

Міністерство освіти та науки України
Харківський національний університет радіоелектроніки

Кафедра ПІ

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КУРСОВОЇ РОБОТИ
з дисципліни “Бази даних”

Тема роботи: «Інформаційна система «Інтернет-магазин побутової техніки»

Виконав

ст. гр. ПЗП-21-7

Ткаченко М.А.

Керівник:

ас. каф. ПІ

Зибіна К.В.

Робота захищена на оцінку

Комісія:

ас. каф. ПІ

доц. каф. ПІ

ст. викл. каф. ПІ

Зибіна К.В.

Мазурова О.О.

Черепанова Ю.Ю.

Харків, 2022 р.

Харківський національний університет радіоелектроніки

Кафедра Програмної інженерії

Дисципліна Бази даних

Спеціальність Програмна інженерія

Курс 2 Група ПЗП-21-7 Семестр 3

ЗАВДАННЯ
на курсову роботу студента

Ткаченка Максима Андрійовича

1. Тема роботи: «Інформаційна система «Інтернет-магазин побутової техніки»

2. Строк здачі закінченої роботи 23.12.2022

3. Вихідні дані для роботи: методичні вказівки до виконання курсової роботи, вимоги до інформаційної системи, предметна область, що пов'язана з діяльністю інтернет-магазину побутової техніки

4. Зміст розрахунково - пояснювальної записки: вступ, аналіз предметної області; постановка задачі; проектування бази даних; опис програми; висновки; перелік посилань.

5. Перелік графічного матеріалу: загальна діаграма класів, UML-діаграми, ER-діаграма, структура 1НФ, 2НФ, 3НФ, схема БД в 3НФ, копії екранів ("скріншоти") прикладної програми, приклади звітів прикладної програми

6. Дата видачі завдання 15.09.22 р.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів курсової роботи	Строк виконання етапів роботи	Примітки
1	Аналіз предметної області	15.09.22 – 30.09.22	Виконано
2	Постановка задачі	15.09.22 – 30.09.22	Виконано
3	Побудова ER-діаграми бази даних	30.09.22 – 23.10.22	Виконано
4	Оформлення розділів 1, 2 та 3.1, 3.2 пояснювальної записки	01.10.22 - 23.10.22	Виконано
5	Перша контрольна точка з курсового проекту	24.10.22 – 30.10.22	Виконано
6	Нормалізація бази даних	30.10.22 - 30.11.22	Виконано
7	Створення демо-версії програми	30.10.22 – 30.11.22	Виконано
8	Тестування програми, наповнення бази даних	01.11.22 - 30.11.22	Виконано
9	Друга контрольна точка з курсового проекту	25.11.22 – 03.12.22	Виконано
10	Реалізація остаточної версії програми	03.12.22 - 13.12.22	Виконано
11	Оформлення інших розділів пояснювальної записки	03.12.22 – 15.12.22	Виконано
12	Захист курсового проекту (третя контрольна точка)	16.12.22 - 23.12.22	Виконано

Студент _____

Керівник _____

ас. Зибіна К.В.

15.09.2022 р.

РЕФЕРАТ

Пояснювальна записка до курсової роботи: 44 с., 23 рис., 1 табл., 5 джерел.

БАЗИ ДАНИХ, ІНТЕРНЕТ-МАГАЗИН, МОВА ПРОГРАМУВАННЯ C#, МОВА ПРОГРАМУВАННЯ SQL, СУБД MICROSOFT SQL SERVER, СИСТЕМА УПРАВЛІННЯ ВМІСТОМ САЙТУ

Метою курсового проектування є розробка програмної системи управління вмістом сайту «Інтернет-магазин побутової техніки», закріплення знань, здобутих при вивченні дисципліни "Бази даних", придбання навичок у проектування та розробці баз даних, опануванні програмних засобів розробки інформаційних систем, що засновані на базах даних.

Методи розробки базуються на використанні інтегрованого середовища розробки Microsoft Visual Studio, мови програмування C#, СУБД Microsoft SQL Server і технології доступу до даних ADO.NET.

В результаті отримана програмна система управління вмістом сайту під назвою “Інтернет-магазин побутової техніки”, яка дозволяє додавати, редагувати, видаляти, зберігати, переглядати контент, отримувати статистики, автоматизувати діяльність, шукати, сортувати, фільтрувати контент, формувати звіти.

ЗМІСТ

Вступ.....	6
1 Аналіз та концептуальне моделювання предметної області.....	7
1.1 Аналіз предметної області.....	7
1.2 Аналіз існуючих аналогів.....	8
1.3 Концептуальне моделювання предметної області.....	10
2 Постановка задачі	17
3 Проектування бази даних	18
3.1 Побудова ER-діаграми	18
3.2 Вибір та побудова логічної моделі бази даних на основі ER-діаграми ...	19
3.3 Побудова логічної моделі бази даних шляхом нормалізації	21
4 Опис програми	26
4.1 Загальні відомості	26
4.2 Виклик і завантаження	27
4.3 Призначення і логічна структура	27
4.4 Опис фізичної моделі бази даних	28
4.5 Опис програмної реалізації	31
4.6 Опис задачі автоматизації	40
Висновки	42
Перелік джерел посилання	43

ВСТУП

Програмна система управління вмістом сайту – це невід'ємна частина інтернет-магазину. В час швидкого масового зростання використання та впровадження Інтернету, інтернет-магазини потребують спрощення управління базами даних. Багато інтернет-магазинів побудовані на системі управління вмістом сайту.

Метою даної курсової роботи є проектування і розробка програмної системи управління вмістом сайту інтернет-магазину побутової техніки, створеній на основі бази даних.

Спроековано реляційну базу даних. База даних створена і керована за допомогою запитів мови програмування SQL. SQL відіграє центральну роль у коді програми. Обрано для розробки інтегроване середовище розробки Microsoft Visual Studio і систему управління базою даних Microsoft SQL Server. Написано читабельний, багаторазово використовуваний код. Розроблено програму з графічним інтерфейсом користувача.

Під час курсового проектування розроблена інформаційна система, яка спрощує управління вмістом сайту інтернет-магазину побутової техніки.

1 АНАЛІЗ ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області

Інформаційна система «Інтернет-магазин побутової техніки» – це фрагмент системи управління вмістом сайту. Ця програма займається організацією контенту сайту продажу товарів в Інтернеті. З інформаційною системою працює адміністратор.

Інформаційна система має об'єкти: товари, замовлення, покупці, продавці.

Перерахуємо процеси системи керування вмістом сайту:

- додавання, редагування, видалення, збереження, перегляд контенту;
- отримання статистики;
- задача автоматизації;
- пошук, фільтрація, сортування товарів;
- формування звітів.

Система має користувача: адміністратора.

Інформаційні потреби користувачів:

- адміністратор повинен додавати, редагувати, видаляти, зберігати, переглядати контент;
- адміністратор повинен отримувати статистику;
- адміністратор повинен автоматизувати діяльність;
- адміністратор повинен шукати, сортувати, фільтрувати контент;
- адміністратор повинен формувати звіти.

Результати діяльності користувачів:

- в результаті система формує звіти, статистику, керує контентом.

Загальні характеристики процесів споживання та обробки інформації:

- інформація в таблиці вноситься вручну;
- звіти формуються в електронному вигляді та друкуються для підпису.

1.2 Аналіз існуючих аналогів

Проаналізуємо відомі аналоги-системи інтернет-магазинів.

Перша система-аналог має назву: інтернет-магазин «Розетка»[6].

Опишемо функціонального призначення системи:

- уведення, збереження і редагування інформації;
- пошук, сортування та фільтрація даних;
- статистика за продажами, по ціні, по рейтингу;
- отримати можливість керувати покупками в особистому кабінеті покупця;
- керувати магазином в особистому кабінеті продавця;
- підготувати продавцю прайс-лист товарів у форматі XML, щоб розмістити товари на маркетплейсі;
- зберігати дані для наступних покупок (отримувач, адреса, платіжна картка) для спрощення повторних покупок;
- робити акції, щоб значно збільшити кількість замовлень товарів (задача автоматизації);
- завантажити електронний чек (звіт);
- надрукувати товарний чек з реквізитами продавця і відправити чек разом із замовленням (звіт);
- обробляти замовлення автоматично 24/7;
- переглянути історію своїх минулих замовлень та відстежити статус актуальних замовлень;
- створити заявку на повернення і попередньо вказати свої побажання що до обміну чи повернення коштів за товар (електронний документообіг).

Опишемо переваги та недоліки системи:

Перерахуємо переваги:

- автоматизація опрацювання замовлень, документообігу;
- масштабованість;

- керованість;
- логістика;
- компактність;
- швидкість;
- низькі трудовитрати.

Перерахуємо недоліки:

- вразливість баз даних до шкідливого програмного забезпечення, наприклад вразливість до вірусу-винищувача даних, що замаскований під програму-вимагач грошей.

Друга система-аналог має назву: інтернет-магазин «Amazon»[7].

Опишемо функціональне призначення системи:

- уведення, збереження і редагування інформації про товари;
- пошук, сортування та фільтрація даних;
- контролювати продажі в особистому кабінеті;
- керувати запасами;
- оновлювати ціни;
- переглядати статус актуальних замовлень;
- керувати і обробляти замовлення;
- керувати своїм магазином за допомогою простих у використанні інформаційних панелей, які пропонують спрощений спосіб управління магазином;
- отримати статистику.

