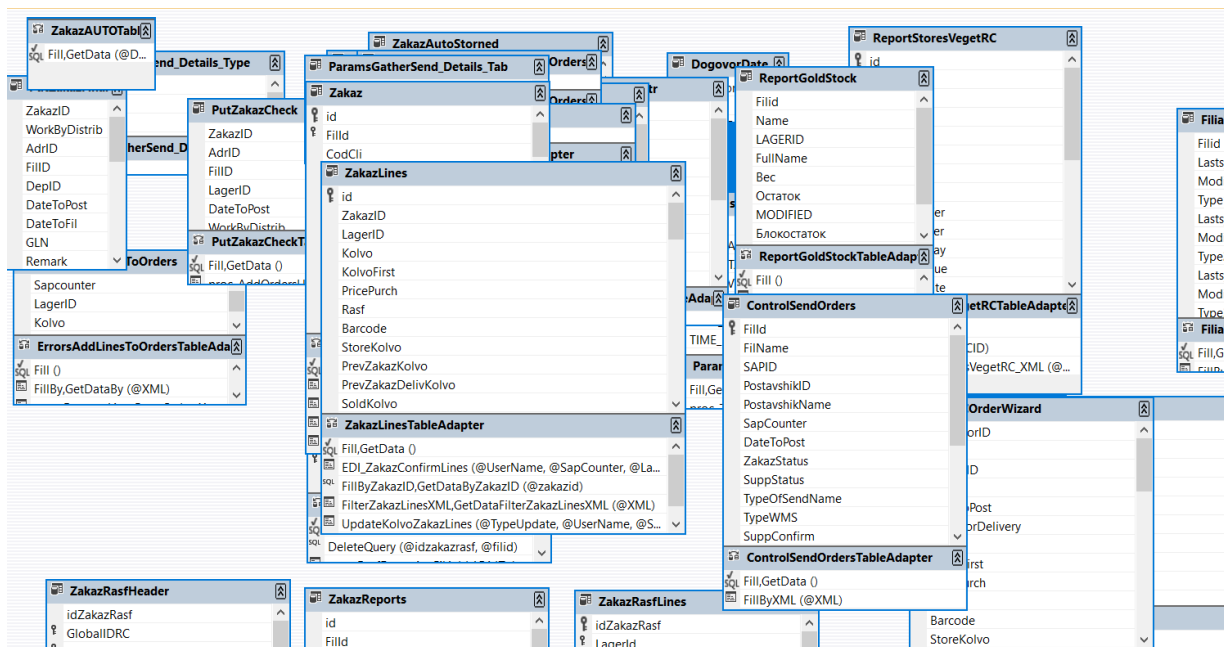


Рисунок 3.49 – TableAdapter-и проекту

Для отримання з коду доступу до процедур SQL використовуються наступні подібні функції, в яких відкривається потік і формується xml файл (рис. 3.50) з

заданими параметрами, які необхідно передати в процедуру і яка у кінці викликається. Процедура відпрацьовує і видозмінює одразу дані, які користувач бачить на формі.

```

//D:\Work\1
private void ThreadWorkRoundToCars_onTreadBackgroundWorker(object sender, DoWorkEventArgs e)
{
    var ms = new MemoryStream();
    var writer = new XmlTextWriter(ms, System.Text.Encoding.Default);
    writer.WriteStartDocument();
    writer.WriteStartElement("PARAM");
    writer.WriteAttributeString("UserName", Security.DataToClient.session.login);
    writer.WriteAttributeString("KolvoPallet", _countCars);

    foreach (var sapcounter in _sapcounters)
    {
        writer.WriteStartElement("SAPCounter");
        writer.WriteAttributeString("Value", sapcounter);
        writer.WriteEndElement();
    }
    writer.WriteEndElement();
    writer.Close();
    var queryXml = System.Text.Encoding.Default.GetString(ms.ToArray());

    _processOrdersResultDt = new DSAutoOrder.ZakazDataTable { PrimaryKey = null };

    var i = 0;
    while (true)
    {
        if (_processOrdersResultDt.Columns[i] != _processOrdersResultDt.SapCounterColumn && _processOrdersRes
        {
            _processOrdersResultDt.Columns.RemoveAt(i);
        }
        else i++;
        if (_processOrdersResultDt.Columns.Count == 2) break;
    }
    zakazTableAdapter.proc_RoundOrdersToCars(_processOrdersResultDt, queryXml);
}

```

Рисунок 3.50 – Формування xml-файлу і передача його у процедуру

У цілому для функціонування форм реалізовано подібна структура з взаємодією з процедурами або робота з даними відбувається безпосередньо у Visual Studio.

Додатково варто сказати про підключення до серверу з базою даних, яке відбувається певним алгоритмом з використанням сталих методів (рис. 3.51).

```

public static void ExecuteProcedureXml(string procedureName, string xml, ref DataTable dtResult)
{
    var connString = Properties.Settings.Default["AUTOORDERTEST|ConnectionString"].ToString();

    var conn = new SqlConnection(connString);
    var command = new SqlCommand(procedureName, conn) { CommandType = CommandType.StoredProcedure };
    command.Parameters.Add("@data", SqlDbType.VarChar).Value = xml;

    var da = new SqlDataAdapter(command);
    try
    {
        conn.Open();
        da.Fill(dtResult);
    }
    finally
    {
        conn.Close();
        da.Dispose();
    }
}

```

Рисунок 3.51 – Налаштування підключення до бази даних

Також якщо є необхідність, завдяки засобам ado.net можна прописувати безпосередньо запити (рис. 3.52).

```

this._adapter.TableMappings.Add(tableMapping);
this._adapter.UpdateCommand = new global::System.Data.SqlClient.SqlCommand();
this._adapter.UpdateCommand.Connection = this.Connection;
this._adapter.UpdateCommand.CommandText = @"UPDATE          ZakazLinesAUTO
Kolvo = ISNULL(ZakazLinesAUTO.Rasf * CEILING(((CASE WHEN @KolvoA < 0 THEN 0 ELSE @KolvoA END) + (CASE WHEN @KolvoAF < 0 THEN 0 ELSE @KolvoAF END)) / ZakazL
KolvoNanAction = CASE WHEN @KolvoAF < 0 THEN 0 ELSE @KolvoAF END, KolvoAction = CASE WHEN @KolvoA < 0 THEN 0 ELSE @KolvoA END,
KolvoFirst = CASE WHEN @KolvoA < 0 THEN 0 ELSE @KolvoA END + CASE WHEN @KolvoAF < 0 THEN 0 ELSE @KolvoAF END
ZakazAUTO INNER JOIN
ZakazLinesAUTO ON ZakazAUTO.id = ZakazLinesAUTO.ZakazID
(ZakazLinesAUTO.id = @id) AND (ZakazLinesAUTO.LagerID = @LagerID) AND (ZakazAUTO.Status <> 3) AND (ZakazAUTO.RealTypeZakaz <> 10) OR
(ZakazLinesAUTO.id = @id) AND (ZakazLinesAUTO.LagerID = @LagerID) AND (ZakazAUTO.Status <> 4)";
this._adapter.UpdateCommand.CommandType = global::System.Data.CommandType.Text;
this._adapter.UpdateCommand.Parameters.Add(new global::System.Data.SqlClient.SqlParameter("@Kolvo", global::System.Data.SqlDbType.Decimal, 5, global::System.Data.
this._adapter.UpdateCommand.Parameters.Add(new global::System.Data.SqlClient.SqlParameter("@id", global::System.Data.SqlDbType.Int, 4, global::System.Data.Paramet
this._adapter.UpdateCommand.Parameters.Add(new global::System.Data.SqlClient.SqlParameter("@LagerID", global::System.Data.SqlDbType.Int, 4, global::System.Data.Pa
this._adapter.UpdateCommand.Parameters.Add(new global::System.Data.SqlClient.SqlParameter("@KolvoA", global::System.Data.SqlDbType.Decimal, 5, global::System.Data
this._adapter.UpdateCommand.Parameters.Add(new global::System.Data.SqlClient.SqlParameter("@KolvoAF", global::System.Data.SqlDbType.Decimal, 5, global::System.Dat

