Міністерство освіти і науки України

Харківський радіотехнічний фаховий коледж

Циклова комісія «Комп’ютерних наук»

**КУРСОВА РОБОТА**

з навчальної дисципліни «БАЗИ ДАНИХ»

на тему ««ПРОЄКТУВАННЯ ТА РОЗРОБКА БД «Магазин взуття»»

Виконав:

студент гр. ПІ-331

Лучанінов Петро Олександрович

Керівник роботи:

Ахмедзянова О.А.

м. Харків – 2024 рік

Харківський радіотехнічний фаховий коледж

Циклова комісія КОМП’ЮТЕРНИХ НАУК

Спеціальність 121 «Інженерія програмного забезпечення»

Курс - 3 Група – ПІ - 331 Семестр - 6

Завдання на курсову роботу

студента *Лучанінова Петра Олександровича*

Тема курсової роботи: «*Проєктування та розробка БД «Магазин взуття»»*

ОСНОВНІ ДАНІ: *База даних створюється для зберігання та обробки інформації про магазини взуття, їх асортимент, бренди та послуги, що надаються. Основна мета проєкту полягає у створенні зручного та надійного інструменту для управління взуттєвою торгівлею та покращення якості обслуговування клієнтів.*

Склад пояснювальної записки:

Вступ

1. Аналіз предметної області і постановка задачі
2. Проєктування реляційної БД
   1. Побудова ER-діаграми
   2. Побудова схеми реляційної моделі даних
   3. Вибір СУБД та опис фізичної моделі даних
   4. Ескіз інтерфейсу користувача
3. Опис програмної реалізації

3.1 Опис структури бази даних

3.2 Опис задач автоматизації та інтерфейсу користувача

Висновки

Перелік використаних джерел

Дата видачі завдання: 17 січня 2024р

Дата здачі виконаної роботи: \_\_\_\_\_\_\_\_\_\_\_\_\_

Керівник роботи /О.А.Ахмедзянова /

Розглянуто на засіданні ЦК

Голова ЦК /Федосєєва А.О./

ЗАЛІКОВИЙ ЛИСТ

Харківський радіотехнічний фаховий коледж

Циклова комісія КОМП’ЮТЕРНИХ НАУК

Спеціальність 121 «Інженерія програмного забезпечення»

Курс - 3 Група – ПІ - 331 Семестр - 6

Курсова робота

студента *Лучанінова Петра Олександровича*

на тему «*Проєктування та розробка БД «Магазин взуття»»*

ДОПУЩЕНИЙ ДО ЗАХИСТУ

Керівник роботи /Ахмедзянова О.А. /

ОЦІНКА\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| Члени комісії: |  | /Ахмедзянова О.А./ |
|  |  | /Радченко О.П. / |
|  |  | /ПироженкоВ.В./ |

« \_\_\_» «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_» 2024 р

ЗМІСТ

ВСТУП 5

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ 6

2 ПРОЄКТУВАННЯ БАЗИ ДАНИХ 12

2.1 Побудова діаграм 12

2.2 Побудова реляційної моделі даних 15

2.3 Вибір СУБД та опис фізичної моделі бази даних 20

2.4 Ескіз інтерфейсу користувача 27

3 ОПИС ПРОГРАММНОЇ РЕАЛІЗАЦІЇ 30

3.1 Опис структури бази даних 30

3.2 Опис задач автоматизації та інтерфейсу користувача 37

ВИСНОВКИ 48

ПЕРЕЛІК ПОСИЛАНЬ 49

# ВСТУП

У сучасному світі інформаційні технології відіграють ключову роль у розвитку бізнесу. Особливо це стосується роздрібної торгівлі, де ефективне управління інформацією про товари, клієнтів та операції є критично важливим для досягнення конкурентоспроможності та задоволення потреб споживачів. Одним з важливих інструментів для досягнення цих цілей є бази даних, які дозволяють організувати, зберігати та аналізувати великі обсяги інформації.

Тема курсової роботи «Проектування та розробка бази даних «Магазин взуття»» є актуальною, оскільки ефективна система управління даними здатна значно покращити роботу магазину, підвищити якість обслуговування клієнтів та оптимізувати бізнес-процеси. Мета цієї роботи полягає у створенні бази даних, яка забезпечить зручний інтерфейс для роботи з інформацією про взуття, бренди, замовлення та клієнтів, а також автоматизує ряд ключових функцій магазину.

У рамках курсової роботи буде проведено аналіз вимог до системи, спроектовано структуру бази даних, створено схему даних та розроблено основні елементи інтерфейсу користувача. Також буде розглянуто питання забезпечення цілісності даних, їх безпеки та продуктивності системи.

Результатом виконання даної роботи стане розроблена та впроваджена база даних, яка дозволить ефективно управляти інформацією у магазині взуття та забезпечить високий рівень обслуговування клієнтів.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

Метою даної курсової роботи є ретельний аналіз специфіки роботи магазину взуття та розробка якісного продукту, який буде відповідати нормам та заданому функціоналу обраної предметної області.

Постановка задачі:

1. Розробка реляційної бази даних для збереження інформації про асортимент взуття, постачальників, клієнтів, та інші необхідні дані.
2. Реалізація функціоналу для введення, редагування та видалення інформації з бази даних.
3. Проведення перевірки коректності введених даних, забезпечення їх консистентності та валідності.
4. Реалізація системи пошуку даних за різними параметрами для забезпечення ефективного використання бази даних.
5. Розробка інтерфейсу користувача, який буде інтуїтивно зрозумілим та зручним для використання.

Вимоги до розробки:

– дизайн інтерфейсу повинен бути сучасним, легким для розуміння та не надто вимогливим до апаратного забезпечення.;

– додаток повинен забезпечувати зручний пошук інформації, чітко та лаконічно надавати цю інформацію користувачеві та гарантувати цілісність даних.

– додаток повинен мати можливість фільтрації та сортування результатів пошуку для підвищення зручності користування.

– додаток повинен виконувати функції підрахунку, потрібні для полегшення роботи магазину.

– повинна бути забезпечена висока якість коду шляхом дотримання найкращих практик програмування та використання стандартів якості.

– наприкінці курсової роботи повинне бути проведене комплексне тестування додатку з метою виявлення та усунення помилок та недоліків.

Предметною областю курсової роботи є магазин взуття.

Магазин взуття займається продажем різноманітного взуття різних брендів, що має одне приміщення в центрі міста Харкова, для якого і необхідно розробити систему.