Опишемо переваги та недоліки системи.

Перерахуємо переваги:

- автоматизація;
- керованість;
- логістика;
- масштабованість;
- швидкість;
- компактність;

- низькі трудовитрати.

Перерахуємо недоліки:

- вразливість баз даних до шкідливого програмного забезпечення.

1.3 Концептуальне моделювання предметної області

Наведено загальну USE-CASE–діаграму (див. рис. 1).

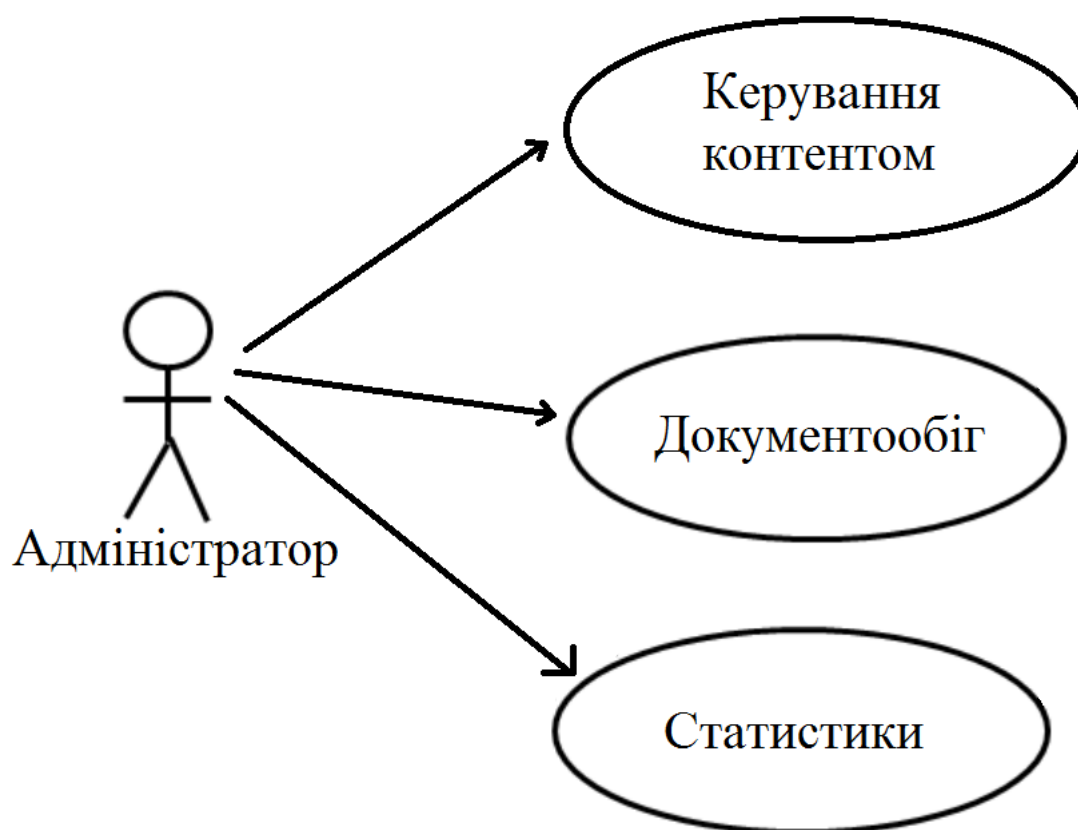


Рисунок 1 – Загальна USE-CASE діаграма

На загальній USE-CASE діаграмі зображен функціонал системи для адміністратора: керування контентом, документообіг, статистики.

1.3.1 Опис концептів ПрО, їх властивостей та зв'язків між ними

Наведено загальну діаграму класів (див. рис. 2).

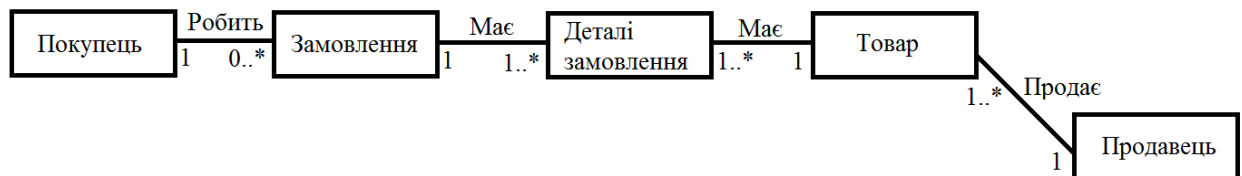


Рисунок 2 – Загальна діаграма класів

Концепт “Покупець” має атрибути: електронна пошта, пароль, ім'я, прізвище, місто, область, поштовий індекс, дата реєстрації, телефон

Концепт “Замовлення” має атрибути: статус, дата, місто, область, поштовий індекс, телефон.

Концепт “Деталі замовлення” має атрибути: назва товару, сума, кількість.

Концепт “Товар” має атрибути: назва, ціна, назва бренду, назва категорії, рік випуску, вага, опис товару, рейтинг, кількість.

Концепт “Продавець” має атрибути: електронна пошта, пароль, ім'я, прізвище, місто, регіон, дата реєстрації, телефон.

1.3.2 Опис інформаційних потреб користувачів

Перерахуємо інформаційні потреби користувачів:

- база даних інтернет-магазину побутової техніки;
- потреба в можливості пошуку товарів по назві товару, фільтрації товарів по рейтингу, сортування по полям: назва, ціна, назва бренду, назва категорії, рік виробництва, вага, рейтинг, кількість;

- потреба в можливості додавання, редагування, видалення, перегляду, товарів для продавця;
- потреба в отриманні для продавця статистику про продажі, рейтинг товарів;
- потреба в автоматичному підборі переможця акційного розіграшу для автоматизації діяльності;
- потреба в формуванні звіту стану запасів для продавця і звіту товарного чеку для покупця.

Наведено основний функціонал на детальній USE-CASE діаграмі (див. рис. 3).



Рисунок 3 – Детальна USE-CASE діаграма

Також опишемо задачу автоматизації.

Підбір клієнта-переможця акційного розіграшу “Клієнт року”, враховуючи критерії. Клієнт року отримає великі знижки в інтернет-магазині.

Актуальність це полегшення підбору клієнта-переможця для акційного розіграшу, який збільшить продажі інтернет-магазину. Складність в написанні коду і SQL запитів, які враховують формулу справедливого вибору у реальному часі для підбору клієнта-переможця.

Наведено діаграму потоків даних для задачі автоматизації (див. рис. 4).

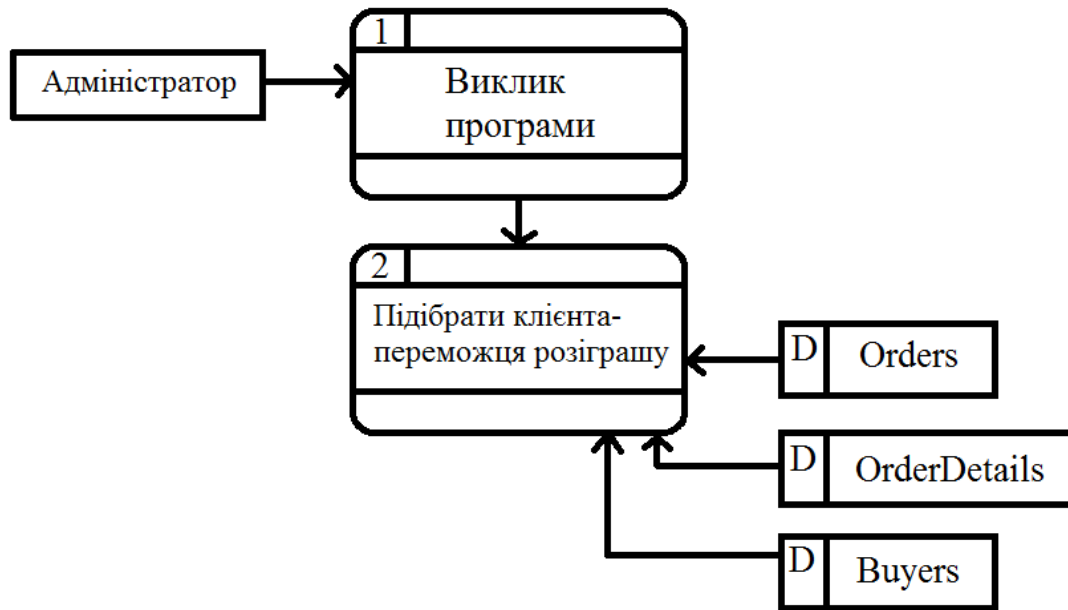


Рисунок 4 – DFD-діаграма для задачі автоматизації

Наведено діаграму видів діяльності для задачі автоматизації (див. рис. 5).