```

Рисунок 3.52 – Приклад запиту до бази даних

І як розглядалося вище є змога користуватися власними бібліотеками для раціоналізації коду, приклад, створення фільтрів. З використанням заданих алгоритмів можна задати якого формату буде фільтр, його структуру, доступи і які дані вже будуть відібрані там одразу після завантаження (рис. 3.53).

```

public FrmOrdersUcFind()
{
    SetGroupContainerCaption(navBarGroupContainer.ViewGroup1, "Глобальные параметры", false);
    SetGroupContainerCaption(navBarGroupContainer.ViewGroup2, "Документ", false);
    SetGroupContainerCaption(navBarGroupContainer.ViewGroup3, "Даты", false);
    SetGroupContainerCaption(navBarGroupContainer.ViewGroup4, "Основной", true);

    UcDataProvided.ucEditDateStart.Value = DateTime.Today;
    UcDataProvided.ucEditDateEnd.Value = DateTime.Today;

    UcDataProvided.ucEditDateStart.HeadXmlQuery = new HeadXmlQuery { Tag = "@FILIAL", StartWhere = " and ", ViewTag = "0" };
    UcDataProvided.ucEditDateStart.LinesXmlQuery = new LinesXmlQuery
    {
        ColumnName = "Zakaz.Proved",
        Type = "DATE",
        Znak = ">=",
        If = " and ",
        Item = null
    };
    UcDataProvided.ucEditDateEnd.HeadXmlQuery = new HeadXmlQuery { Tag = "@FILIAL", StartWhere = " and ", ViewTag = "0" };
    UcDataProvided.ucEditDateEnd.LinesXmlQuery = new LinesXmlQuery
    {
        ColumnName = "Zakaz.Proved",
        Type = "DATE",
        Znak = "<",
        If = " and ",
        Item = null
    };
};

```

Рисунок 3.53 – Налаштування відображення компонентів фільтру

У даному прикладі задано структуру фільтру (рис. 3.54), а саме чотири блоки - глобальні параметри, документ, дати та основний. І в кожен блок додається необхідний компонент на рисунку це дата початку і дата кінця, а далі безпосередньо прописується принцип фільтрування для процедур запитів вже у базі даних.

Рисунок 3.54 – Фільтр

У результаті роботи з компонентами бібліотек отримуємо фільтр на рис. 3.54.

3.8 Робота користувачів з системою

Варто зазначити, що робота з інтерфейсною частиною перш за все починається з авторизації і відповідно до цього користувач зможе користуватися лише тою частиною функціоналу, яка доступна для його ролі, певні категорії менеджерів можуть працювати з певними видами замовлень, фахівці зі складів тільки з замовленнями своїх підрозділів тощо.

Аби користувачі мали змогу працювати з замовленнями використовується додаток. Робота з замовленнями можлива з використанням вкладки Замовлення (рис. 3.55).

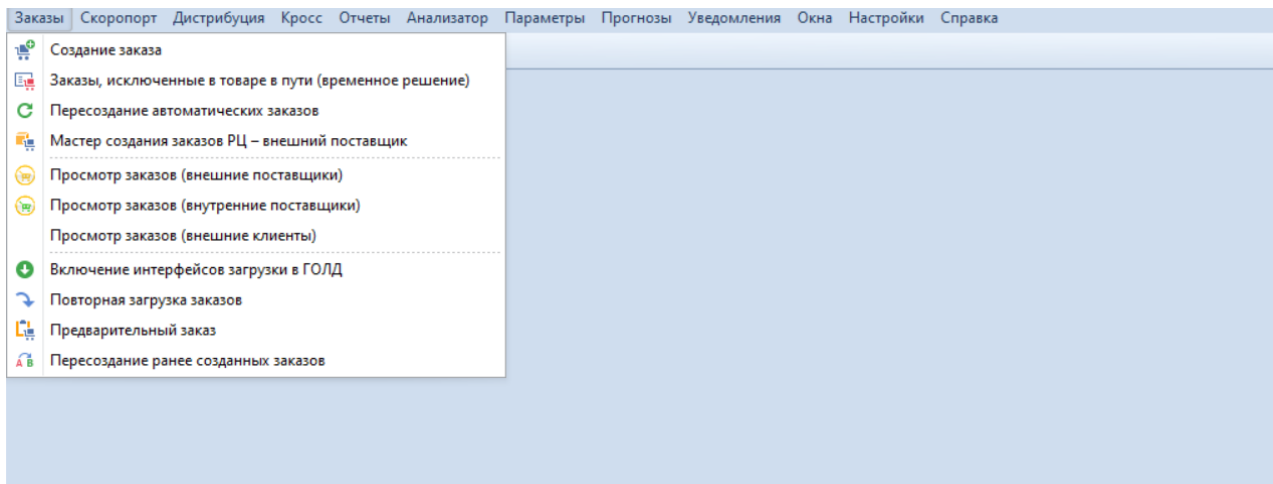


Рисунок 3.55 – Вигляд програми після авторизації

У меню «Замовлення» у ПЗ передбачена можливість створення замовлення, додавання замовлень, перестворення автоматичних замовлень, створення замовлень на зовнішнього постачальника тощо.

Крім того, в цьому розділі меню доступний перегляд вже існуючих замовлень на зовнішнього (рис. 3.56) та внутрішнього постачальника, а також зовнішнього клієнта.

Просмотр заказов (внешние поставщики)

Действия

Фильтр

Глобальные параметры

Документ

№ заказа:

№ SAP заказа: PO000BLD81

Статус заказа:

Статус отпр.:

Тип расчета:

Тип заказа:

Тип логистик.:

Главный заказ:

Даты

Проведен: с до

Сторнирован: с до

Запланирована поставка: с до

Основной

№ заказа	Код SAP филиала	Название магазина	Код SAP постав.	Название поставщика	Дата создания	Дата поставки	Дата постав
18371890	SO2U	С ХАРЬКОВ Пушанская ул, 2	0024364528	ІДС ПрАТ	20.10.2021 11:40:35	21.10.2021 00:00:00	

Рисунок 3.56 – Перегляд замовлення

Меню «Дії» (рис. 3.57) на закладці включає варіанти дій, які можна зробити над замовленням.

Просмотр заказов (внешние поставщики)

Действия

Действие	Код SAP постав.	Название поставщика
Шортедж по А Б С категории		
Заливка цен в заказы	0000377511	Карлсберг Украина ПрАТ
Заливка количества в заказы	0038249784	Торнадо-Захід ПП
Заливка Времени поставки в заказы	0024364528	ІДС ПрАТ
Работа с импортом	AZA0000104	Azuba Polska Sp. z o.o.
Округлить до машин...	0035871504	БАЯДЕРА ЛОГСТИК ТОВ
Отправка поставщику	0035479380	Хмельницьк-млин ТОВ
(EDI) Добавить подтверждение кол-ва поставщиком	0035591588	Маревен Фуд Европа ТОВ
Сбор выбранных заказов	0036038348	АСТРЕЯ-7 ТОВ
Ручной запуск автозаказа	0032704716	Ольховый ЧП
Отправить заказ в G.O.L.D.	0036064430	Дагама ПП
Утвердить изменения по выделенному	0013745606	ТРИ СТАР ТОВ
Сторнировать	0034845075	ІМПЕРОВО ФУДЗ ТОВ
Указать причину заказа	0000382220	Монделіс Україна ПрАТ
Установить статус - Редактирование шортеджем	0035233190	Еконія ПШ ТОВ
Удалить статус - Редактирование шортеджем	0039668439	ДИСПЛЕЙ УКРАЇНА ТОВ
Объединение заказов	0038737621	БЕВЕРІДЖ ТРЕЙДІНГ КОМПА
Поиск	0022430008	САНДОРА ТОВ
Экспорт	0032967502	Лагода КФ ПрАТ
Настройки	0034899754	БІКО КОМПАНІЯ ТОВ
	0038416454	Прайм-П ТОВ
	0021360263	ВЕНД ПП
	0034845075	ІМПЕРОВО ФУДЗ ТОВ
	0000377163	ЛУЦЬК ФУДЗ ПрАТ

Рисунок 3.57 – Варіанти дій у меню «Дії»

Ручне створення замовлень можливе через завантаження mdb файлу і вибору у меню типу постачальника та замовлення (рис. 3.58).