Основними видами діяльності компанії являються: продаж різноманітних взуттєвих товарів, консультація та обслуговування клієнтів, організація розпродажів та акцій, співпраця з постачальниками, аналіз даних про продажі та популярність товарів для прийняття стратегічних рішень щодо асортименту та маркетингових кампаній.

База даних для магазину взуття створюється для інформаційного обслуговування співробітників магазину, ефективного зберігання, організації та керування інформацією про асортимент взуття, клієнтів, постачальників та операції, що дозволяє полегшити облік товарів, оптимізувати процеси продажу, підвищити точність інвентаризації та надати зручний інтерфейс для ведення бізнесу.

На підприємстві працюють наступні співробітники:

1. Менеджер магазину: керує денними операціями магазину, виконує управління персоналом та розподіл обов'язків, планує та веде інвентаризацію, відповідає за планування та впровадження стратегій продажу та акцій, забезпечує високий рівень обслуговування клієнтів та вирішує проблеми, відповідає за вибір постачальників та укладання угод з ними, здійснює планування та контроль закупівель для забезпечення належного запасу товарів.
2. Продавці-консультанти: вітають та обслуговують покупців, надають консультації по моделям взуття та розмірам, стежать за наявністю товару та поповненням асортименту, підтримують порядок в приміщенні магазину.
3. Касири: здійснюють операції з оплати за касовими апаратами, ведуть облік грошових потоків, ведуть звітність, надають клієнтам квитанції та інформацію про політику повернення.
4. Складські працівники: приймають та розміщують товар на складі, забезпечують організацію та підтримку порядку на складі, комплектують замовлення для магазину.

Співробітники магазину не можуть вільно вводити і редагувати дані, що зберігаються в базі даних. Додаток буде мати дві ролі користувачів: адміністратор та співробітник.

Магазин отримує товар від постачальників, з якими укладено договори, які зберігаються в паперовому вигляді. Для відстеження постачань використовуються накладні, які зберігаються в базі даних.

Оскільки в магазина відсутня онлайн-система продажу та наявне лише одне приміщення у центрі міста Харків, тому він приймає тільки один вид валюти – гривні.

У магазині є три відділи взуття: чоловіче, жіноче, унісекс.

Взуття має наступні характеристики: стать, тип, бренд, розмір, колір, матеріал, з якого виготовлене, ціну та розділяється за сезонами (зима, весна, літо, осінь, демісезон, всесезон).

Магазин має систему знижок для своїх постійних клієнтів у вигляді дисконтних карт. Інформація про клієнтів, така як їх ім'я та контактні дані, зберігається для індентифікації. Для отримання статусу постійного клієнта необхідно здійснити покупки на суму від 3000 грн, після чого касир реєструє клієнта і видає йому унікальну дисконтну картку з ідентифікаційним номером.

При отриманні дисконтної картки клієнт одразу отримує 5% знижку. При подальших покупках з дисконтною карткою, клієнту нараховуватись знижка в залежності від загальної суми всіх покупок: 5% при загальній сумі від 5000 грн, 7% при загальній сумі від 7000 грн, 10% при загальній сумі від 10000 грн. Максимальна можлива знижка становить 15%, від 15000грн відповідно.

Якщо постійний клієнт використовує знижку на дисконтній картці, накопичена сума та знижка скидаються до нуля. На дисконтну картку зачислюється одноразова знижка від загальної суми всіх покупок.

Наведемо перелік таблиць та їх атрибутів, які потрібно зберігати в базі даних:

Товар:

– Код товару – унікальне, числове, обов’язкове;

– Назва товару – текстове значення, обов’язкове;

– Стать – чоловіче, жіноче;

– Тип – текстове, обов’язкове;

– Бренд – текстове, обов’язкове;

– Розмір – числове, обов’язкове;

– Колір – текстове, обов’язкове;

– Матеріал – текстове, обов’язкове;

– Сезон – зима, весна, літо, осінь, всесезон, демісезон;

– Ціна за одиницю – числове значення, обов’язкове;

– Кількість одиниць на складі – числове значення, обов’язкове;

Постійний клієнт:

– Код клієнта – унікальне, числове, обов’язкове;

– Ім’я клієнта – текстове, обов’язкове;

– Прізвище клієнта– текстова, обов’язкове;

– По батькові клієнта – текстова, обов’язкове;

– Номер телефону – числове(10) обов’язкове значення, у форматі 0957653464;

– Електрона пошта – текстове значення;

– Код дисконтної картки – унікальне, числове;

– Накопичена сума – числове;

Співробітник:

– Код співробітника – унікальне, числове, обов’язкове;

– Ім’я співробітника – текстове, обов’язкове;

– Прізвище співробітника – текстова, обов’язкове;

– По батькові співробітника – текстова, обов’язкове;

– Посада: менеджер магазину, продавець–консультант, касир, складський працівник, менеджер маркетингу, менеджер закупівель;

– Номер телефону – числове(10), обов’язкове, у форматі 0957653464;

– Електрона пошта – текстове значення;

– Адреса – текстове значення, обов’язкове;

– Стать: чоловік, жінка;

– Дата народження – дата, обов’язкове.

Постачальник:

– Код постачальника – унікальне, числове, обов’язкове;

– Назва фірми – текстове значення, обов’язкове;

– Код ЄДРПОУ – унікальне, числове(8), обов’язкове;

– Перший номер телефону – числове(10), обов’язкове;

– Другий номер телефону (якщо є) – числове(10) значення;

– Адреса – текстове значення;

– Розрахунковий рахунок – унікальне, текстове;

Продаж:

– Код продажу – унікальне, числове, обов’язкове;

– Код товару – числове, обов’язкове;

– Код співробітника – числове, обов’язкове;

– Код постійного клієнта (якщо використовується дисконтна картка) – числове значення;

– Кількість придбаного товару – числове, обов’язкове;

– Дата та час покупки – дата, обов’язкове;

Накладна:

– Код накладної – унікальне, числове, обов’язкове;

– Код співробітника – числове, обов’язкове;

– Код постачальника – числове, обов’язкове;

– Дата постачання – дата, обов’язкове;

Список постачання:

– Код постачання – унікальне, числове, обов’язкове;

– Код товару – унікальне, числове, обов’язкове;

– Код накладної – числове, обов’язкове;

– Ціна за одиницю – числове значення, обов’язкове;

– Кількість одиниць – числове значення, обов’язкове;

Повернення:

– Код повернення – унікальне, числове, обов’язкове;

– Код продажу – числове, обов’язкове;

– Дата повернення – дата, обов’язкове;

– Причина повернення – текстове, обов’язкове;

Функції програми:

– перегляд інформації про товари, клієнтів, постачальників, співробітників, продаж, постачання, накладні, повернення та дисконтні картки;

– можливість додавати, видаляти та редагувати дані;

– пошук в таблицях за їх атрибутами;

– нарахунок суми на дисконту картку, при покупці з нею (якщо не використовуєть знижка);

– розрахунок вартості при проведенні оплати;

– розрахунок днів при поверненні товару;

– формування чеку.