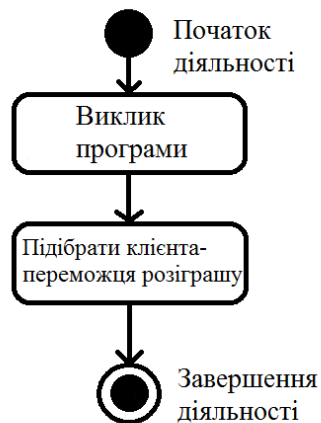


Рисунок 5 – Діаграма видів діяльності для задачі автоматизації

Наведено діаграму зміни станів БД для задачі автоматизації (див. рис. 6).

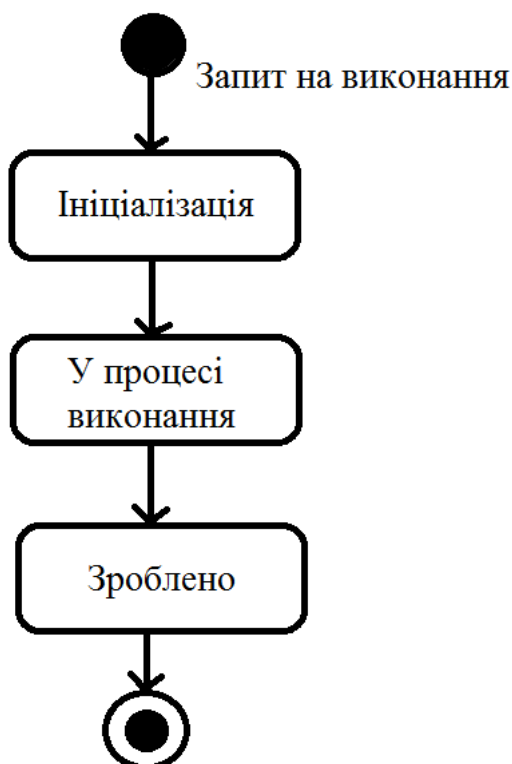


Рисунок 6 – Діаграма зміни станів БД для задачі автоматизації

Отже, діаграма наочно демонструє зміни станів БД для задачі автоматизації.

1.3.3 Опис існуючого документообігу в ПрО

Документообіг складається з наступних документів:

- звіт “Клієнт року”, де вказано ім’я і прізвище клієнта-переможця акційного розіграшу;
- звіт “Товарний чек”, де вказано назви, кількість, ціни товарів, дата, сума. Звіти формуються в Word за допомогою бібліотеки Microsoft.Office.Interop.Word заміною тексту шаблону на значення з бази даних.

1.3.4 Опис алгоритмічних залежностей показників

Залежності стосовно системи відбору клієнта-переможця акційного розіграшу, який отримає значні знижки в інтернет-магазині.

Формула вибору клієнта-переможця акційного розіграшу S повинна враховувати наступний показник. Розглянемо формулу (1).

$$W = f(MS), \quad (1)$$

де MS – клієнт, який замовив найдорожчий товар

1.3.5 Опис обмежень цілісності

Обмеження стосовно ідентифікації:

- кожен покупець однозначно ідентифікується його номером;
- кожен продавець однозначно ідентифікується його номером;
- кожен товар однозначно ідентифікується його номером;
- кожне замовлення однозначно ідентифікується його номером;
- кожна деталь замовлення однозначно ідентифікується його номером.

Обмеження стосовно відношень між поняттями (сутностями):

- кожен покупець може зробити нуль або більше замовлень;
- кожне замовлення має одне або більше деталей замовлення;
- кожен товар має одне або більше деталей замовлення;
- кожен продавець продає один або більше товарів.

1.3.6 Вимоги до ІС та існуючі обмеження

Перерахуємо вимоги:

- вимоги до СУБД: підтримка паралельної роботи близька 1000 користувачів, використання СУБД Microsoft SQL Server;
- вимоги до ОС: система повинна працювати під операційну систему Windows 7 та вище;
- вимоги до додаткового ПО: система повинна експортувати документи в формат Microsoft Word (.docx).

1.3.7 Лінгвістичні відносини

Особистий кабінет — місце на сайті, в якому міститься інформація про персональні дані власника акаунту (користувача), про проведені операції, а також інша інформація.

Логін — унікальний ідентифікатор користувача, що слугує для позначення його в системі.

Акційний розіграш — це маркетингова акція/пропозиція від інтернет-магазину про значні знижки клієнту-переможцю.

Контент — інформація про товар що розміщується на сайті: назва та опис товарних позицій, фотографії, характеристики товару, інформація про продавця.

Інтернет-магазин — веб-сайт, що являє собою комунікаційну платформу для розміщення продавцями інформації про товари з метою їх продажу.

2 ПОСТАНОВКА ЗАДАЧІ

Тема: ІС «Інтернет-магазин побутової техніки»

Існує інформація про товари: назва, ціна, назва бренду, назва категорії, рік виробництва, вага, рейтинг. Є інформація про покупців і продавців: електронна пошта, пароль, ім'я, прізвище, місто, область, поштовий індекс, дата реєстрації, телефон. Є інформація про замовлення: статус, дата, місто, область, поштовий індекс, телефон, назва товару, сума, кількість.

Необхідно реалізувати:

- а) додавання, редагування, видалення, збереження, перегляд інформації;
- б) пошук товарів по назві товару, фільтрацію товарів по рейтингу, сортування товарів по назві, ціні, назві бренду, назві категорії, року виробництва, вазі, рейтингу, кількості;
- в) 4 найбільш часто виникаючих до бази даних запити:
 - 1) отримати топ-10 товарів за рейтингом (назва, кількість рейтингу);
 - 2) отримати топ-10 товарів новинок за роком випуску (назва товару, рік випуску);
 - 3) отримати топ-10 найдорожчих товарів (назва, ціна в грн);
 - 4) отримати топ-10 найдешевших товарів (назва, ціна в грн);
- г) можливість звертання з довільним запитом до бази даних;
- д) рахування підсумкової суми товарного чека (обчислювальне поле);
- е) підбір клієнта-переможця для акційного розіграшу “Клієнт року”, який отримає великі знижки в інтернет-магазині (задача автоматизації);
- є) формування звіту “Клієнт року”, де вказано ім'я і прізвище клієнта-переможця (звіт);
- ж) формування звіту “Товарний чек”, де вказана дата, назви товарів, кількість, ціни, сума (звіт).

3 ПРОЕКТУВАННЯ БАЗИ ДАНИХ

3.1 Побудова ER-діаграми

Розглянемо ER-діаграму (див. рис. 7).

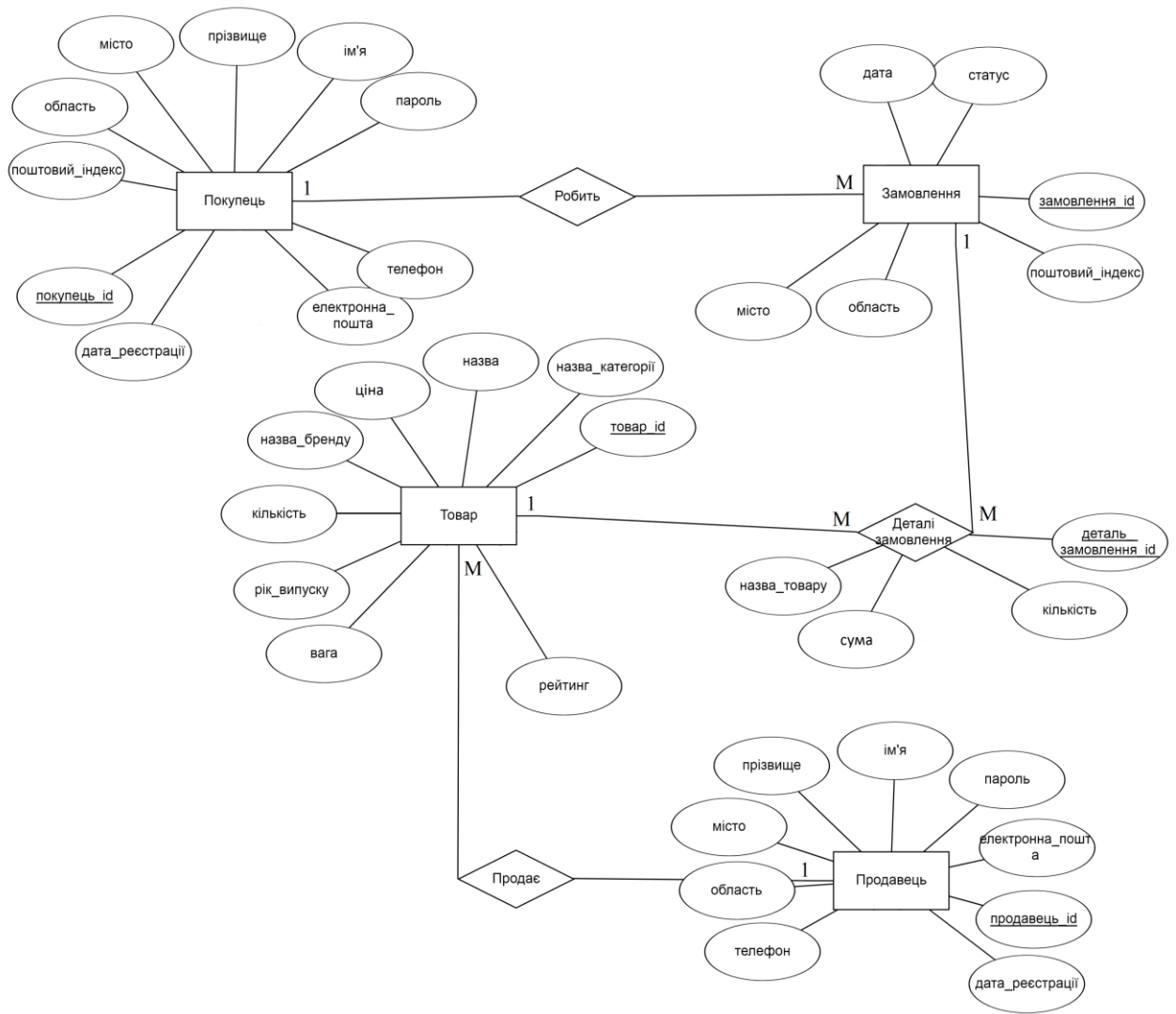


Рисунок 7 – ER-діаграма за нотацією Чена

Далі наведемо словесний опис.