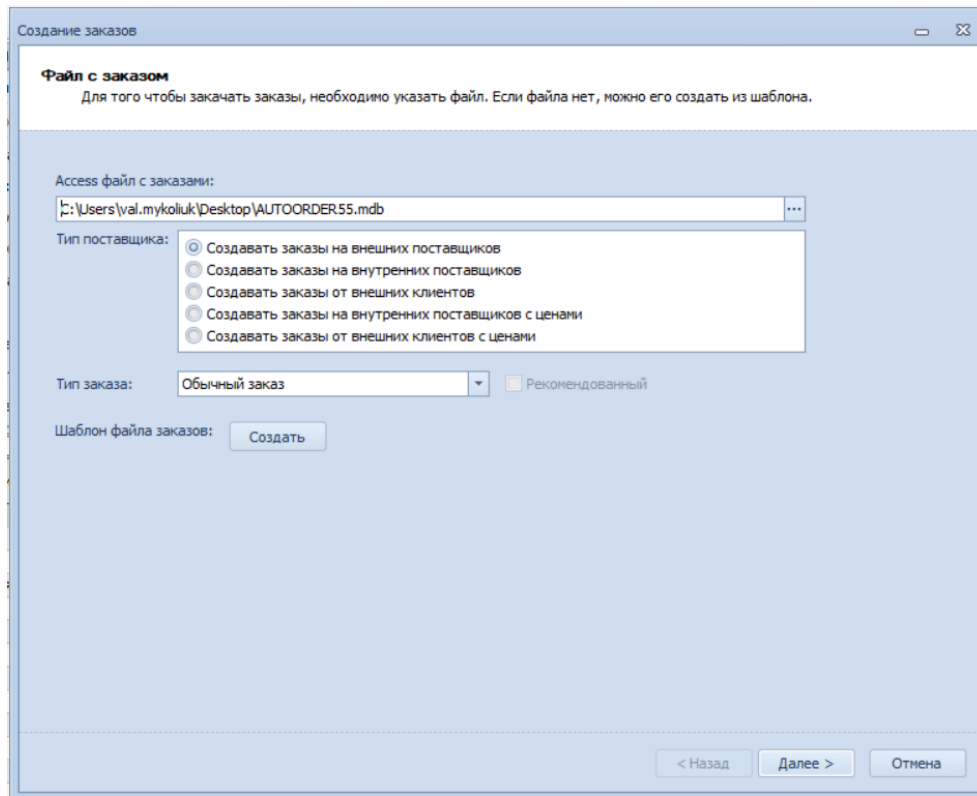


Рисунок 3.58 – Завантаження mdb-файлу із замовленнями

Якщо mdb-файл створено коректно, після натискання на кнопку «Далі» відкриється форма замовлення та підтвердження коректності даних.

Щоб продовжити формування замовлення в базі необхідно активувати умову «Дані коректні» (рис. 3.59).

Щоб заборонити надсилання замовлення постачальнику необхідно активувати умову «Не надсилати» (рис. 3.59).

Щоб дозволити надсилання замовлення на виробництво необхідно активувати «Замовлення на виробництво» (рис. 3.59).

Щоб заборонити вивантажувати замовлення в складську систему необхідно активувати «Не вивантажувати» (рис. 3.59).

Создание заказов

Просмотр заказов в файле
Просмотрите заказы и подтвердите корректность данных.

Заказ	Код причи...	Причина	Код поставщ...	Название поставщика	Код филиала	Название филиала
1	1	Распределение ДК	2251	МХП ПРАТ	452	БЛОК С КИЕВ Омелян.-Павленка

Orders_Orders Lines

#заказа	Lager ID	Lager Name	Kolvo	Kolvo Action	Sold Kolvo	Store Kolvo	Goods In Way	Price Wit Tax
1	461800	КуриноЕКг...	13,400	0,000	0,00	0,00	0,00	144,50

☒ Данные корректны
☒ Не отправлять поставщику (для заказов без признака MI)
☒ Заказ на производство
☒ Не выгружать в ГОЛД/Рубикон

Найдено заказов 1 позиций 1

Рисунок 3.59 – Перегляд замовлень у файлі та підтвердження коректності даних

Після завершення створення замовлення у формі з'явиться інформація про кількість оброблених та створених замовлень (рис. 3.60).

Создание заказов

Добавление заказов
Подождите, идет формирование заказов в базе.

Обработка заказов 2 из 2 Создано 2 ошибок 0

Обработка строк заказа обработано 1

Создание заказов закончено без ошибок. Для продолжения нажмите Далее.

Рисунок 3.60 – Завершення обробки замовлення

При формуванні замовлення можуть виникнути такі помилки (рис. 3.61):

- «Немає прав створювати. Філіал: [номер філіалу]» – у користувача немає прав створювати замовлення в рамках конкретного департаменту;
- «Філіал не існує» – вказано неіснуючий філіал;
- «Постачальник не існує» – вказано неіснуючого постачальника;
- «Вихідне замовлення не знайдено: [SapCountOrig]» – при перезаливці замовлення не знайдено оригіналу замовлення;
- «Не вказано номер акційного тижня» – не вказано номер акційного тижня у СВП та In-Out (види акційних замовлень) замовленнях;
- «Не знайдено активний контракт» – не знайдено договору між зовнішнім постачальником і філіалом;
- «Необхідно заповнити SAPCountOrig або DogovorIDRC для транзитного замовлення» – при заливанні транзитного замовлення не вказано номер договору або № SAP старого замовлення;
- «Замовлення '+ @SAPCountOrig +' не знайдено» – не знайдено номер замовлення;
- «У договорі не вказано транзитного складу для філіалу» – у транзитному замовленні вказано номер договору, але не вказано транзитний склад (поле заповнюється у договорі) ;
- «Не вказано договір» – не вказано номер договору замовлення зовнішніх клієнтів (як із ціною, так і без ціни) ;
- «Немає графіка поставок на склад» – у договорі немає графіка поставок зовнішніх постачальників;
- «Неправильна дата поставки. Дата має бути більшою» – у договорі є графік поставок зовнішніх постачальників, але дата поставки в договорі більша за дату поставки, зазначену в mdb-файлі;
- «Дата філіалу менша дати поставки» – у замовленні зовнішнього клієнта дата поставки на філіал менша дати поставки;

- «Не вказано номер постачання» – у замовлення на імпорт на зовнішнього постачальника не додано коментар;
- «Має бути зазначена 1-на ціна для артикула:[артикул]» – у замовленні з ціною не збігається ціна з ПДВ та ціна без ПДВ;
- «Артикул не знайдено в договорі. [Артикул]» – зазначений артикул, якого немає в активному договорі у всіх типах замовлень із зовнішніми та внутрішніми постачальниками, крім типів замовлення «Замовлення без асортименту», «Замовлення за надлишками», «Повторне переміщення», «Попереднє замовлення», «Транзит через склад», «Ручне замовлення на роздріб» ;
- «Вказана акція: Action_id для артикула не знайдена. Артикул: [артикул]» – у замовленні на зовнішнього постачальника вказано номер акції, але він не знайдений у календарі активностей;
- «Замовлена кількість не кратна розфасовці. Артикул: [артикул]» – у замовленні на зовнішнього постачальника, внутрішнього постачальника та зовнішнього клієнта вказана кількість не кратна розфасовці;
- «Артикул з іншого договору. Артикул: [артикул]» – у замовленні на зовнішнього постачальника є артикули з різних договорів;
- «Артикул не знайдено у договорі. Артикул: [артикул]» – не знайдено договору із зовнішнім постачальником;
- «Артикул закрито на складі. Артикул: [артикул]» – на замовлення на зовнішнього постачальника артикул не відкритий на складі (перевірка на асортимент не пройдена) ;
- «Артикул перетинається у кількох договорах. Артикул: [артикул]» – в одному замовленні на внутрішнього постачальника є артикули з різних договорів;
- «Немає запису в LagerAssortiment для Артикулу: [артикул]» – у замовленні на внутрішнього постачальника артикул не відкрито у філіалі;
- «На складі не вказано розф. для замовлень. Артикул: [артикул]» – у замовленні на внутрішнього постачальника по артикулу не прописано розфасовку на складі;

- «Артикула немає. Артикул: [артикул]» – артикула, вказаного у замовленні, немає;
- «Дубль артикула на замовлення. Артикул: [артикул]» – той самий артикул повторюється в одному замовленні;
- «Кількість менша за нуль. Артикул: [артикул]» – у замовленні вказано кількість менша за нуль;
- «Артикул в асортименті філіалу FilID не знайдено. Артикул: [артикул]» – артикул відсутній в асортименті філіалу.