При введенні даних необхідно врахувати такі обмеження:

– постійний клієнт не може мати дві дисконтні картки (за винятком особливих випадків);

Таким чином при дослідженні предметної області – магазину взуття, була визначена основна інформація яку необхідно відображати в базі даних, а також встановлені функції що повинні бути реалізовані в додатку.

# 2 ПРОЄКТУВАННЯ БАЗИ ДАНИХ

## 2.1 Побудова діаграм

Модель "сутність-зв'язок" представляє собою модель даних, що дозволяє описувати концептуальні схеми через узагальнені конструкції блоків. ER-модель слугує інструментом для опису моделей даних. Існує ряд моделей для представлення знань, але одним з найзручніших інструментів уніфікованого представлення даних, незалежного від програмного забезпечення, що його реалізує, є модель «сутність-зв'язок».

Модель "сутність-зв'язок" є результатом систематичного процесу, який описує та визначає певну предметну область. Вона не визначає сам процес, а лише візуалізує його. Дані представлені у вигляді компонентів (сутностей), які пов'язані між собою конкретними зв'язками, що виражають залежності і вимоги між ними.

ER-діаграми (діаграми сутність-зв'язок) представляють собою графічний інструмент для змоделювання та опису взаємозв'язків між сутностями у базі даних. Вони служать для візуалізації структури даних та зв'язків між різними сутностями у системі. ER-діаграми відображають сутності (об'єкти), їх атрибути (характеристики) та зв'язки між ними.

У рамках курсової роботи створюється ER-діаграма для візуалізації структури бази даних, опису зв'язків між таблицями, визначення атрибутів сутностей та моделювання потоку даних в системі. Ця діаграма допомагає зрозуміти взаємозв'язки між різними компонентами системи і спрощує розробку та розуміння бази даних.

На ER-діаграмі сутності представлені у вигляді прямокутників, а зв'язки між ними - у вигляді стрілок або ліній. Кожна сутність має свої атрибути, які відображаються всередині еліпсів. Зв'язки між сутностями та атрибутами показуються стрілками, що з'єднують відповідні прямокутники.

З урахуванням особливостей предметної області та атрибутів, спроектуємо діаграму сутність-зв’язок для бази даних (рисунок 2.1).

Зображення, що містить текст, схема, чорно-білий, почерк

Автоматично згенерований опис

Рисунок 2.1 – ER-діаграма

Діаграми варіантів використання (usecase diagrams) використовуються для відображення сценаріїв використання системи (usecases) та користувачів системи (actors), які використовують її функції.

Актори на діаграмі варіантів використання позначаються символом людини, а варіанти використання – еліпсом. Актори та варіанти використання поєднуються напрямленою асоціацією (unidirectional association) – стрілкою, що спрямована від актора до варіанта використання. Також актори можуть поєднуватися з використанням зв’язків узагальнення.

В процесі аналізу предметної області було виділено два типи користувачів (акторів) – адміністратор та співробітник.

USE CASE відображена на рисунку 2.2.

Зображення, що містить ескіз, малюнок, схема, коло

Автоматично згенерований опис

Рисунок 2.2 – USE CASE

## 2.2 Побудова реляційної моделі даних

Реляційна модель даних - це структурований підхід до організації даних у вигляді таблиць з рядками і стовпцями. Кожна таблиця представляє собою відношення, а кожний рядок в таблиці - кортеж (запис), що представляє конкретний запис даних. Кожен стовпець в таблиці відповідає певному атрибуту (полю) даних.

Головний постулат баз даних – дані не повинні бути надлишковими. Для цього вводиться поняття «нормалізація».

Нормалізація в базі даних – це процес усунення повторюваної інформації. У першій нормальній формі (1НФ), кожна таблиця відповідає основним властивостям, має визначене ключове поле та відповідає таким вимогам для сутностей:

– всі поля є атомарними (неподільними);

– всі стовпці є однорідними;

– кожен рядок є унікальним;

– таблицю можна редагувати в будь-якому напрямку.

Для досягнення 1НФ створюються прямокутні таблиці (один атрибут – один стовпець), складні атрибути розбиваються на прості, а багатозначні атрибути виділяються в окремі сутності, такі як "Категорія". Усі таблиці мають зовнішні та внутрішні ключові поля.

Друга нормальна форма (2НФ) вимагає, щоб всі атрибути були функціонально повно залежними від ключа, а не від його окремих частин. Відношення з первинним ключем, який включає лише один атрибут, завжди відповідає 2НФ.

Третя нормальна форма (3НФ) виключає транзитивні залежності між неключовими атрибутами. Крім того, схема повинна відповідати всім вимогам 2НФ.

Схеми відношень, що зведені до 3НФ представлені в таблицях 2.1-2.11.

Таблиця 2.1 - Відношення «Товар»

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Код товару | Назва товару | Стать | Код типу | Код бренду | Код матеріалу | Сезон | Ціна за одиницю |
|  |  |  |  |  |  |  |  |

Для відношення Товар додано декілька таблиць з характеристиками:

Таблиця 2.2 - Відношення «Бренд»

|  |  |
| --- | --- |
| Код бренду | Назва бренду |
|  |  |

Таблиця 2.3 - Відношення «Колір»

|  |  |
| --- | --- |
| Код кольору | Назва кольору |
|  |  |

Таблиця 2.4 - Відношення «Тип»

|  |  |
| --- | --- |
| Код типу | Назва типу |
|  |  |

Таблиця 2.5 - Відношення «Розмір»

|  |  |
| --- | --- |
| Код розміру | Розмір |
|  |  |

Таблиця 2.6 - Відношення «Матеріал»

|  |  |
| --- | --- |
| Код матеріалу | Назва матеріалу |
|  |  |

Таблиця 2.7 - Відношення «Товар\_кількість»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код\_Товар\_Кількість | Код товару | Код розміру | Код кольору | Кількість |
|  |  |  |  |  |

Таблиця 2.8 - Відношення «Постійний клієнт»

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Код постійного клієнта | Ім'я постійного клієнта | Прізвище постійного клієнта | По батькові постійного клієнта | Номер телефону | Електрона пошта | Код дисконтної картки | Накопичена сума |
|  |  |  |  |  |  |  |  |