Діаграма має сутності: покупець, замовлення, деталі замовлення, товар, продавець, запас.

Сутність “Покупець” має атрибути: ідентифікатор покупця, електронна пошта, пароль, ім'я, прізвище, місто, область, поштовий індекс, дата реєстрації, телефон.

Сутність “Замовлення” має атрибути: ідентифікатор замовлення, ідентифікатор покупця, статус, дата, місто, область, поштовий індекс, телефон.

Сутність “Деталі замовлення” має атрибути: ідентифікатор деталі замовлення, ідентифікатор замовлення, ідентифікатор товару, назва товару, сума, кількість.

Сутність “Товар” має атрибути: ідентифікатор товару, ідентифікатор продавця, ідентифікатор запасу, назва, ціна, назва бренду, назва категорії, рік випуску, вага, опис товару, рейтинг.

Сутність “Продавець” має атрибути: ідентифікатор продавця, електронна пошта, пароль, ім'я, прізвище, місто, регіон, дата реєстрації, телефон.

Діаграма має зв'язок “один-до-багатьох” між сутностями покупець і замовлення, зв'язок “один-до-багатьох” між сутностями замовлення і деталі замовлення, зв'язок “один-до-багатьох” між сутностями товар і деталі замовлення, зв'язок “один-до-багатьох” між сутностями продавець і товар.

3.2 Вибір та побудова логічної моделі бази даних на основі ER-діаграми

Вивчивши аналоги створення логічної моделі бази даних для інтернет-магазинів, більшість логічних моделей – реляційні та вони мають загальну закономірність побудови ефективної схеми. Реляційна логічна модель має проміжну сутність, в моєму випадку “деталі замовлення” між сутностями товар і замовлення, щоб уникнути зв'язок “багато-до-багатьох”. Це дизайн бази

даних з усіма необхідними таблицями та стовпчиками для типового сайту електронної комерції. Логічна модель реляційної бази даних була побудована в Microsoft Access. Далі наведена схема реляційної БД (див. рис. 8).

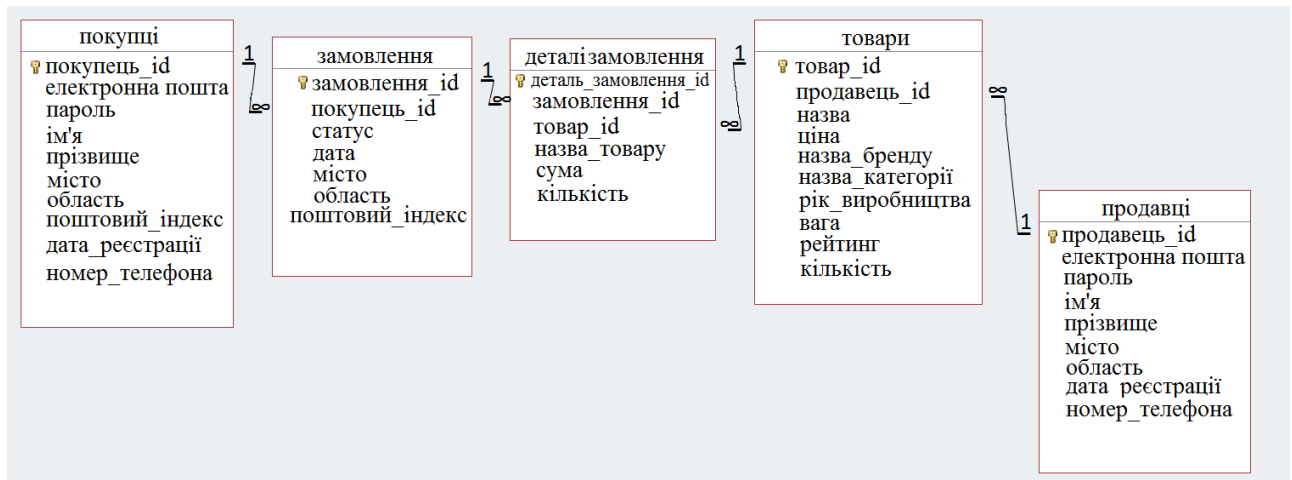


Рисунок 8 – Схема реляційної БД

Проведемо перевірку кожної таблиці схеми на відповідність вимогам третьої нормальної форми.

Таблиці мають першу нормальну форму, тому що на перетині будь-якого стовпця та будь-якого запису знаходиться атомарне значення.

Таблиці мають другу нормальну форму, тому що вони знаходяться у першій нормальній формі і мають по одному простому первинному ключ, отже кожен неключовий атрибут повно функціонально залежить від первинного ключа.

Таблиці знаходяться в третій нормальній формі, тому що вони знаходяться у другій нормальній формі та між неключовими атрибутами не має транзитивних залежностей.

Далі намальовано функціональні залежності в кожній таблиці (див. рис. 9).

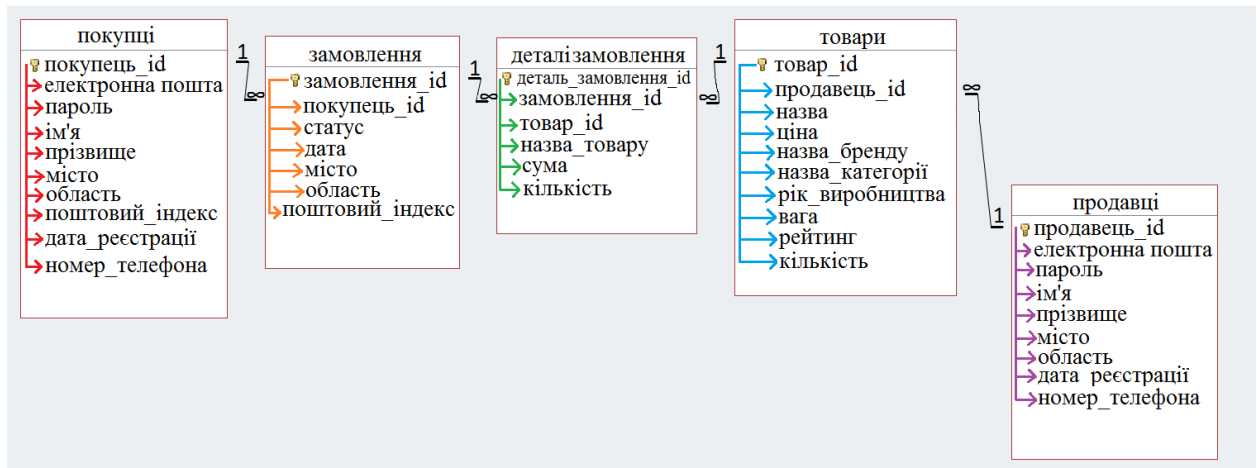


Рисунок 9 – Функціональні залежності

Отже, отримана схема БД знаходиться в третій нормальній формі.

3.3 Побудова логічної моделі бази даних шляхом нормалізації

Логічна модель реляційної бази даних будується шляхом нормалізації універсального відношення «Інтернет-магазин побутової техніки» (див. рис. 10). Візьмемо зв'язку з 5-ти сутностей з ER-діаграми, об'єднавши їх атрибути в одне універсальне відношення та проведемо його покрокову нормалізацію до 3-ї нормальної форми. Оберемо первинний простий ключ товар_id. Наведемо всі залежності на діаграмі універсального відношення (див. рис. 10). Покроково приведемо універсальне відношення до 3-ї нормальної форми. Доведемо, що отримані залежності задовільняють усім вимогам 1-ї, 2-ї та 3-ї нормальних форм.