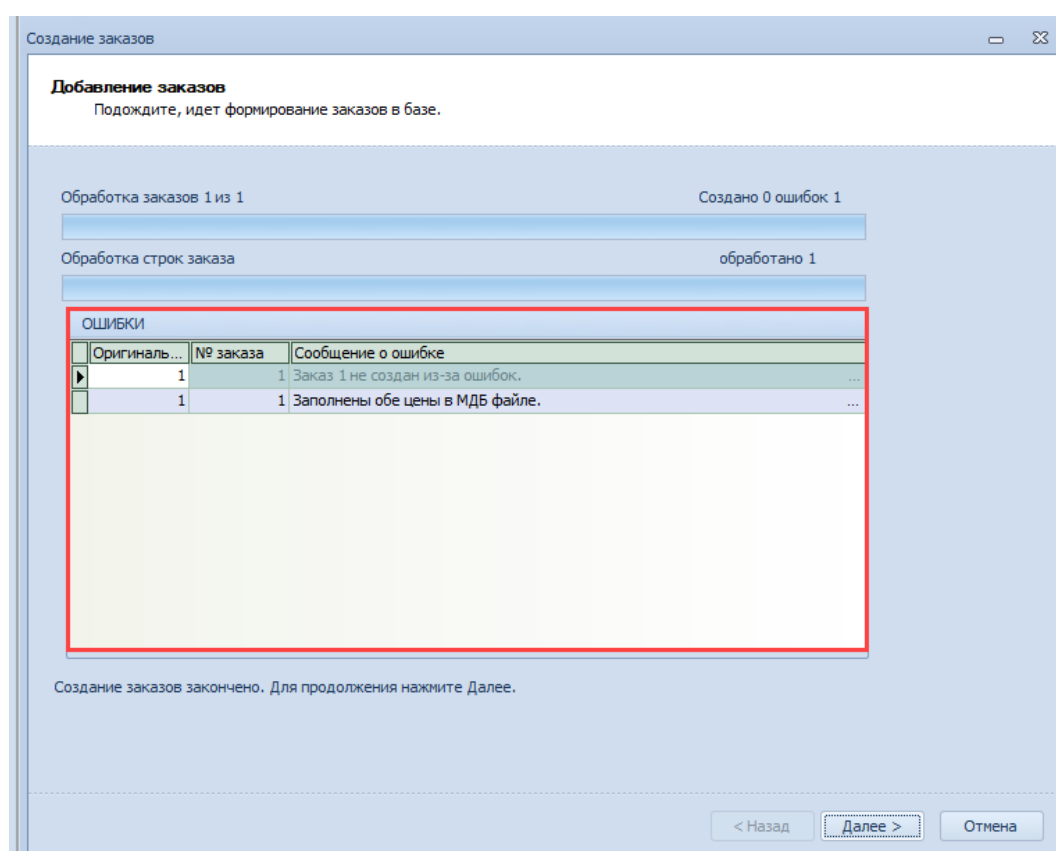
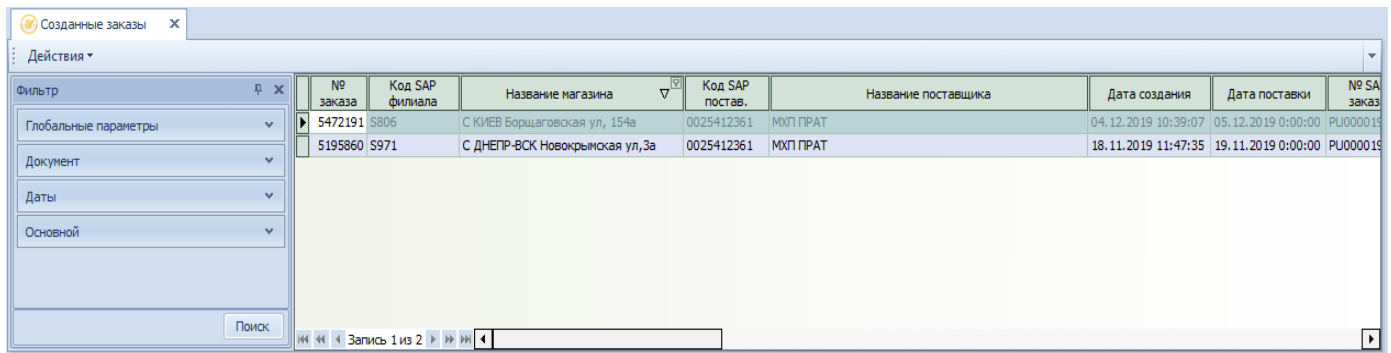


Рисунок 3.61 – Помилки, що виникли під час створення замовлення

Після натискання на кнопку «Далі» на екрані з'явиться повідомлення про успішне завершення замовлення і можна перейти до перегляду замовлень на закладці «Створені замовлення» (рис. 3.62).



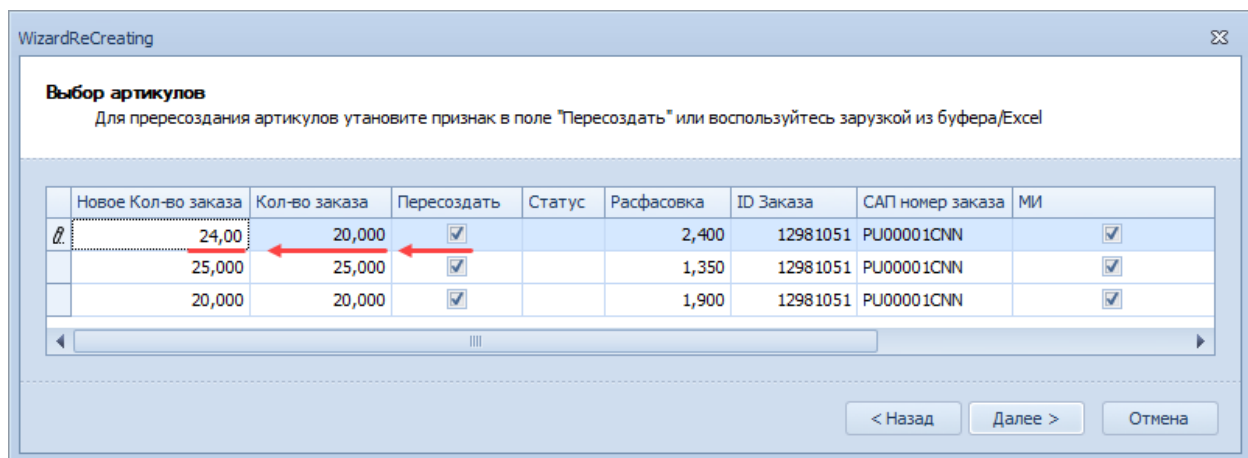
№ заказа	Код SAP филиала	Название магазина	Код SAP постав.	Название поставщика	Дата создания	Дата поставки	№ SA заказа
5472191	5806	С КИЕВ Борщаговская ул., 154а	0025412361	МХП ПРАТ	04.12.2019 10:39:07	05.12.2019 0:00:00	PU000019
5195860	5971	С ДНЕПР-ВСК Новокрымская ул, 3а	0025412361	МХП ПРАТ	18.11.2019 11:47:35	19.11.2019 0:00:00	PU000019

Рисунок 3.62 – Закладка «Створені замовлення»

Перестворення раніше створених замовлень.

Щоб перестворити замовлення по артикулу, необхідно обрати у табличній області «Перестворити замовлення по артикулу». В результаті відкриється вікно вибору артикулів та нової дати постачання

У вікні «Вибір артикулів» (рис. 3.63) виконується вибір артикулів та вказується кількість замовлення: вручну або імпортується із буфера/Excel.



Новое Кол-во заказа	Кол-во заказа	Пересоздать	Статус	Расфасовка	ID Заказа	САП номер заказа	МИ
24,00	20,000	☒		2,400	12981051	PU00001CNN	☒
25,000	25,000	☒		1,350	12981051	PU00001CNN	☒
20,000	20,000	☒		1,900	12981051	PU00001CNN	☒

Рисунок 3.63 – Редагування замовлення вручну

Для цього потрібно -

- активувати ознаку «Перестворити» в однойменній колонці, після чого в колонці «Нова Кількість замовлення» з'явиться значення з колонки «Кількість замовлення» ;
- Відредагувати значення в колонці «Нова Кількість замовлення».

Значення в колонці «Нова Кількість замовлення» має бути кратне значенню в колонці «Розфасовка». Якщо кількість розфасовки не кратна, при спробі застосувати її відкривається повідомлення про помилку «Замовлення не кратне розфасовці», яке після закриття буде відображатися в колонці «Статус» (рис. 3.64).

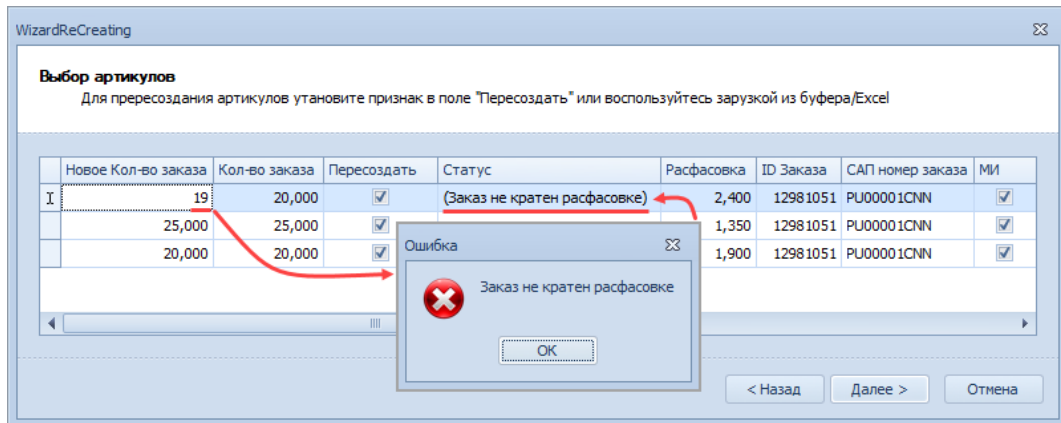


Рисунок 3.64 – Повідомлення про помилку «Замовлення не крате розфасовці»

Для вставки з буфера необхідно розписати дані в наступному вигляді –

- «Номер замовлення»;
- «Артикул»;
- «Кількість».