Таблиця 2.9 - Відношення «Співробітник»

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Код співробітника | Ім’я співробітника | Прізвище співробітника | По батькові співробітника | Посада | Номер телефону | Електрона пошта | Адреса | Стать | Дата народження |
|  |  |  |  |  |  |  |  |  |  |

Таблиця 2.10 - Відношення «Постачальник»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Код  постачальника | Назва фірми | Код ЄДРПОУ | Перший номер телефону | Другий номер телефону | Адреса | Розрахунковий рахунок |
|  |  |  |  |  |  |  |

Таблиця 2.5 - Відношення «Продаж»

|  |  |  |  |
| --- | --- | --- | --- |
| Код продажу | Код співробітника | Код постійного  клієнта | Дата та час покупки |
|  |  |  |  |

Оскільки в одній покупці може бути декілька товарів з різною кількістю, тоді створимо відношення «Список товарів» для відношення «Продаж», в якому буде зберігатися код товару та його кількість.

Таблиця 2.6 - Відношення «Список куплених товарів»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код списку куплених товарів | Код продажу | Код\_Товар\_Кількість | Ціна | Кількість |
|  |  |  |  |  |

Таблиця 2.7 - Відношення «Повернення»

|  |  |  |  |
| --- | --- | --- | --- |
| Код повернення | Код продажу | Дата повернення | Причина повернення |
|  |  |  |  |

Оскільки в одному повернені може бути повернуті не всі товари з різною кількістю, тоді створимо відношення «Список повернення» для відношення «Повернення», в якому буде зберігатися код товару та його кількість.

Таблиця 2.8 - Відношення «Список повернення»

|  |  |  |  |
| --- | --- | --- | --- |
| Код списку повернення | Код повернення | Код\_Товар\_Кількість | Кількість |
|  |  |  |  |

Таблиця 2.9 - Відношення «Список постачань»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код постачання | Код накладної | Код\_Товар\_Кількість | Ціна за одиницю | Кількість одиниць |
|  |  |  |  |  |

Таблиця 2.10 - Відношення «Накладна»

|  |  |  |  |
| --- | --- | --- | --- |
| Код накладної | Код співробітника | Код постачальника | Дата постачання |
|  |  |  |  |

# 2.3 Вибір СУБД та опис фізичної моделі бази даних

Формування фізичної структури бази даних є наступним кроком після створення логічної моделі бази даних. На цьому етапі конвертується логічна модель реляційної бази даних у фізичні об'єкти за допомогою команд створення, що визначають внутрішню схему бази даних. Також може бути створена зовнішня схема для відображення перспективи користувачів на дані в базі даних. Отриманий скрипт може бути використаний для створення фізичної бази даних.

Основна мета полягає в перетворенні логічної моделі реляційної бази даних у послідовність команд SQLite для створення реляційних об'єктів. Це означає відображення відношень логічної моделі (предметна область у нормалізованій формі на ER-діаграмах) у таблиці та індекси реляційної бази даних.

Для даної курсової роботи була вибрана Система Керування Базами Даних MySQL. MySQL - це відкрита реляційна система управління базами даних, яка широко використовується у веб-розробці. Вона підтримує стандартні SQL-запити та володіє великою спільнотою користувачів. MySQL дозволяє створювати бази даних, таблиці, індекси та здійснювати роботу з ними за допомогою SQL-команд.

Після ретельного аналізу різних систем керування базами даних було вирішено використовувати MySQL через його широкі можливості та стабільність. Проте слід зазначити, що для використання MySQL потрібно мати сервер, на відміну від SQLite, який є вбудованою системою. Однак MySQL володіє значними перевагами, такими як підтримка збережених процедур, вбудована підтримка unicode та придатність для додатків, які часто звертаються до бази даних.

Фізична модель даних, відображена в таблиці 2.11, буде адаптована до вимог MySQL.