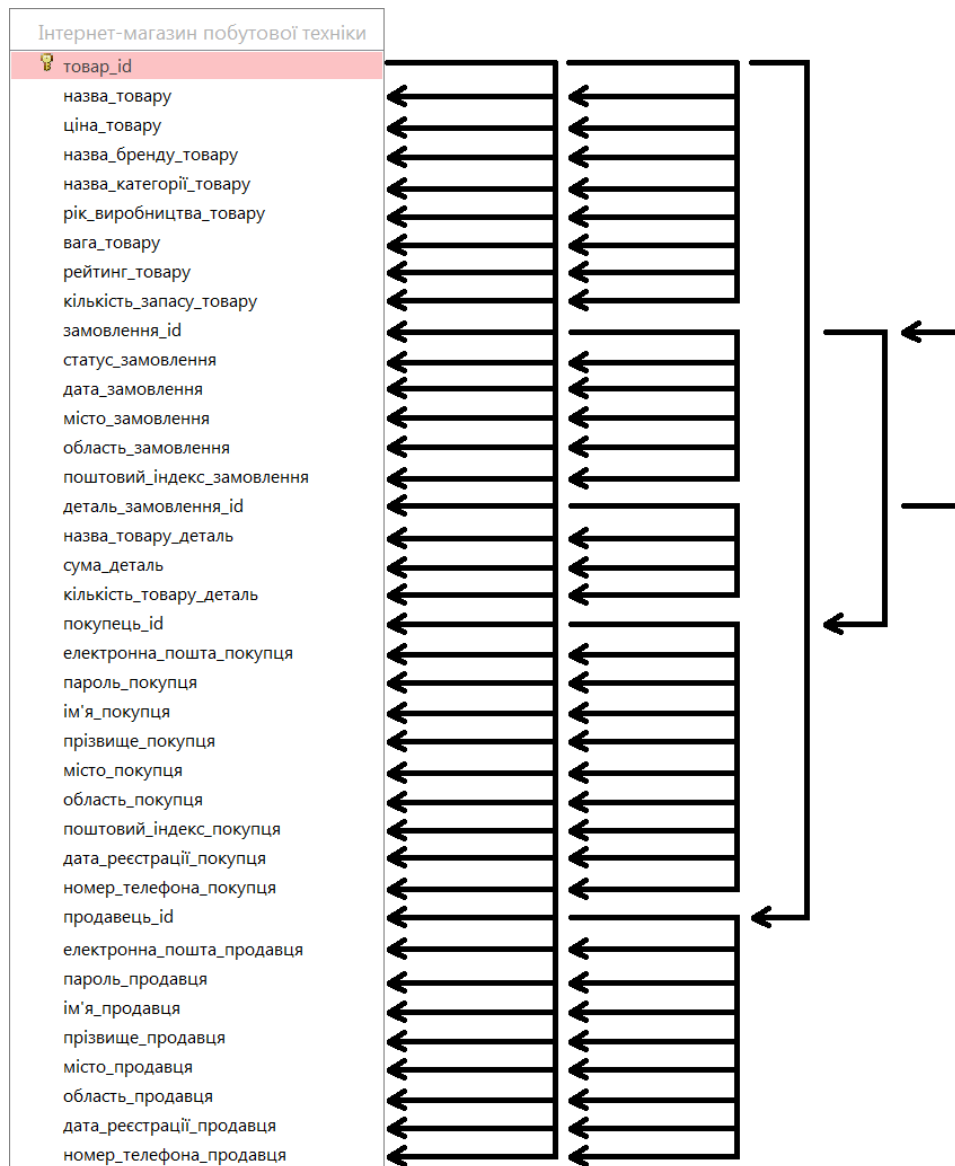


Рисунок 10 – Універсальне відношення «Інтернет-магазин побутової техніки» з залежностями

Відношення знаходиться в першій нормальній формі, тому що на перетині будь-якого стовпця та будь-якого запису знаходиться атомарне значення. І відношення знаходиться у другій нормальній формі тому що відношення знаходиться у першій нормальній формі і відношення має один простий первинний ключ, отже кожен неключовий атрибут повністю функціонально залежить від простого первинного ключа відношення.

Перерахуємо функціональні залежності від первинного ключа:

- {товар_id} -> назва_товару;
- {товар_id} -> ціна_товару;
- {товар_id} -> назва_бренду_товару;
- {товар_id} -> назва_категорії_товару;
- {товар_id} -> рік виробництва_товару;
- {товар_id} -> вага_товару;
- {товар_id} -> рейтинг_товару;
- {товар_id} -> кількість_запасу_товару;
- {товар_id} -> замовлення_id;
- {товар_id} -> статус_замовлення;
- {товар_id} -> дата_замовлення;
- {товар_id} -> місто_замовлення;
- {товар_id} -> область_замовлення;
- {товар_id} -> поштовий_індекс_замовлення;
- {товар_id} -> деталь_замовлення_id;
- {товар_id} -> назва_товару_деталь;
- {товар_id} -> сума_деталь;
- {товар_id} -> кількість_товару_деталь;
- {товар_id} -> покупець_id;
- {товар_id} -> електронна_пошта_покупця;
- {товар_id} -> пароль_покупця, ім'я покупця;
- {товар_id} -> прізвище_покупця;
- {товар_id} -> місто_покупця;
- {товар_id} -> область_покупця;
- {товар_id} -> поштовий_індекс_покупця;
- {товар_id} -> дата_реєстрації_покупця;
- {товар_id} -> номер_телефона_покупця;
- {товар_id} -> продавець_id;
- {товар_id} -> електронна_пошта_продавця;
- {товар_id} -> пароль_продавця, ім'я_продавця;

- {товар_id} -> прізвище_продавця;
- {товар_id} -> місто_продавця;
- {товар_id} -> область_продавця;
- {товар_id} -> дата_реєстрації_продавця;
- {товар_id} -> номер_телефона_продавця.

Щоб логічна модель бази даних знаходилась в третій нормальній формі, потрібно щоб відношення знаходилося у другій нормальній формі та між неключовими атрибутами не було транзитивних залежностей. Перерахуємо транзитивні залежності:

- {товар_id} -> назва_товару;
- {товар_id} -> ціна_товару;
- {товар_id} -> назва_бренду_товару;
- {товар_id} -> назва_категорії_товару;
- {товар_id} -> рік_виробництва_товару;
- {товар_id} -> вага_товару;
- {товар_id} -> рейтинг_товару;
- {товар_id} -> кількість_запасу_товару;
- {товар_id} -> продавець_id;
- {замовлення_id} -> статус_замовлення;
- {замовлення_id} -> дата_замовлення;
- {замовлення_id} -> місто_замовлення;
- {замовлення_id} -> область_замовлення;
- {замовлення_id} -> поштовий_індекс_замовлення;
- {замовлення_id} -> покупець_id;
- {деталь_замовлення_id} -> замовлення_id;
- {деталь_замовлення_id} -> назва_товару_деталь;
- {деталь_замовлення_id} -> сума_деталь;
- {деталь_замовлення_id} -> кількість_товару_деталь;
- {покупець_id} -> електронна_пошта_покупця;
- {покупець_id} -> пароль_покупця, ім'я_покупця;

- {покупець_id} -> прізвище_покупця;
- {покупець_id} -> місто_покупця;
- {покупець_id} -> область_покупця;
- {покупець_id} -> поштовий_індекс_покупця;
- {покупець_id} -> дата_реєстрації_покупця;
- {покупець_id} -> номер_телефона_покупця;
- {продавець_id} -> електронна_пошта_продавця;
- {продавець_id} -> пароль_продавця, ім'я_продавця;
- {продавець_id} -> прізвище_продавця;
- {продавець_id} -> місто_продавця;
- {продавець_id} -> область_продавця;
- {продавець_id} -> дата_реєстрації_продавця;
- {продавець_id} -> номер_телефона_продавця.

Виконаємо процедуру приведення логічної моделі бази даних до третьої нормальної форми, тобто розіб'ємо універсальне відношення на наступні таблиці, щоб між неключовими атрибутами не було транзитивних залежностей. Залишилися тільки функціональні залежності від первинних ключів.

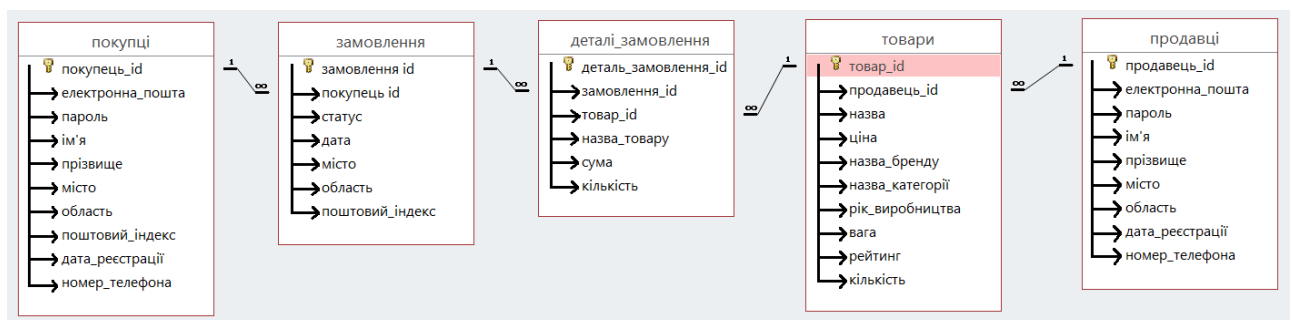


Рисунок 11 – Логічна модель бази даних в третій нормальній формі, яка отримана шляхом нормалізації

Отже, отримана логічна модель бази даних в третій нормальній формі шляхом нормалізації, яка збігається з схемою, що була спроектована на рисунку 8 в підрозділі 3.2.