Якщо при виборі артикулів були виявлені помилки, їх опис з'явиться в колонці «Статус», а також відкриється повідомлення про помилку (рис. 3.65):

- «Перетворене замовлення» – замовлення вже скопійоване;
- «Закрито асортимент» – артикул закрито на складі;
- «Немає договору» – договір з оригінального номера закрито;
- «Оригінальне замовлення» – це замовлення скопійоване;
- «Замовлення не розфасоване» - замовлення не кратне розфасовці;
- «К-во змінювати не можна» – замовлення в якому не можна змінювати кількість.

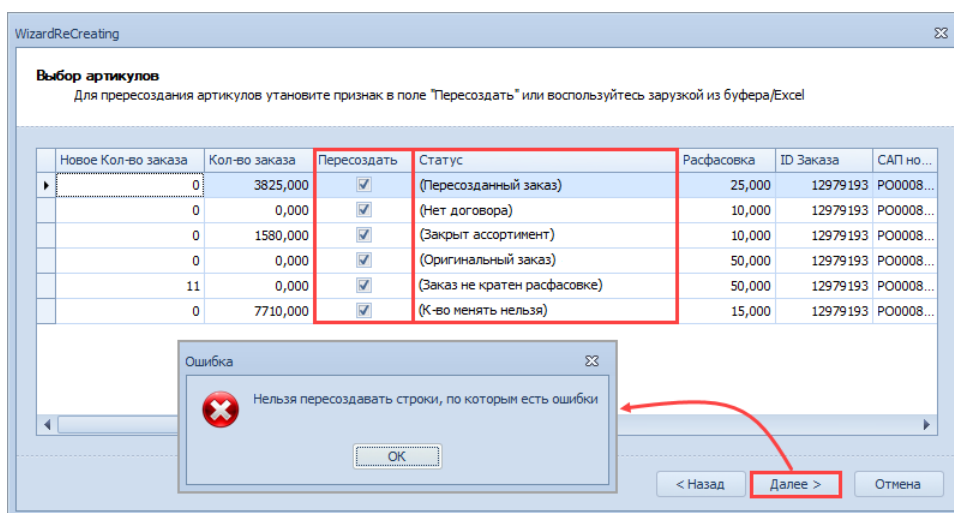


Рисунок 3.65 – Помилки під час вибору артикулів

У вікні «Вибір дати доставки» встановлюється нова дата доставки. Ліворуч від табличної області розташований блок із налаштуваннями (рис 3.66):

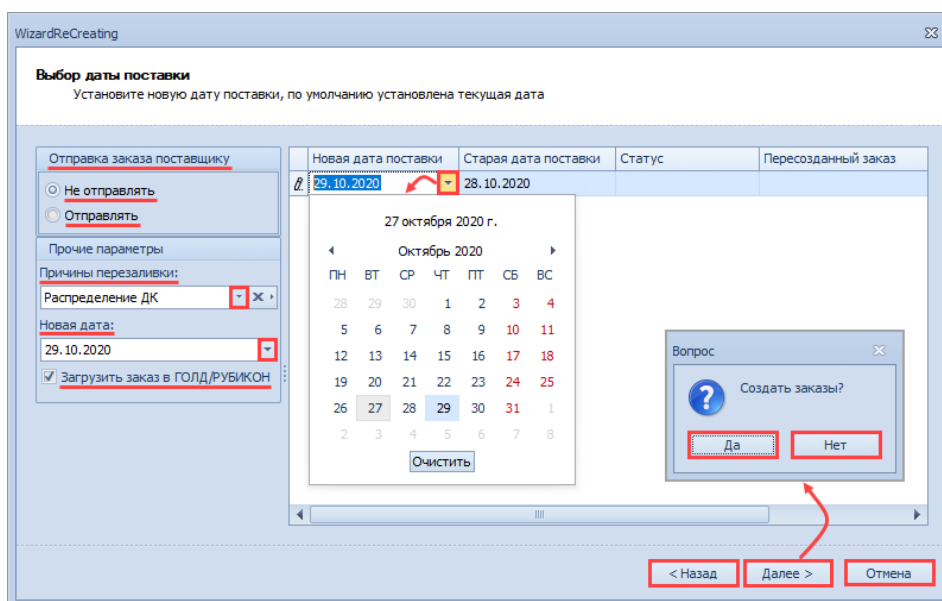


Рисунок 3.66 – Вікно «Вибір дати постачання»

В результаті з'явиться закладка «Перестворені замовлення» (рис. 3.67) з усіма перествореними замовленнями.

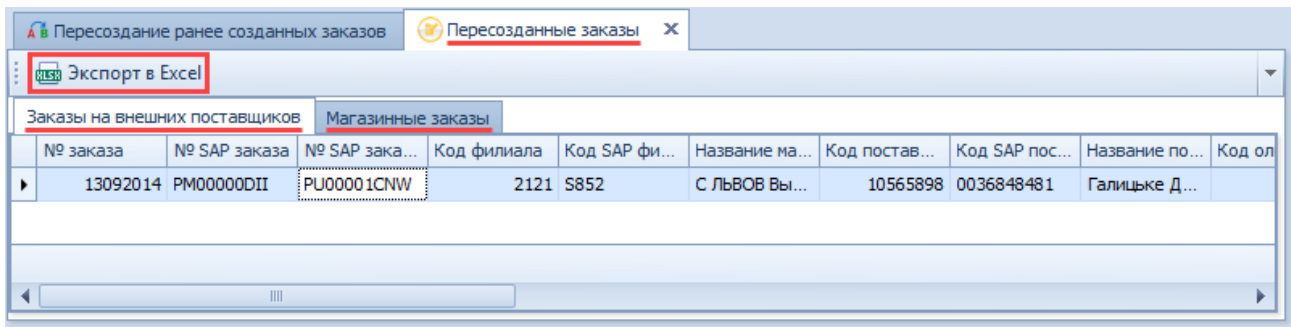


Рисунок 3.67 – Закладка «Перестворені замовлення»

Майстер створення замовлень зовнішній постачальник.

Щоб створити замовлення, необхідно встановити параметри в блоках «Основний», «Артикула», «Філіали», «Договори» (рис. 3.68).

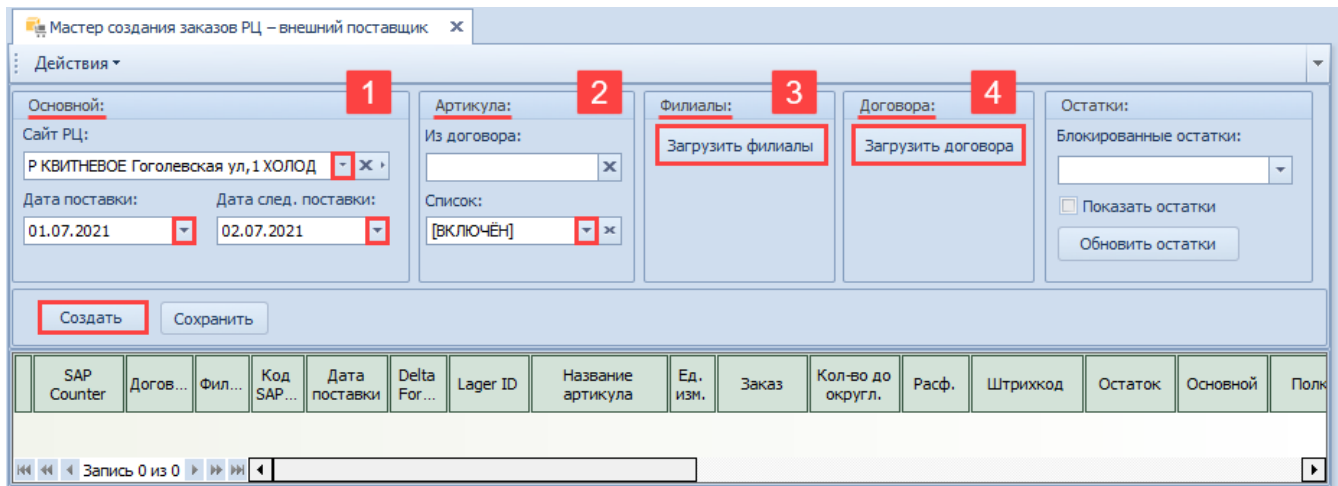


Рисунок 3.68 – Налаштування параметрів замовлення

У блоці «Основний» необхідно заповнити поля (Рис 3.68, Показчик 1):

- «Сайт РЦ» – обрати склад у списку;
- «Дата поставки» – вказати дату поставки;
- «Дата наст. Поставки» – вказати дату наступної поставки.