Таблиця 2.11 – Фізична модель бази даних

| **Назва стовпця** | **Тип даних** | **Обмеження** | **Підпис** |
| --- | --- | --- | --- |
| Таблиця «Товар» | | | |
| product\_id | Unsigned smallint | Primary key, auto increment, unigue | Код товару |
| product\_name | Varchar(100) | Not null, unique | Назва товару |
| product\_gender | Enum | чоловіче, жіноче, унісекс | Стать |
| type\_id | Unsigned smallint | Foreign key, not null | Код типу |
| brand\_id | Unsigned smallint | Foreign key, not null | Код бренду |
| material\_id | Unsigned smallint | Foreign key, not null | Код матеріалу |
| product\_season | Enum | зима, весна, літо, осінь, демісезон, всесезон | Сезон |
| product\_price | Float | Not null | Ціна за одиницю |
| Таблиця «Товар\_кількість» | | | |
| product\_quantity\_id | Unsigned smallint | Primary key, auto increment | Код товар\_кількість |
| product\_id | Unsigned smallint | Foreign key, not null | Код товару |
| size\_id | Unsigned smallint | Foreign key, not null | Код розміру |
| color\_id | Unsigned smallint | Foreign key, not null | Код кольору |
| product\_quantity | Unsigned smallint | Not null, default 0 | Кількість |
| Таблиця «Бренд» | | | |
| brand\_id | Unsigned smallint | Primary key, auto increment | Код бренду |
| brand\_name | Varchar(100) | Not null | Назва бренду |
| Таблиця «Колір» | | | |
| color\_id | Unsigned smallint | Primary key, auto increment | Код коліру |
| color\_name | Varchar(100) | Not null | Назва коліру |
| Таблиця «Тип» | | | |
| type\_id | Unsigned smallint | Primary key, auto increment | Код типу |
| type\_name | Varchar(100) | Not null | Назва типу |
| Таблиця «Розмір» | | | |
| size\_id | Unsigned smallint | Primary key, auto increment | Код розміру |
| size\_name | Varchar(100) | Not null | Назва розміру |
| Таблиця «Матеріал» | | | |
| material\_id | Unsigned smallint | Primary key, auto increment | Код матеріалу |
| material\_name | Varchar(100) | Not null | Назва матеріалу |
| Таблиця «Постійний клієнт» | | | |
| customer\_id | Unsigned smallint | Primary key, auto increment | Код постійного клієнта |
| customer\_first\_name | Varchar(50) | Not null | Ім’я постійного клієнта |
| customer\_surname | Varchar(50) | Not null | Прізвище постійного клієнта |
| сustomer  \_middle\_name | Varchar(50) | Not null | По батькові постійного клієнта |
| customer \_phone\_number | Bigint | Not null | Номер телефону |
| customer\_email | Varchar(50) |  | Електронна пошта |
| discount\_card\_id | Unsigned smallint | Unique | Код дисконтної картки |
| discount\_card\_  accumulation | Float | default 0 | Накопичена сума |
| Таблиця «Співробітник» | | | |
| employee \_id | Unsigned smallint | Primary key, auto increment | Код співробітника |
| employee \_first\_name | Varchar(50) | Not null | Ім’я співробітника |
| employee \_surname | Varchar(50) | Not null | Прізвище співробітника |
| employee \_middle\_name | Varchar(50) | Not null | По батькові співробітника |
| employee \_position | Enum | менеджер, продавець-консультант, касир, складський працівник | Посада |
| employee \_phone\_number | Bigint | Not null | Номер телефону |
| employee \_email | Varchar(50) | Not null | Електронна пошта |
| employee \_address | Varchar(255) | Not null | Адреса |
| employee \_gender | Enum | чоловік, жінка | Стать |
| employee \_date\_of\_birth | Date | Not null | Дата народження |
| Таблиця «Постачальник» | | | |
| supplier \_id | Unsigned smallint | Primary key, auto increment | Код постачальника |
| supplier \_name | Varchar(255) | Not null | Назва фірми |
| supplier \_edrpou\_code | Bigint | Not null | Код ЄРДПОУ |
| supplier \_fphone\_number | Bigint | Not null | Перший номер телефону |
| supplier \_sphone\_number | Bigint | Not null | Другий номер телефону |
| supplier\_address | Varchar(255) | Not null | Адреса |
| supplier \_сurrent\_account | Bigint | Not null, unique | Розрахунковий рахунок |
| Таблиця «Продаж» | | | |
| sales\_id | Unsigned int | Primary key, auto increment | Код продажу |
| employee \_id | Unsigned smallint | Foreign key, not null | Код співробітника |
| customer\_id | Unsigned smallint | Foreign key, not null | Код постійного клієнта |
| sales\_time | Datetime | Not null | Дата та час покупки |
| Таблиця «Список куплених товарів» | | | |
| sales\_list\_id | Unsigned int | Primary key, auto increment | Код списку куплених товарів |
| sales\_id | Unsigned int | Foreign key, not null | Код продажу |
| product\_quantity\_id | Unsigned smallint | Foreign key, Not null | Код товару |
| sales\_list\_quantity | Unsigned smallint | Not null, default 1 | Кількість |
| sales\_list\_price | Unsigned smallint | Not null, default 1 | Ціна |
| Таблиця «Повернення» | | | |
| product\_return\_id | Unsigned int | Primary key, auto increment | Код повернення |
| sales\_id | Unsigned smallint | Foreign key, not null | Код продажу |
| return\_ reason | Varchar(255) | Not null | Причина повернення |
| Таблиця «Список повернення» | | | |
| product\_return\_list\_id | Unsigned int | Primary key, auto increment | Код списку куплених товарів |
| product\_return\_id | Unsigned int | Foreign key, not null | Код продажу |
| product\_quantity\_id | Unsigned smallint | Foreign key, Not null | Код товару |
| product\_return\_  list\_quantity | Unsigned smallint | Not null, default 1 | Кількість |
| Таблиця «Накладна» | | | |
| invoice\_id | Unsigned int | Primary key, auto increment | Код накладної |
| employee \_id | Unsigned smallint | Foreign key, not null | Код співробітника |
| supplier\_id | Unsigned smallint | Foreign key, not null | Код постачальника |
| supply\_time | Datetime | Not null | Дата постачання |
| Таблиця «Список постачань» | | | |
| supply\_list\_id | Unsigned int | Primary key, auto increment | Код постачання |
| invoice\_id | Unsigned int | Foreign key, not null | Код накладної |
| product\_id | Unsigned smallint | Foreign key, Not null | Код товару |
| supply\_list\_price | Float | Not null | Ціна за одиницю |
| supply\_list\_quantity | Unsigned smallint | Not null, default 1 | Кількість одиниць |

# 2.4 Ескіз інтерфейсу користувача

Для забезпечення зручності роботи з даними БД та інформацією, що зберігається в таблицях, необхідно розробити додаток. Перед його створенням потрібно розробити попередні прототипи інтерфейсу на базовому рівні.

Інтерфейс користувача є критично важливою складовою будь-якої інформаційної системи, оскільки саме через нього користувач взаємодіє з базою даних. Для проєкту «Магазин взуття» інтерфейс користувача розробляється з урахуванням принципів зручності, інтуїтивності та функціональності.

Основні компоненти інтерфейсу користувача для бази даних «Магазин взуття» включають наступні елементи:

1. Вікно авторизації

Вікно логіну призначене для автентифікації користувачів перед входом у систему. Воно зазвичай містить наступні елементи:

* Поле для вводу імені користувача
* Поле для вводу пароля

Ескіз вікна авторизації додатку зображено на рисунку 2.3.

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Рисунок 2.3 – Ескіз вікна авторизації

1. Головне вікно

На головному вікні розміщено меню навігації, яке дозволяє користувачу швидко перейти до основних розділів системи.

Ескіз вікна головного вікна додатку зображено на рисунку 2.4.

Зображення, що містить текст, знімок екрана, число

Автоматично згенерований опис

Рисунок 2.4 – Ескіз головного вікна

1. Вікно додання(зміни) об’єкта до БД:

Вікно додання або зміни об'єкта до бази даних призначене для введення нової інформації або редагування існуючої. Вікно буде відрізнятися за полями в залежності від таблиці БД. Ескіз вікна авторизації додатку зображено на рисунку 2.5.

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Рисунок 2.5 – Ескіз вікна додання(зміни) об’єкту

1. Вікно кошику

Вікно кошика призначене для відображення та управління товарами, які користувач планує придбати. Ескіз вікна авторизації додатку зображено на рисунку 2.6.

Зображення, що містить знімок екрана, текст, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис

Рисунок 2.6 – Ескіз вікна кошику

# 3 ОПИС ПРОГРАММНОЇ РЕАЛІЗАЦІЇ

## 3.1 Опис структури бази даних

База даних «Shoe\_store\_DB» створена в СКБД MySQL. БД використовується для зберігання даних про товари, покупців, співробітників та постачання.