4 ОПИС ПРОГРАМИ

4.1 Загальні відомості

Програмна система управління вмістом сайту створена в інтегрованому середовищі розробки Microsoft Visual Studio. Програма використовує реляційну систему управління базою даних Microsoft SQL Server. Система управління реляційними базами даних є найпоширенішим стандартом, яка є основою більшості інтернет-магазинів. Microsoft SQL Server – це система управління реляційними базами даних, яка ідеально підходить як для малих, так і для великих додатків. Microsoft SQL Server — це безкоштовна реляційна система керування базами даних, яка ідеально підходить як для малих, так і для великих програм, дуже швидка, надійна, масштабована і проста у використанні, притримується стандарту SQL. Також системи управління вмістом сайту інтернет магазину використовують цю СУБД. Недоліком може бути, що Microsoft SQL Server має обмежену платформну сумісність. Microsoft SQL Server призначений лише для роботи на машинах під керуванням Windows. Неможливість запуску програми на платформах, відмінних від Windows. Додаток, написаний на мові програмування C#, використовує .NET Framework, тому є сенс використовувати технологію доступу до даних ADO.NET і інтегроване середовище розробки Visual Studio, оскільки все це створено компанією Microsoft. Необхідні мінімальні характеристики персонального комп'ютеру, що необхідні для нормального функціонування програми – це 32 бітний двухядерний процесор, монітор Full HD, установлена операційна система Windows 7 і більше, установлене інтегроване середовище розробки Visual Studio, наявність миши.

4.2 Виклик і завантаження

Програма – це десктопний додаток. Для виклику програми потрібно установити наступне програмне забезпечення: інтегроване середовище розробки Visual Studio і систему управління базами даних Microsoft SQL Server. В інсталяторі Visual Studio виберіть робоче навантаження розробки десктопних додатків на .NET. Таким чином ви установите провайдер даних .NET framework, тобто зв'язок програми з базою даних налагоджено шляхом технології доступу до даних ADO.NET. Потім потрібно скачати архів коду на персональний ком'ютер і распакувати в якусь папку. Відкрити папку за допомогою Visual Studio, наприклад перетягнувши папку поверх іконки Visual Studio. Відкрити рішення store.sln в Solution Explorer. Тепер все готово для виклику програми. Виклик програми можна здійснити за допомогою пункту Start Without Debugging меню Debug або комбінації клавіш Ctrl + F5.

4.3 Призначення і логічна структура

Інформаційна система – це програмна система управління вмістом сайту інтернет-магазину побутової техніки, створеного на основі реляційної бази даних. База даних – це невід'ємна частина інтернет-магазину. Програма потрібна для спрощення управління базою даних сайту через графічний інтерфейс користувача. З інформаційною системою працює адміністратор. Програма може виконувати наступні функції: додавання, редагування, видалення, збереження, перегляд товарів і інших таблиць, пошук товарів по назві товару, фільтрація товарів по рейтингу, сортування товарів по стовпцям таблиці товари, отримання статистики, можливість звертання з довільним запитом до бази даних, задача автоматизації підбору клієнта-переможця для

акційного розіграшу “Клієнт року”, документобігу, а саме формування звітів “Клієнт року” і “Товарний чек”, рахування підсумкової суми товарного чека. Меню-орієнтований інтерфейс містить пункти, які відкривають віконні модулі. Функція введення, збереження і редагування інформації про товари, пошук товарів по назві товару, фільтрація товарів по рейтингу, сортування по стовпцям таблиці товари реалізована в модулі MainForm. Функція отримання статистики реалізована в модулі Stats. Функція можливості звертання з довільним запитом до бази даних реалізована в модулі QueryEdit. Задача автоматизації підбору клієнта-переможця акційного розіграшу “Клієнт року” реалізована в модулі Report1WithAutomatization. Функція формування звіту “Клієнт року” реалізована в модулі Report1WithAutomatization. Функція формування звіту “Товарний чек” і визначення підсумкової суми товарного чека реалізована в модулі Report2.

4.4 Опис фізичної моделі бази даних

За допомогою команд CREATE TABLE мови SQL і Microsoft SQL Server Management Studio у базі даних реалізовані типові таблиці для інтернет-магазину: товари, покупці, продавці, замовлення і деталі замовлення. Наведено найменування стовпців і загалом структуру таблиць. Обрані типи даних – це цілочисельний int, строковий varchar, часовий date. Таблиці мають обмеження первинного і зовнішнього ключа. SQL-запит має спочатку команди на створення таблиць. Далі наведемо SQL-запити на створення таблиць бази даних за допомогою команди CREATE TABLE:

```
CREATE TABLE buyers (  
buyer_id int PRIMARY KEY,  
email varchar(50) ,  
password varchar(50) ,
```

```
first_name varchar(50) ,  
last_name varchar(50) ,  
city varchar(50) ,  
region varchar(50) ,  
postal_code varchar(5) ,  
registration_date date ,  
phone varchar(50)  
);
```

```
CREATE TABLE orders (  
order_id int PRIMARY KEY ,  
buyer_id int FOREIGN KEY REFERENCES buyers(buyer_id) ,  
status varchar(50) ,  
date date ,  
city varchar(50) ,  
region varchar(50) ,  
postal_code varchar(5)  
);
```

```
CREATE TABLE sellers (  
seller_id int PRIMARY KEY ,  
email varchar(50) ,  
password varchar(50) ,  
first_name varchar(50) ,  
last_name varchar(50) ,  
city varchar(50) ,  
region varchar(50) ,  
registration_date date ,  
phone varchar(50)  
);
```

```
CREATE TABLE products (  
product_id int PRIMARY KEY,  
seller_id int FOREIGN KEY REFERENCES sellers(seller_id),  
name varchar(50),  
price int,  
brand_name varchar(50),  
category_name varchar(50),  
production_year int,  
weight varchar(50),  
rating int,  
quantity int  
);
```

```
CREATE TABLE orderdetails (  
order_detail_id int PRIMARY KEY,  
order_id int FOREIGN KEY REFERENCES orders(order_id),  
product_id int FOREIGN KEY REFERENCES  
products(product_id),  
product_name varchar(50),  
sum int,  
quantity int  
);
```

SQL-запит після має команди заповнення бази даних INSERT INTO. Заповнення бази даних – це трудомісткий процес. Наведено вибірка команд INSERT INTO (див. рис. 12).

```

INSERT INTO products VALUES (1, 1, 'Холодильник SAMSUNG
RB38T600FSA/UA', 21999, 'Sumsung', 'Холодильники', 2021,
'71 кг', 5, 20);
INSERT INTO products VALUES (2, 2, 'Телевізор Samsung
UE55AU7100UXUA', 20999, 'Sumsung', 'Телевізори', 2020,
'14.2 кг', 4, 25);
INSERT INTO products VALUES (3, 3, 'Пральна машина
SAMSUNG WW62J32G0PW/UA', 16199, 'Sumsung', 'Пральні
машини', 2020, '53 кг', 4, 20);
INSERT INTO products VALUES (4, 4, 'Пилосос Philips 2000
series XB2125/09', 3599, 'Philips', 'Пилососи', 2019, '4
кг', 5, 30);
INSERT INTO products VALUES (5, 5, 'Праска PHILIPS Azur
GC4909/60', 3399, 'Philips', 'Праски', 2019, '1.669 кг',
5, 50);

```

Рисунок 12 – Фрагмент вибірки команд INSERT INTO

Заповнення бази даних виконано за допомогою команд INSERT INTO, але база даних також може бути заповнена через графічний інтерфейс.

4.5 Опис програмної реалізації

Інтерфейс користувача функції додавання, редагування, видалення, збереження, перегляду товарів і інших таблиць реалізовано за допомогою DataGridView і bindingNavigator. Інтерфейс користувача функції пошуку товару по назві товару реалізовано за допомогою textBox. Інтерфейс користувача функції фільтрації товарів по рейтингу реалізовано за допомогою comboBox. Інтерфейс користувача функції сортування товарів по стовпцям таблиці товари

реалізован за допомогою comboBox з значеннями з бази даних. Код функцій головної форми реалізован за допомогою мови програмування C# і SQL. Використовувалися функції-обробники подій, оператор switch, SqlConnection, SqlDataAdapter, DataTable, SQL запити. Далі наведено код реалізації пошуку товару по назві, фільтрації по рейтингу, сортування по стовпцям таблиці товарів. Наведено код функцій пошуку, фільтрації, сортування з SQL-запитами:

```
private void MainForm_Load(object sender, EventArgs e)
{
    connection = new SqlConnection(@"Data
Source=ADMIN\SQLEXPRESS;Initial Catalog=store;Integrated
Security=True");
    connection.Open();
    SqlDataAdapter dataAdapter = new
SqlDataAdapter("SELECT * FROM products", connection);
    DataSet dataBase = new DataSet();
    dataAdapter.Fill(dataBase);
    dataGridView1.DataSource =
dataBase.Tables[0];
}
private void textBox1_TextChanged(object sender,
EventArgs e)
{
    (dataGridView1.DataSource as
DataTable).DefaultView.RowFilter = $"name LIKE
'{textBox1.Text}%'";
}
private void comboBox1_SelectedIndexChanged(object
sender, EventArgs e)
{
    switch (comboBox1.SelectedIndex)
```



```

        {
            case 0:
                (dataGridView1.DataSource as
DataTable).DefaultView.RowFilter = $"rating='5'";
                break;
            case 1:
                (dataGridView1.DataSource as
DataTable).DefaultView.RowFilter = $"rating='4'";
                break;
            case 2:
                (dataGridView1.DataSource as
DataTable).DefaultView.RowFilter = $"rating='3'";
                break;
            case 3:
                (dataGridView1.DataSource as
DataTable).DefaultView.RowFilter = "";
                break;
        }
    }

    private void comboBox2_SelectedIndexChanged(object
sender, EventArgs e)
    {
        switch (comboBox2.SelectedIndex)
        {
            case 0:

                SqlDataAdapter dataAdapter1 = new
SqlDataAdapter("SELECT * FROM products ORDER BY name;",
connection);