У блоці «Артикула» необхідно заповнити поля (Рис 3.68, Показчик 2):

- «З договору» – поле обов'язкове для заповнення, якщо не заповнено поле «Список» – необхідно вказати номер договору, щоб підтягнути з нього артикули;
- «Список» – поле обов'язкове для заповнення, якщо не заповнене поле «З договору» – необхідно обрати артикули у додатковому вікні.

Якщо заповнені обидва поля, потреба буде сформована за артикулом із договору.

У блоці «Філіали» можна завантажити перелік філіалів, наповнивши даними шаблон у Excel-файлі:

- «ІД філіалу» –код філіалу;
- «Дельта днів» –кількість днів від замовлення до постачання на магазини.

У блоці «Договори» можна завантажити перелік договорів, наповнивши даними шаблон у Excel-файлі:

- «ІД договору» –код договору;
- «ІД складу» – код складу;
- «Дата доставки» – необхідно вказати дату доставки;
- «Дата постачання наст.» – необхідно вказати дату наступного постачання.

Примітка: під час створення замовлення за номером договору блоки «Основний», «Артикула», «Філіали» стають неактивними (рис. 3.69).

Скриншот вікна «Мастер создания заказов РЦ – внешний поставщик». Вікно має кілька вкладок: «Основной», «Артикула», «Филиалы» та «Договора». Вкладка «Договора» активна, на ній є таблиця з колонками: «Dogovor ID», «RC_ID», «Дата п», «Дата п». В таблиці є один рядок з даними: «11636», «2082», «09.12.», «12.12.». Під таблицею є кнопка «Очистить», яка виділена червоною рамкою. Над таблицею є кнопки «Загрузить договор» та «Загрузить филиалы». В інших вкладках є поля для введення даних: «Сайт РЦ», «Дата поставки», «Дата след. поставки», «Из договора», «Список». Внизу вікна є кнопки «Создать» та «Сохранить».

Рисунок 3.69 – Результат завантаження договорів

Якщо у блоці «Залишки» (рис. 3.70) активована умова «Показати залишки» відображатиметься інформація в колонках «Залишок вага», «Залишок блокований», «Залишок РЛЦ», «Прогноз продажу».

Delta For...	Lager ID	Название артикула	Ед. изм.	Заказ	Кол-во до округл.	Расф.	Штрихкод	Остаток	Остаток (вес)	С бло
1	8853	Форель Филе С/м	кг	0,000	-9057,590	21,000	0250005012...	9471,590	9471,590	▲
1	12940	Креветки в пан...	170г	0,000	-62,730	8,000	4770190036...	48,000	9,888	□
1	14949	Хек филе с/м В ...	кг	0,000	-1850,000	10,000	0250005012...	2150,000	2110,000	□
1	34870	Салака С/м	кг	0,000	-290420,...	10,000	0250005013...	292380,000	292040,000	▼

Рисунок 3.70 – Результат створення замовлення згідно з заданими параметрами

Висновки до розділу. У даному розділі було розглянуто особливості і основні параметри для створення замовлень. Виконано огляд розробленої системи, а саме процедур та інших процесів формування замовлень, що відбуваються на стороні бази даних, детально вивчено особливості роботи автоматичної корекції замовлень та алгоритму контролю товарного запасу. Обґрунтовано підхід та розглянуто на прикладі реалізацію вивантаження даних з інших серверів.

Розглянуто структуру проєкту у Visual Studio який реалізовує інтерфейс користувача – основні принципи при роботі та обміні інформацією з базою даних, підключення до неї, використання власних бібліотек для можливості створення і модифікації власних стандартних об'єктів.

Проведено показ основних функцій інтерфейсу користувача – перегляд і робота з автоматично створеними замовленнями та ручне створення/перестворення замовлень.

4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

4.1 Опис ідеї проєкту

Для того аби зрозуміти мету розробки даної системи необхідно перш за все зрозуміти суть ідеї і які вигоди вона у собі несе для кінцевого споживача (табл. 4.1).

Таблиця 4.1 – Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробити систему для формування замовлень та прогнозування для автоматизації процесів логістики ритейлу з використанням методів аналізу даних	Торгівля, логістика	Оптимізувати і автоматизувати основні логістичні процеси ритейлу, запобігати додатковим витратам і втратам завдяки аналізу і прогнозуванню, контролювати процеси і приймати рішення в автоматизованому режимі.

Для того щоб дослідити актуальність проєкту і доцільність розробки наведено порівняльний аналіз з існуючими аналогами. В якості конкурентів розглядаються системи «Вас Управління торгівлею» і «1С Підприємство» (табл. 4.2).

Таблиця 4.2 – Визначення характеристик ідеї проєкту

Ідея	Концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
	Мій проєкт	BAS	1С П			
Автоматичне формування замовлень	+	-	-			+

Ідея	Товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
	Мій проект	BAS	1С П			
Інтеграція з системами, що відповідають за перевезення, транспорт, складське забезпечення	+	+-	-			+
Прогнозування створення замовлень	+	-	+-			+
Автоматичний аналіз і формування даних для замовлень	+	+-	-			+
Автоматична відправка документації, документообіг	+	-	+		+	
Моніторинг помилок системи	+	-	-			+

Як видно з порівняльного аналізу сильних і слабких сторін розробляємої системи та існуючих аналогів, реалізовані ідеї мають повну перевагу над існуючими функціональними рішеннями, що говорить про доцільність розробки даного проєкту.

4.2 Технічний аудит ідеї проєкту

У даному розділі проведеться аналіз необхідних технологій, їх наявність та доступність для реалізації ідеї проєкту.

Перш за все перед розробкою будь-якої системи варто розуміти можливість використання та наявність засобів реалізації (табл. 4.3).

Таблиця 4.3 – Технологічна здійсненність ідеї проєкту

Ідея проєкту	Технології реалізації	Наявність технологій	Доступність технологій
Розробка системи для формування замовлень та прогнозування для автоматизації логістичних процесів рітейлу з використанням методів аналізу даних	Мова програмування C#	Наявна	Доступна безкоштовно
	Мова програмування T-SQL	Наявна	Доступна безкоштовно
	СУБД MS SQL Server	Наявна	Доступна безкоштовно
	IDE MS Visual Studio	Наявна	Доступна безкоштовно

З результатів аналізу видно, що розробка проєкту не потребує використання нових, невідомих технологій, для цього достатньо наявних засобів, які можна використовувати безкоштовно.

4.3 Аналіз ринкових можливостей запуску стартап-проєкту

У даному розділі проведено аналіз ринкових можливостей для запуску стартап-проєкту.

У таблиці 4.4 наведено характеристику потенційного ринку.

Таблиця 4.4 – Попередня характеристика потенційного ринку проєкту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	4 (Excel, бази і сховища даних, CRM-системи, нейронні мережі)
2	Загальний обсяг продаж (реалізації послуг), грн/ум.од	35000\$ * 50 од. = 1750000 \$
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Відсутні
5	Специфічні вимоги до стандартизації та сертифікації	Відповідність стандартам ISO/IEC 9000, ISO 9001: 2008, ISO/IEC 25000:2014, ISO/IEC 25010:2011, SEI SW-CMM, ДСТУ ISO/IEC/IEEE 16326:2015, ДСТУ 3396.1-96, ДСТУ 2941-94, ДСТУ 2850-94, ДСТУ 2851-94, ДСТУ 2853-94
6	Середня норма рентабельності в галузі (або по ринку), %	$(1348150/1750000)*100=77\%$

На даний момент ринок має динаміку росту і гарні показники для входу зі своїм продуктом у галузі торгівлі про що говорить невелика кількість конкурентів, відсутність обмежень для входу та висока рентабельність.

Для запуску продукту необхідним є розуміння незадоволених потреб у галузі і цільову аудиторію для якої вирішуватимуться наявні проблеми (табл. 4.5).

Таблиця 4.5 – Характеристика потенційних клієнтів проєкту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
- Багатокористувацький інтерфейс - Резервне копіювання даних - Модулі збору інформації	- Ритейлери, мережі магазинів, ТРЦ. - Фізичні особи – підприємці галузей пов'язаних з торгівлею - Роздрібні магазини	Масштаби діяльності, кількість підпорядкованих закладів, забезпеченість, прибутки, необхідність в оптимізаційних процесах.	- Безпека роботи системи - Безпека даних - Інтерфейс відповідно до ролі - Випуск оновлень - Допомога у впровадженні - Мережеве поширення

При розробці і впровадженні будь-якого продукту завжди є як зовнішні так і внутрішні ризики, що можуть призвести до значних втрат, запобігти цьому допоможе заключення договорів та формування офіційних документів, а також підтримка та контроль стану програмного і технічного забезпечення.