Створення бази даних:

create database Shoe\_store\_DB default character set utf8;

Таблиця "товар" призначена для зберігання інформації про товари. SQL-скрипт для створення таблиці:

create table product (

product\_id smallint unsigned auto\_increment primary key,

product\_name varchar(100) not null unique,

product\_gender enum('чоловіче', 'жіноче', 'унісекс'),

type\_id smallint unsigned not null,

brand\_id smallint unsigned not null,

material\_id smallint unsigned not null,

product\_season enum('зима', 'весна', 'літо', 'осінь', 'демісезон', 'всесезон'),

product\_price float not null,

foreign key (type\_id) references product\_type(type\_id),

foreign key (brand\_id) references brand(brand\_id),

foreign key (material\_id) references material(material\_id)

);

Таблиця "товар\_кількість" використовується для запису кількості товарів у різних розмірах та кольорах. SQL-скрипт для створення таблиці:

create table product\_quantity (

product\_quantity\_id smallint unsigned auto\_increment primary key,

product\_id smallint unsigned not null,

size\_id smallint unsigned not null,

color\_id smallint unsigned not null,

product\_quantity smallint unsigned not null default 0,

foreign key (product\_id) references product(product\_id),

foreign key (size\_id) references size(size\_id),

foreign key (color\_id) references color(color\_id)

);

Таблиця "бренд" містить дані про різні бренди товарів. SQL-скрипт для створення таблиці:

create table brand (

brand\_id smallint unsigned auto\_increment primary key,

brand\_name varchar(100) not null

);

Таблиця "колір" зберігає інформацію про різні кольори товарів. SQL-скрипт для створення таблиці:

create table color (

color\_id smallint unsigned auto\_increment primary key,

color\_name varchar(100) not null

);

Таблиця "тип" використовується для категоризації товарів за їхнім типом. SQL-скрипт для створення таблиці:

create table product\_type (

type\_id smallint unsigned auto\_increment primary key,

type\_name varchar(100) not null

);

Таблиця "розмір" містить дані про різні розміри товарів. SQL-скрипт для створення таблиці:

create table size (

size\_id smallint unsigned auto\_increment primary key,

size\_name varchar(100) not null

);

Таблиця "матеріал" призначена для зберігання інформації про матеріали, з яких виготовляються товари. SQL-скрипт для створення таблиці:

create table material (

material\_id smallint unsigned auto\_increment primary key,

material\_name varchar(100) not null

);

Таблиця "клієнт" містить дані про постійних клієнтів магазину. SQL-скрипт для створення таблиці:

create table customer (

customer\_id smallint unsigned auto\_increment primary key,

customer\_first\_name varchar(50) not null,

customer\_surname varchar(50) not null,

customer\_middle\_name varchar(50) not null,

customer\_phone\_number bigint not null,

customer\_email varchar(50),

discount\_card\_id bigint unsigned not null unique,

discount\_card\_accumulation float default 0

);

Таблиця "співробітник" використовується для зберігання інформації про працівників магазину. SQL-скрипт для створення таблиці:

create table employee (

employee\_id smallint unsigned auto\_increment primary key,

employee\_first\_name varchar(50) not null,

employee\_surname varchar(50) not null,

employee\_middle\_name varchar(50) not null,

employee\_position enum('менеджер', 'продавець-консультант', 'касир', 'складський працівник') not null,

employee\_phone\_number bigint not null,

employee\_email varchar(50) not null,

employee\_address varchar(255) not null,

employee\_gender enum('чоловік', 'жінка') not null,

employee\_date\_of\_birth date not null

);

Таблиця "постачальник" містить дані про постачальників товарів для магазину. SQL-скрипт для створення таблиці:

create table supplier (

supplier\_id smallint unsigned auto\_increment primary key,

supplier\_name varchar(255) not null,

supplier\_edrpou\_code bigint not null,

supplier\_fphone\_number bigint not null,

supplier\_sphone\_number bigint not null,

supplier\_address varchar(255) not null,

supplier\_current\_account bigint not null unique

);

Таблиця "продаж" використовується для запису інформації про продажі товарів. SQL-скрипт для створення таблиці:

create table sales (

sales\_id int unsigned auto\_increment primary key,

employee\_id smallint unsigned not null,

customer\_id smallint unsigned,

sale\_time datetime not null,

foreign key (employee\_id) references employee(employee\_id),

foreign key (customer\_id) references customer(customer\_id)

);

Таблиця "список\_куплених\_товарів" призначена для зберігання інформації про товари, які були продані у кожній окремій операції продажу. SQL-скрипт для створення таблиці:

create table sales\_list (

sales\_list\_id int unsigned auto\_increment primary key,

sales\_id int unsigned not null,

product\_quantity\_id smallint unsigned not null,

sales\_list\_price float not null,

sales\_list\_quantity smallint unsigned not null default 0,

foreign key (sales\_id) references sales(sales\_id),

foreign key (product\_quantity\_id) references product\_quantity(product\_quantity\_id)

);

Таблиця "повернення" містить дані про повернені товари. SQL-скрипт для створення таблиці:

create table product\_return (

product\_return\_id int unsigned auto\_increment primary key,

sales\_id int unsigned not null,

return\_reason varchar(255) not null,

foreign key (sales\_id) references sales(sales\_id)

);

Таблиця "список\_повернення" містить список товарів та їхні кількості кожного повернення:

create table product\_return\_list (

product\_return\_list\_id int unsigned auto\_increment primary key,

product\_return\_id int unsigned not null,

sales\_list\_id int unsigned not null,

product\_return\_list\_quantity smallint unsigned not null default 1,

foreign key (product\_return\_id) references product\_return(product\_return\_id),

foreign key (sales\_list\_id) references sales\_list(sales\_list\_id)

);

Таблиця "накладна" використовується для зберігання інформації про накладні на постачання товарів. SQL-скрипт для створення таблиці:

create table invoice (

invoice\_id int unsigned auto\_increment primary key,

employee\_id smallint unsigned not null,

supplier\_id smallint unsigned not null,

supply\_time datetime not null,

foreign key (employee\_id) references employee(employee\_id),

foreign key (supplier\_id) references supplier(supplier\_id)

);

Таблиця "список\_постачань" призначена для запису інформації про постачані товари у кожній окремій поставці. SQL-скрипт для створення таблиці:

create table supply\_list (

supply\_list\_id int unsigned auto\_increment primary key,

invoice\_id int unsigned not null,

product\_quantity\_id smallint unsigned not null,

supply\_list\_price float not null,

supply\_list\_quantity smallint unsigned not null default 1,

foreign key (invoice\_id) references invoice(invoice\_id),

foreign key (product\_quantity\_id) references product\_quantity(product\_quantity\_id)

);

Таблиця "користувач" призначена для зберігання користувачів:

create table db\_user (

user\_id tinyint unsigned auto\_increment primary key,

user\_login varchar(50) not null,

user\_name varchar(150) not null,

user\_pass varchar(50) not null,

user\_mode enum('admin', 'employee') not null

);

На рисунку 3.1 відображено схему взаємозв’язку таблиць між собою.