```

```

        DataTable dataTable1 = new
DataTable();

        dataAdapter1.Fill(dataTable1);
        dataGridView1.DataSource =
dataTable1;

        break;
    case 1:
        SqlDataAdapter dataAdapter2 = new
SqlDataAdapter("SELECT * FROM products ORDER BY price;",
connection);

        DataTable dataTable2 = new
DataTable();

        dataAdapter2.Fill(dataTable2);
        dataGridView1.DataSource =
dataTable2;

        break;
    case 2:
        SqlDataAdapter dataAdapter3 = new
SqlDataAdapter("SELECT * FROM products ORDER BY
brand_name;", connection);

        DataTable dataTable3 = new
DataTable();

        dataAdapter3.Fill(dataTable3);
        dataGridView1.DataSource =
dataTable3;

        break;
    case 3:
        SqlDataAdapter dataAdapter4 = new
SqlDataAdapter("SELECT * FROM products ORDER BY
category_name;", connection);

```

```
        DataTable dataTable4 = new
DataTable();

        dataAdapter4.Fill(dataTable4);
        dataGridView1.DataSource =
dataTable4;

        break;
    case 4:
        SqlDataAdapter dataAdapter5 = new
SqlDataAdapter("SELECT * FROM products ORDER BY
production_year;", connection);

        DataTable dataTable5 = new
DataTable();

        dataAdapter5.Fill(dataTable5);
        dataGridView1.DataSource =
dataTable5;

        break;
    case 5:
        SqlDataAdapter dataAdapter6 = new
SqlDataAdapter("SELECT * FROM products ORDER BY weight;",
connection);

        DataTable dataTable6 = new
DataTable();

        dataAdapter6.Fill(dataTable6);
        dataGridView1.DataSource =
dataTable6;

        break;
    case 6:
        SqlDataAdapter dataAdapter7 = new
SqlDataAdapter("SELECT * FROM products ORDER BY rating;",
connection);
```

```

        DataTable dataTable7 = new
DataTable();

        dataAdapter7.Fill(dataTable7);
        dataGridView1.DataSource =
dataTable7;

        break;
    case 7:
        SqlDataAdapter dataAdapter8 = new
SqlDataAdapter("SELECT * FROM products ORDER BY
quantity;", connection);

        DataTable dataTable8 = new
DataTable();

        dataAdapter8.Fill(dataTable8);
        dataGridView1.DataSource =
dataTable8;

        break;
    }
}

```

Наведемо текст SQL-запитів з пошуку, фільтрації, сортування:

```

//пошук
$"SELECT * FROM products WHERE name LIKE
'#{textBox1.Text}%'";

//фільтрація
SELECT * FROM products WHERE rating='5';
SELECT * FROM products WHERE rating='4';
SELECT * FROM products WHERE rating='3';

//сортування
SELECT * FROM products ORDER BY name;
SELECT * FROM products ORDER BY price;
SELECT * FROM products ORDER BY brand_name;

```

```

SELECT * FROM products ORDER BY category_name;
SELECT * FROM products ORDER BY production_year;
SELECT * FROM products ORDER BY weighth;
SELECT * FROM products ORDER BY rating;
SELECT * FROM products ORDER BY quantity;

```

Наведемо форму додавання, редагування, видалення, збереження, перегляду, пошуку, фільтрації, сортування товарів (див. рис. 14)

product_id	seller_id	name	price	brand_name	category_name	production_year	weight	rating	quantity
1	1	Холодильник S...	21999	Samsung	Холодильники	2021	71 кг	5	20
2	2	Телевізор Sams...	20999	Samsung	Телевізори	2020	14.2 кг	4	25
3	3	Пральна машин...	16199	Samsung	Пральні машини	2020	53 кг	4	20
4	4	Пилосос Philips...	3599	Philips	Пилососи	2019	4 кг	5	30
5	5	Правка PHILIPS...	3399	Philips	Правки	2019	1.669 кг	5	50
6	6	Посудомийна м...	12999	Whirlpool	Посудомийні ма...	2020	29.6 кг	4	20
7	7	Плита HANSA F...	9549	Hansa	Плити	2020	41 кг	3	20
8	8	Кондиціонер HY...	13499	Hyundai	Кондиціонери	2022	28 кг	4	20
9	9	Котел газовий I...	26277	Immergas	Котли опалення	2018	29 кг	5	25

Рисунок 14 – Форма додавання, редагування, видалення, збереження, перегляду, пошуку, фільтрації, сортування товарів

Інтерфейс користувача функції документобігу, а саме формування звіту “Товарний чек” реалізован за допомогою DateTimePicker, ComboBox, TextBox і Button (див. рис. 15). Також рахується підсумкова сума в товарному чеку. Найменування і ціна товару представлена як ComboBox з випадаючими значеннями з бази даних. Кількість реалізована як TextBox. Звіти формуються в Word за допомогою бібліотеки Microsoft.Office.Interop.Word заміною тексту шаблону на значення з бази даних. В звітах використовується try catch синтаксис для збереження працездатності програми у разі помилок.

Звіт "Товарний чек"

Новий товарний чек

Перший товар

Дата: 19 грудня 2022 г.

Найменування: Холодильник SAMSUNG RB38T600FSA/UA

Кількість: 1

Ціна: 21999

Другий товар

Найменування: Телевізор Samsung UE55AU7100UXUA

Кількість: 1

Ціна: 20999

Третій товар

Найменування: Пральна машина SAMSUNG WW62J32G0PW/UA

Кількість: 1

Ціна: 16199

Сформувати

Рисунок 15 – Форма звіту “Товарний чек”

Наведемо сформований звіт “Товарний чек” (див. рис. 16).

товарний_чек.docx - Word (Сбой активации продукта)

Товарний чек

Від 19.12.2022

№	Найменування	Кількість	Ціна
1	Холодильник SAMSUNG RB38T600FSA/UA	1	21999
2	Телевізор Samsung UE55AU7100UXUA	1	20999
3	Пральна машина SAMSUNG WW62J32G0PW/UA	1	16199
4			
5			
6			
7			
8			
9			

Всього на суму: 59197

Товар відгустив: Іван Іванов Підпис:

Рисунок 16 – Сформований звіт “Товарний чек”

Інтерфейс користувача функції отримання статистики реалізован за допомогою Button і DataGridView (див. рис. 17). Використовується

SqlConnection, SqlDataAdapter, DataTable. Використовується синтаксис try catch для збереження працездатності програми у разі виникнення помилок.

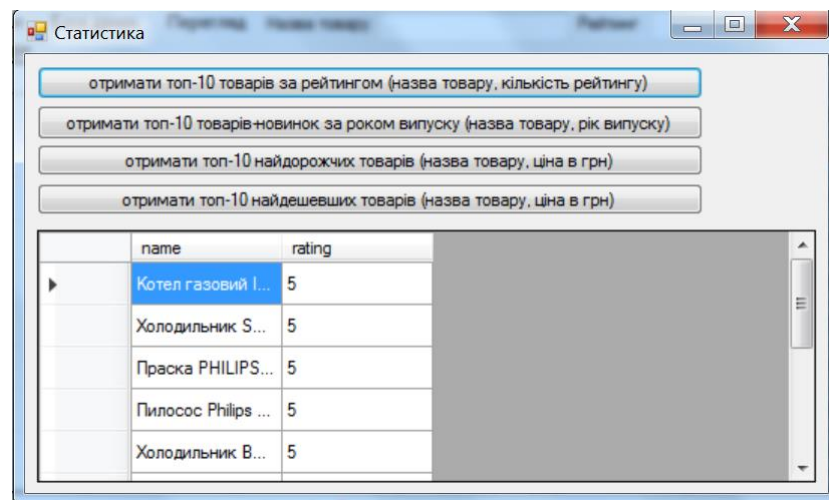


Рисунок 17 – Форма статистики

Наведемо тексти SQL-запитів на статистику (див. рис. 18).

```
SELECT TOP 10 name, rating FROM products ORDER BY rating
DESC;
SELECT TOP 10 name, production_year FROM products ORDER
BY production_year DESC;
SELECT TOP 10 name, price FROM products ORDER BY price
DESC;
SELECT TOP 10 name, price FROM products ORDER BY price;
```

Рисунок 18 – Текст запитів на статистику

Інтерфейс користувача функції можливості звертання з довільним запитом до бази даних реалізован за допомогою RichTextBox, Button, DataGridView (див. рис. 19). Використовувалися функції-обробники подій. Використовується синтаксис try catch для збереження працездатності програми у разі виникнення помилок. Використовується SqlConnection, SqlDataAdapter, DataTable.