Детальніше можливі загрози та варіанти їх вирішення наведено у таблиці 4.6.

Таблиця 4.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Криза замовника	Нестабільний економічний стан замовника, що загрожує не виплатою коштів.	Заклучення договорів з відтермінуванням оплат, оплатами по частинам тощо.
2.	Зовнішні чинники (загальноекономічна криза, стихійні лиха, карантин тощо)	Зменшення забезпеченості населення у цілому та підприємців і неготовність вкладати кошти у нові послуги.	Заклучення договорів з відтермінуванням оплат, оплатами по частинам тощо. Надання звітів, як даний проєкт допоможе зберегти і заощадити кошти.
3.	Збої у системах замовника	Не правильне функціонування системи через проблеми з забезпеченням споживача.	Проводити огляд програмного і технічного забезпечення замовника перед впровадженням
4.	Невелика зацікавленість, відсутність у клієнта інформації про проєкт	Недостатня реклама проєкту.	Збільшити вклад у рекламну кампанію

Вихід на ринок з новим продуктом супроводжується не тільки загрозами, а і на противагу певними можливостями, які варто розглядати як переваги і користуватися ними для успішного представлення власної пропозиції (табл. 4.7).

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Відсутність повних аналогів і конкуренції	Не висока конкуренції	Розробка і розвиток проєкту.
2.	Розвиток торгівлі в Україні	Велика кількість зацікавлених споживачів	Розвиток рекламної кампанії.
3.	Проведення консультацій і демонстрацій	Зацікавлення потенційних клієнтів.	Пошук можливостей для демонстрацій, безкоштовна демонстрація з виїздом на територію клієнта.

Використовувати ринкові можливості як переваги можна тільки враховуючи кампанію, пропозиції конкурентів та знаючи свою нішу (табл. 4.8-4.10).

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Монополія	Один виробник товару	Контролювати захищеність проєкту, відсутність плагіату, додавати унікальність.
Національний	Конкуренти по Україні	Розробка продукту по міжнародним правилам і стандартам аби вигідно вирізнитися.
Внутрішньогалузева	Конкуренція у сфері продажів, торгівлі	Надання як можна більше функціональних можливостей, які є важливими для даної галузі.

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Товарно-родова	Конкуренція між різними специфікаціями продукту	Розвиток власного продукту, надання більшого функціоналу.
Нецінова	Конкуренція у якості продукту	Вдосконалення і оновлення продукту.
Не марочна	Роль торгової марки не впливає значно на конкуренцію	Зацікавлення власне продуктом, а не маркою.

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Відсутні	Схожі застосунки	Менша ціна	Прибутки	Простіші для розуміння, менша ціна
Висновки	Інтенсивність на нульовому рівні, оскільки прямі конкуренти відсутні.	Аналоги є, але функціонал не зможе повністю замінити проєкт.	Диктують цінову політику	Залежно від забезпеченості клієнта, він має певну платоспроможність	Необхідно складну систему розділяти на простіші модулі, допомагати у впровадженні, виставляти конкретну і обґрунтовану ціну

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)
1.	Ціна	Зберігати ціну на рівні, але надавати більший функціонал.
2.	Унікальність продукту	Вивчати аналоги і оновлювати та покращувати продукт.
3.	Додаткові послуги до продукту	Допомога у впровадженні, налаштуванні, надання інструкцій.
4.	Результативність наданого функціоналу	Генерувати нові корисні знання.

Провівши аналіз конкуренції можна зробити висновок, що хоча прямі конкуренти відсутні, все ж варто враховувати схожі застосунки і власну розробку актуалізовувати у таких аспектах як ціна, функціональність, унікальність, надання підтримки та додаткових послуг (табл. 4.11).

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін

№ п/ п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні зі стартап-проєктом						
			-3	-2	-1	0	+1	+2	+3
1	Ціна	16					+		
2	Унікальність продукту	18							+
3	Додаткові послуги до продукту	13				+			
4	Результативність наданого функціоналу	18						+	

Підсумки з аналізу ринкових можливостей наведені у таблиці 4.12.

Таблиця 4.12 – SWOT- аналіз стартап-проєкту

Сильні сторони: відсутність аналогів даного проєкту на ринку, допомога у впровадженні, надання інструкцій і оновлень в подальшому.	Слабкі сторони: необхідність будувати базу клієнтів і своє ім'я з нуля, вміти зацікавити і довести унікальність та важливість продукту; відсутність відгуків-рекомендацій.
Можливості: керування логістичними процесами у галузі торгівлі, захист даних, аутентифікація, авторизація, мережевий доступ і поширення, формування звітів, отримання даних з локальних магазинів, складів і їх аналіз, прогнозування.	Загрози: невідповідне програмне і технічне забезпечення клієнта, зовнішні фактори, криза у середовищі клієнта.

Швидкість виходу і завоювання ринку залежить від обраної стратегії ринкової поведінки – завоювання клієнта (табл. 4.13).

Таблиця 4.13 – Альтернативи ринкового впровадження проєкту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Реклама у мережі інтернет	75%	3-4 місяці
2.	Організація зустрічей з клієнтами	90%	1-2 місяці
3.	Презентація на тематичних зустрічах, форумах, конференціях	80%	1-2 місяці

У результаті аналізу ринкових можливостей запуску стартап-проєкту видно, що конкуренція і стан ринку зараз в Україні дозволяє увійти з даною системою. Швидкому входу у ринок сприятиме реклама, публічні виступи-презентації та організація приватних демонстрацій, безпека розповсюдження і гарантії можна

досягти укладаючи договори, а затребуваність завдяки актуалізації функціоналу, цін і т.д.

4.4 Розробка ринкової стратегії стартап-проекту

Перш за все для розробки ринкової стратегії необхідно визначитися на які категорії споживачів буде орієнтована система (табл. 4.14); визначитися з базовою стратегією розвитку, конкурентної поведінки та позиціонування (табл. 4.15 – 4.17).

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Ритейлери, мережі магазинів, ТРЦ.	Висока, великі ритейлери зацікавлені у оптимізації логістичних процесів між усіма своїми одиницями.	90% - високий попит, через великі масштаби діяльності і необхідність контролю.	Низька конкуренція, через відсутність повних аналогів продукту.	Легко, є високий попит і низька конкуренція.
2.	Фізичні особи – підприємці галузей пов'язаних з торгівлею	Середня, споживач хоче оптимізувати роботу, але з обережністю ставиться до витрат.	65% - високий попит на продукцію, що є надійною і зможе задовольнити потреби.	Низька конкуренція, через відсутність повних аналогів продукту.	Середній рівень складності, є попит і низька конкуренція, але постає питання про доцільність витрат.

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
3.	Роздрібні магазини	Висока, бо компанії зацікавлені покращити свій сервіс	20% - низький попит, через невеликі масштаби діяльності.	Висока конкуренція, оскільки роздрібні магазини можуть задовольнятися менш функціональним ПЗ, виходячи зі своїх потреб.	Важко, через низький попит і можливість заміни схожим ПЗ.
Які цільові групи обрано: ритейлери, мережі магазинів, ТРЦ					

Таблиця 4.15 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Розробка системи прогнозування і формування замовлень на поставку товару	Стратегія диференційованого маркетингу	Багатокористувацький режим, доступ по мережі, резервне копіювання даних, доступний інтерфейс, прогнозування, аналіз та повністю автоматичне (з можливостями автоматизованого) формування ключової інформації	Стратегія диференціації

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Ні, оскільки є програми-аналоги, які все ж не в змозі покрити весь спектр необхідного функціоналу	Буде шукати нових споживачів і за можливості забирати у конкурентів	Буде використовувати стандартні (базові) алгоритми і ідеї з додаванням власних наробок і нових модулів	Стратегія широкої диференціації

Таблиця 4.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проєкту	Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту (три ключових)
1.	Багатокористувацький інтерфейс	Стратегія диференціації	Інтерфейс по ролям	• Можливість авторизації і аутентифікації • Збереження для себе у нотатках у власному просторі необхідних даних • Захист від несанкціонованого доступу

№ п/ п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспромо жні позиції власного стартап- проекту	Вибір асоціацій, які мають сформувані комплексну позицію власного проекту (три ключових)
2.	Резервне копіювання даних	Стратегія диференц іації	Захист інформації	• Попередження втрати даних • Можливість відкату при внесенні помилкових даних • Можливість ведення архіву
3.	Прогнозува ння і розрахунок необхідних взаємодій з товаром, генерування інформації для замовлень, формування замовлень	Стратегія диференц іації	Автоматизування робочих процесів	• Аналітика діяльності • Прогнозування, рекомендації до подальшої стратегії • Автоматичне формування замовлень без залучення користувача

У результаті за базову стратегію розвитку було обрано стратегію диференціації, як стратегію конкурентної поведінки - стратегію широкої диференціації.