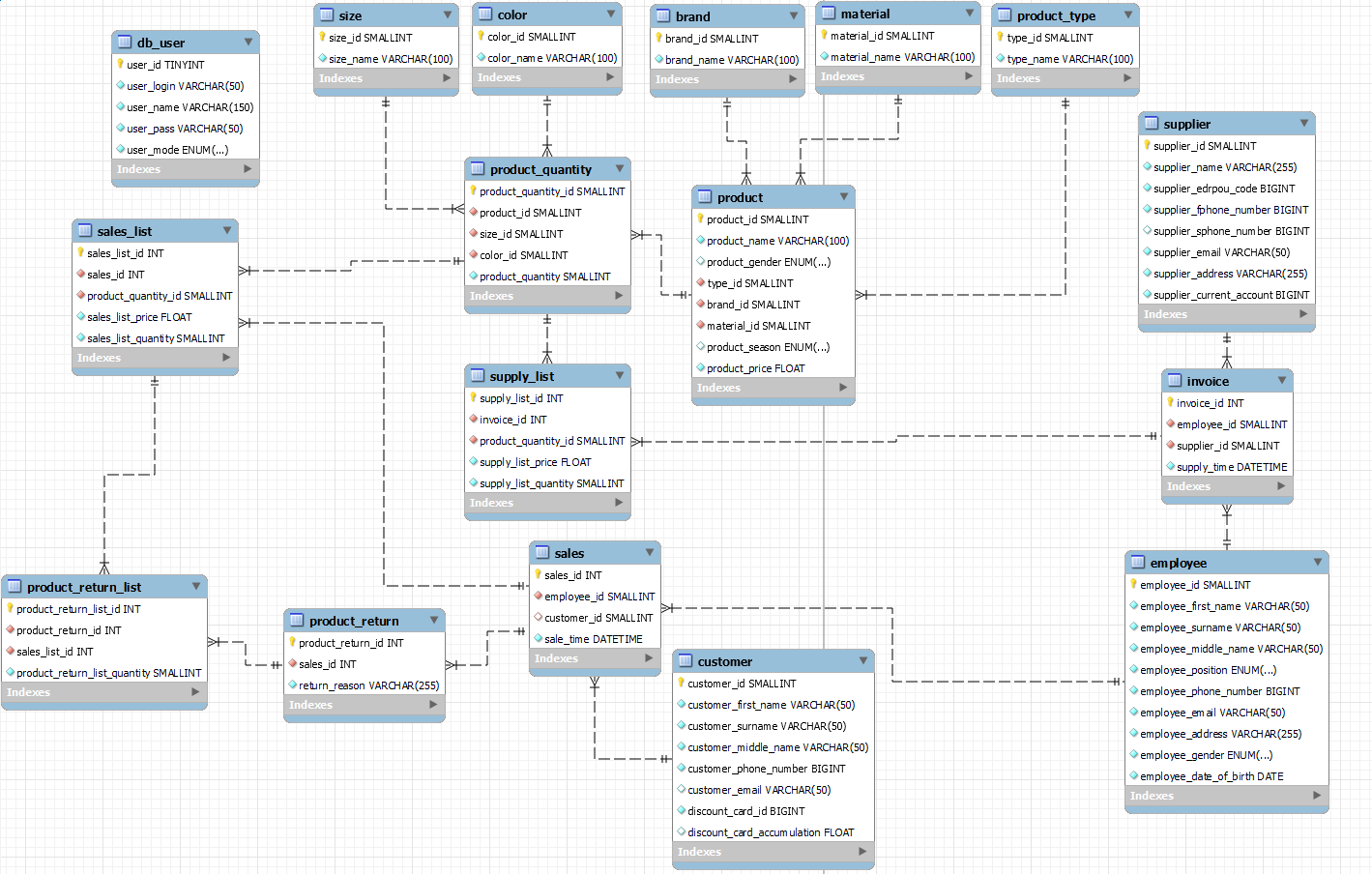


Рисунок 3.1 – Схема бази даних

## 3.2 Опис задач автоматизації та інтерфейсу користувача

Першим, що бачить користувач при вході у додаток є вікно авторизації. Користувач повинен ввести логін та пароль, щоб працювати з додатком далі. Вікно авторизації зображено на рисунку 3.2.

Зображення, що містить текст, знімок екрана, Шрифт, Мобільний телефон

Автоматично згенерований опис

Рисунок 3.2 – Вікно авторизації

При успішній авторизації відкриється головне вікно, функціїї якого залежать від рівня доступу користувача. Зліва у вікна знаходиться навігація, яка змінює вікно роботи з таблицею, відкриває вікно кошику та статистики. Таблиці головного вікна зображені на рисунках 3.3 та 3.4.

Зображення, що містить текст, знімок екрана, число

Автоматично згенерований опис

Рисунок 3.3 – Головне вікно адміністратора (Таблиця продукти)

Зображення, що містить текст, знімок екрана, число, програмне забезпечення

Автоматично згенерований опис

Рисунок 3.4 – Головне вікно працівника (Таблиця продукти)

Зображення, що містить текст, знімок екрана, число, програмне забезпечення

Автоматично згенерований опис

Рисунок 3.5 – Головне вікно з таблицею накладних

Зображення, що містить текст, знімок екрана, програмне забезпечення, число

Автоматично згенерований опис

Рисунок 3.6 – Головне вікно з таблицею клієнтів

Зображення, що містить текст, знімок екрана, число, Шрифт

Автоматично згенерований опис

Рисунок 3.7 – Головне вікно з таблицею продажів

Зображення, що містить текст, знімок екрана, число, програмне забезпечення

Автоматично згенерований опис

Рисунок 3.8 – Головне вікно з таблицею повернень

Зображення, що містить текст, знімок екрана, число, програмне забезпечення

Автоматично згенерований опис

Рисунок 3.9 – Головне вікно з таблицею співробітників

Зображення, що містить текст, знімок екрана, число, програмне забезпечення

Автоматично згенерований опис

Рисунок 3.10 – Головне вікно з таблицею постачальників

Кожне вікно складається з фільтру, який знаходиться зверху, та таблиці, яка знаходиться знизу. Цей фільтр дозволяє робити пошук та відфільтровувати дані в таблицях. Фільтр має дві кнопки: скинути (скидає параметри фільтру) та застосувати (застосовує фільтр та оновлює таблицю). Приклад використання фільтру зображено на рисунку 3.11.