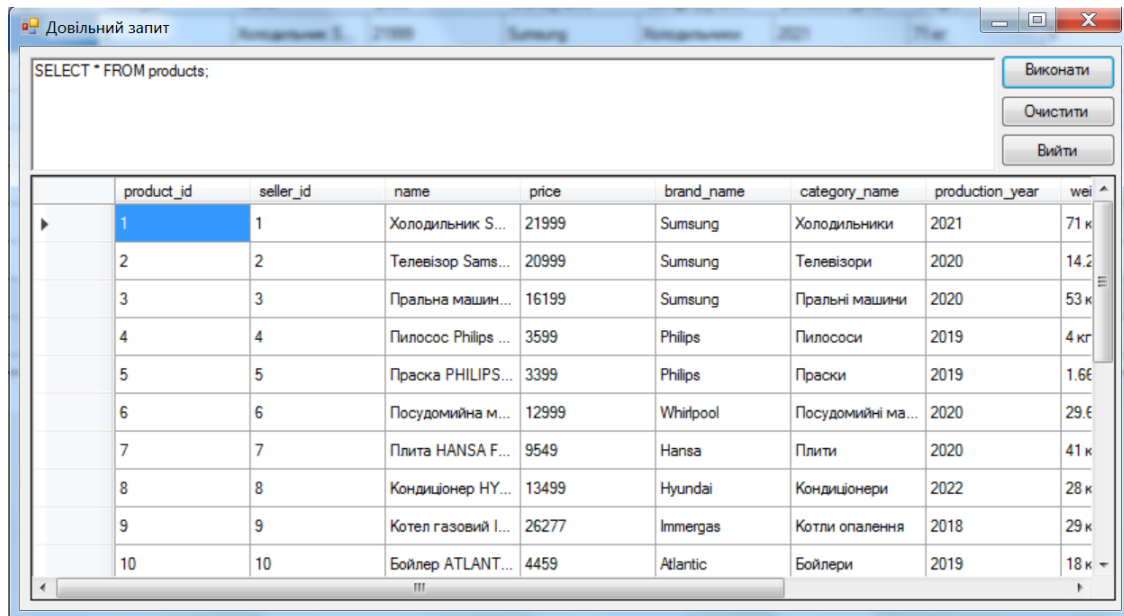


Рисунок 19 – Форма редактору довільного запиту

Адміністратор в основному буде керувати базою даних магазину за допомогою графічного інтерфейсу. Але також додана можливість керування базою даних за допомогою довільного SQL-запиту.

4.6 Опис задачі автоматизації

Задача автоматизації полягає в підборі клієнта-переможця акційного розіграшу “Клієнт року”, враховуючи критерії. Критерій для клієнта-переможця – це клієнт, який замовив найдорожчий товар, який реалізується за допомогою SQL-запитів і коду-оброблювача. Клієнт року отримає великі знижки в інтернет-магазині. Актуальність це полегшення підбору клієнту для акційного розіграшу, який збільшить продажі інтернет-магазину. Складність в написанні коду і SQL-запитів, які враховують формулу справедливого вибору у реальному часі для підбору клієнта-переможця. Інтерфейс користувача задачі автоматизації і функції документобігу, а саме формування звіту “Клієнт року”

реалізован за допомогою Button і ComboBox з випадаючими значеннями з бази даних (див. рис. 20).

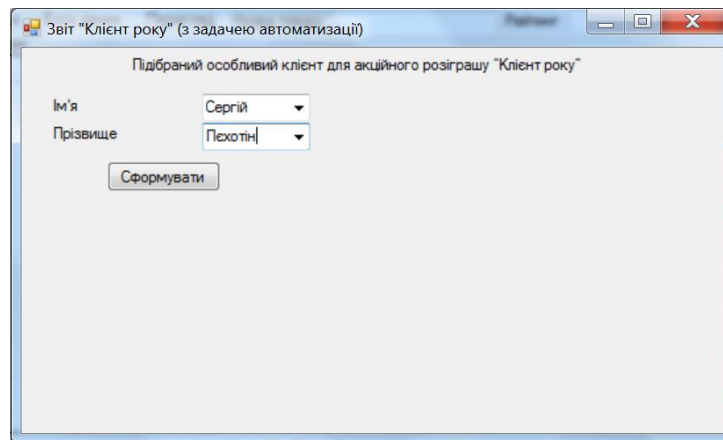


Рисунок 20 – Форма задачі автоматизації підбору клієнта-переможця акційного розіграшу “Клієнт року” і формування в результаті звіту “Клієнт року”

Наведемо результат задачі автоматизації, тобто сформований звіт “Клієнт року” (див. рис. 21).

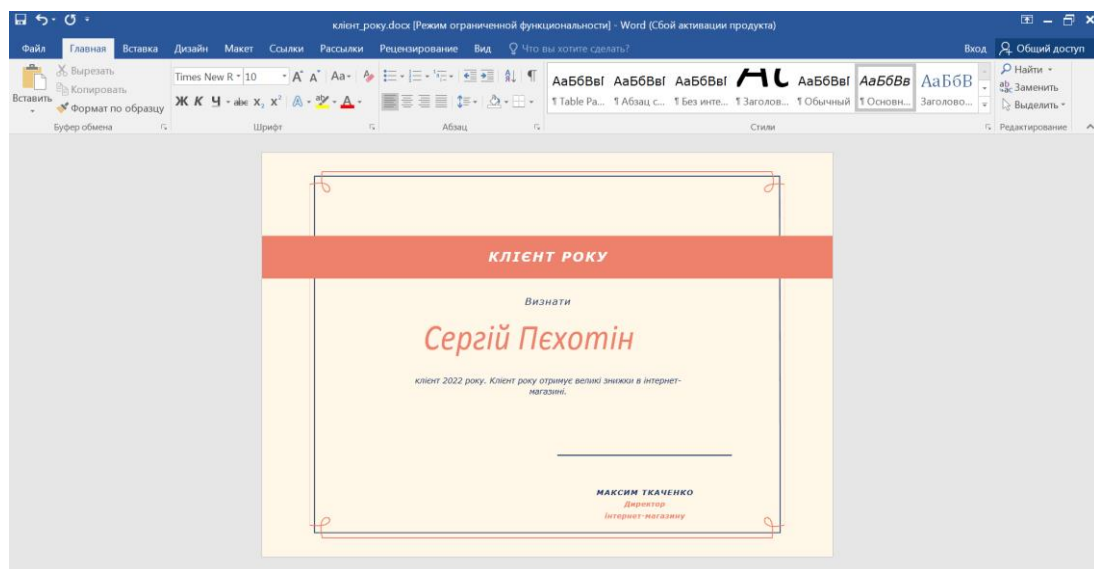


Рисунок 21 – Результат задачі автоматизації - це сформований звіт “Клієнт року”

Отже, був описан реалізований інтерфейс і функціонал програми. SQL відіграє ключову роль в коді програми.

ВИСНОВКИ

В результаті курсового проектування отриман фрагмент інформаційної системи управління вмістом сайту інтернет-магазину побутової техніки, створений на основі бази даних. Область використання програми – це керування вмістом сайту інтернет-магазину. Запланований базовий функціонал в постановці задачі розроблен. Розроблена інформаційна система має наступний функціонал: додавання, редагування, видалення, збереження, перегляд контенту, пошук, фільтрація, сортування товарів, отримання статистики, задача автоматизації, формування звітів. Обрана і розроблена реляційна база даних. Закріплені на практиці всі рівні моделювання реляційної бази даних: концептуальне, логічне та фізичне. Використано мову SQL для роботи з базою даних, яка відіграє суттєве значення в коді програми. Використано СУБД Microsoft SQL Server для розробки інформаційної системи на основі бази даних. Користь інформаційної системи в спрощенні управління вмістом сайту інтернет-магазину. Засвоїв урок тайм-менеджменту при створені наступних курсових проектів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Дейт К. Дж. Введение в системы баз данных, 8-е изд. - М.: Изд. дом «Вильямс», 2008, - 1046 с.
2. Мартин Грабер. SQL. - К.: Ид-во “ЛОРИ”, 2003. - 644 с.
3. Гарсия-Молина, Гектор, Ульман, Джеффри, Уидом, Дженнифер, Системы баз данных. Полный курс.: Пер. С англ.. – М.:Издательский дом „Вильямс”, 2003. – 1088 с.: ил. – Парал. тит. англ.
4. Теория и практика построения баз данных. 8-е изд. / Д.Кренке. – СПб.: Питер, 2003. – 800 с.: ил.
5. SQL Tutorial. *W3Schools Online Web Tutorials*. URL: <https://www.w3schools.com/sql/> (дата звернення: 14.12.2022).
6. Rozetka.com.ua. Rozetka.com.ua. URL: <https://rozetka.com.ua/> (дата звернення: 14.12.2022).
7. Amazon.com. *Amazon.com*. URL: <https://www.amazon.com/> (дата звернення: 14.12.2022).