4.5 Розробка маркетингової програми стартап-проєкту

У даному розділі буде сформовано маркетингову програму стартап-проєкту. Перш за все за основу для реклами і для успішної взаємодії з клієнтом необхідно оперувати ключовими перевагами продукту (табл. 4.18).

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Багатокористувацький інтерфейс	Доступ відповідно до ролі	Робота у системі відповідно до посади
2.	Прогнозування та аналіз даних	Можливість автоматизованого прийняття рішень	Зменшення витрат на аналітику і можливість проводити її глибше за рахунок навчання на попередніх даних
3.	Автоматичне створення замовлень	Впровадження модулів для автоматичного формування замовлень	Можливість формування замовлення як у автоматизованому режимі так і повністю автоматично на основі наявної інформації

Для аналізу отриманого результату розробки проєкту наведено опис його на трьох рівнях як товару, а саме як товар за задумом (що було закладено, яка ціль стояла на початку роботи з ідеєю та вподальшому), товару у реальному виконанні (що було отримано у результаті проектування, які особливості має продукт) та товару з підкріпленням (як можна вдосконалити продукт одразу і через деякий час після випуску) (табл. 4.19).

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Система формування замовлень та прогнозування для автоматизації логістичних процесів рітейлу з використанням методів аналізу даних		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Операційна система	М	Тх/Ор
	2. Оперативна пам'ять	М	Тх/Ор
	3. Процесор	М	Тх/Ор
	4. Швидкість інтернету	Нм	Тх/Вр
	5. Вільне місце на диску	Нм	Тх/Ор
	6. Кількість і вид портів	М	Тх
	7. Наявність пристроїв	М	Тх/Вр
	8. Кількість користувачів	Нм	Ор
	Якість: відповідність стандартам ISO/IEC 90003 і ISO 9001: 2008		
Пакування: носій інформації з системою, буклет з зазначенням торгової марки, назви продукту і інших загальних відомостей, інструкція з впровадження і описом необхідних технологій та навичок.			
Марка: VAMYK Software «Система формування замовлень та прогнозування для автоматизації логістичних процесів рітейлу з використанням методів аналізу даних»			
III. Товар із підкріпленням	До продажу: надання інструкції користувача, демонстрація		
	Після продажу: допомога у впровадженні, випуск оновлень		
За рахунок чого потенційний товар буде захищено від копіювання: патенту та авторського права			

Для успішної реалізації продукту необхідно встановити межі цін (табл. 4.20) та сформувати систему збуту і концепт маркетингових комунікацій (табл. 4.21-4.22).

Таблиця 4.20 – Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1000\$-5000\$	-	5000\$-100000\$	5000\$-35000\$

Таблиця 4.21 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Споживачі купують проєкт безпосередньо у розробника	- Аналітика конкурентних пропозицій - Випуск оновлень	0 - канал нульового рівня	Мережа інтернет, особиста зустріч на території клієнта/постачальника з демонстрацією.

Таблиця 4.22 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Споживачі дізнаються про проєкт за допомогою рекламних кампаній і консультацій	Мережа інтернет, консультації у телефонному режимі	Продукт без аналогів	Надати інформацію про існування даного продукту, зацікавити.	«Збільшуй прибутки, оптимізуй і автоматизуй роботу, приймай вірні рішення і отримай можливість керувати логістикою без безпосереднього втручання»

У результаті даного розділу було створено маркетингову програму проєкту, а саме визначено ключові переваги проєкту, здійснено опис моделі товару, визначено межі цін, сформовано систему збуту та концепцію маркетингових комунікацій [22].

Висновки до розділу. У даному розділі було виконано опис ідеї проєкту та дослідження аналогів, здійснено технічний аудит і виявлено, що всі засоби для розробки системи існують і є доступними; проаналізовано ринкові можливості і виявлено, що є умови для виходу з даним продуктом на український ринок.

ВИСНОВКИ

В результаті роботи над даною магістерською дисертацією була розроблена система для автоматичного формування замовлень на поставку товару та прогнозування, поставлені завдання були виконані, розроблена система відповідає сформованим вимогам.

У першому розділі було здійснено аналіз предметної області для якої проектується система та виокремлено особливості бізнес-процесу для якого розроблялася система, які впливають та задають принципи формування замовлень.

Було проведено пошук та дослідження існуючих рішень-аналогів для логістики галузі ритейлу – «BAS Управління торгівлею» та «1С Підприємство». У результаті виявлено, що дані продукти не задовольняють усім потребам користувачів як і при формуванні замовлень на поставку товару, так і у цілому по можливостям синхронізації, обміну інформацією з системами, тому розробка розглядаємої системи є доцільною. Також на основі виконаного аналізу було здійснено постановку задачі.

У другому розділі було описано використані методи (аналізу даних та прогнозування) та технології при розробці проєкту, а саме мов програмування C# та T-SQL, IDE MS Visual Studio, MS Sql Server, Ado.NET, SQL Server Integration Services (SSIS) та розглянуто їх особливості, які стали вирішальними під час вибору їх як інструменти для реалізації системи.

У третьому розділі розглянуто основний функціонал системи формування замовлень на поставку товару, принципи, особливості створення замовлень, подальші кроки у роботі з ними, основний інтерфейс та його програмну структуру. Для забезпечення функціонування даного логістичного процесу реалізовано взаємодію інтерфейсу з базою даних, вивантаження даних з різних систем та серверів, формування замовлень за розкладом, що є можливим за допомогою використання джобів, які у певний час запускають відповідний процес (процедури).

У четвертому розділі було виконано маркетинговий аналіз стартап-проєкту, а саме виконано дослідження ідеї, ринкової ситуації, маркетингової програми та обґрунтовано доцільність та можливість виходу даної системи на ринок.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Валерія Миколюк. Аналіз показників необхідних для автоматичного формування замовлень на поставку товару у галузі рітейлу//Наука онлайн: Международный научный электронный журнал. - 2021. - №11. URL: <https://nauka-online.com/publications/tehnicheskie-nauki/2021/11/05-5/>
- 2) 8 Must Have Features For Logistics and Supply Chain management Software. URL: <https://www.shipsoft.co/8-must-have-features-for-logistics-and-supply-chain-management-software/>
- 3) The five elements of logistics. URL: <https://www.futurelearn.com/info/courses/principles-global-management-logistics-assets/0/steps/65243>
- 4) BAS Управління торгівлею. Основна поставка на 1 робоче місце. URL: <https://soft.integra.ua/uk/product/upravlinnya-torgivleyu-dlya-ukrajiny/>
- 5) BAS управління торгівлею. URL: <https://akros.com.ua/bas-ut/>
- 6) BAS Управління торгівлею. URL: <https://elsygroup.com/ru/blog/bas-upravlinnya-torgivleyu2-36/>
- 7) Закупівлі під управлінням "1С:Підприємство 8.0. Управління торгівлею". URL: <https://dtki.com.ua/automation/ukr/avtomat/2004/10/10avto1.html>
- 8) Автоматическое формирование заказов поставщикам. URL: https://neokonsalt.ru/?page_id=2699
- 9) Забезпечення потреб в ERP. URL: <https://tqm.com.ua/ua/likbez/bp/zabezpechennya-potreb-v-1s-erp-2>
- 10) Язык C# и платформа .NET Core. URL: <https://metanit.com/sharp/tutorial/1.1.php>
- 11) 10 лучших IDE. URL: <https://timeweb.com/ru/community/articles/5-luchshih-ide-1>
- 12) Работа с Visual Studio. URL: https://professorweb.ru/my/csharp/charp_theory/level2/2_4.php

- 13) Введение в MS SQL Server и T-SQL. URL: <https://metanit.com/sql/sqlserver/1.1.php>
- 14) Что такое T-SQL. URL: <https://info-comp.ru/what-is-t-sql>
- 15) SQL Server Integration Services (SSIS). URL: <https://habr.com/ru/post/330618/>
- 16) Службы SSIS. URL: <https://docs.microsoft.com/ru-ru/sql/integration-services/ssis-how-to-create-an-etl-package?view=sql-server-ver15>
- 17) Введение в ADO.NET. URL: <https://metanit.com/sharp/adonet/1.1.php>
- 18) Дикань Н.В. Менеджмент: навч. посібник / Н.В. Дикань, П Борисенко. - К.: Знання, 2008. - 389 с.
- 19) Рубаков, С. В. Современные методы анализа данных / С. В. Рубаков // Управление наукой и наукометрия. – 2008. – № 7. – С. 165-176.
- 20) Акобир Шахиди. Деревья решений – общие принципы работы. URL: <http://www.basegroup.ru/trees/>
- 21) Нейронные сети. URL: <https://habr.com/ru/post/312450/>
- 22) Розроблення стартап-проекту : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с.