Зображення, що містить текст, знімок екрана, число, схема

Автоматично згенерований опис

Рисунок 3.11 – Приклад використання фільтру

Під кожною таблицею знаходяться чотири основні кнопки:

* Оновити – оновлює таблицю
* Додати – відкриває вікно додання
* Змінити – відкриває вікно зміни
* Видалити – видаляє вибраний елемент з таблиці

Додання та зміна даних таблиці відбувається з використання одного вікна (кожна таблиця має своє вікно). При зміні даних, додаток сам заповнює поля. Приклад вікна додання(зміни) зображено на рисунку 3.12.

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Рисунок 3.12 – Вікно додання(зміни) для таблиці продуктів

Вікна з таблицями продуктів, накладних, продажів та повернень мають додаткову кнопку, для показу відповідних додаткових даних, яка називається "Показати дані". Щоб її активувати, треба спочатку вибрати елемент таблиці, по якому ви хочете побачити додаткову інформацію, та потім натискаєте кнопку. Приклад використання кнопки "Показати дані" зображено на рисунку 3.13.

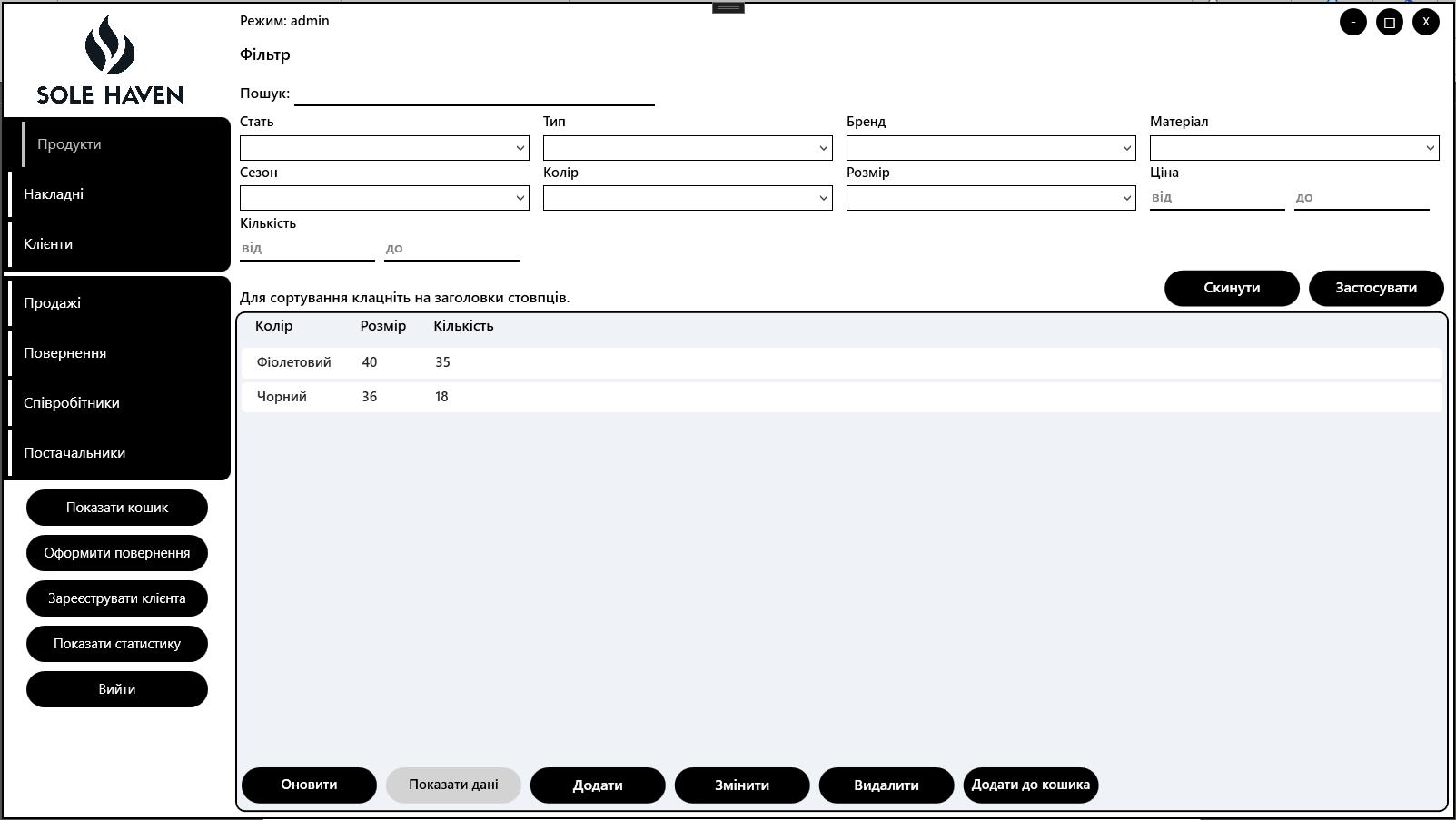


Рисунок 3.13 Перегляд додаткової інформації про продукт

Для оформлення покупки необхідно вибрати товар у таблиці продуктів та натиснути кнопку "Додати до кошика". Після цього виберіть кількість товару, і він буде доданий до кошика. Приклад додання товару до кошику показано на рисунках 3.14 та 3.15.

Зображення, що містить текст, знімок екрана, число, програмне забезпечення

Автоматично згенерований опис

Рисунок 3.14 – Вибір товару, який буде додано до кошику

Зображення, що містить знімок екрана, текст, число, схема

Автоматично згенерований опис

Рисунок 3.15 – Вибір кольору та розміру товару

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Рисунок 3.16 – Вікно запису кількості товару

Щоб переглянути кошик, натисніть кнопку "Переглянути кошик" на панелі навігації. Вікно кошика містить три кнопки для роботи з товарами:

* Оновити – оновлює вміст кошика.
* Змінити – дозволяє змінити кількість товару через відповідне поле.
* Видалити – видаляє товар з кошика.

Справа від кошика розташовані опції для вибору співробітника, який оформлює покупку, та постійного клієнта для застосування або накопичення знижки. Тут також відображається інформація щодо замовлення та кнопка "Оформити покупку", яка завершує процес покупки та формує чек. Вікно кошику та сформований чек зображене на рисунку 3.17 та 3.18.

Зображення, що містить знімок екрана, текст, програмне забезпечення

Автоматично згенерований опис

Рисунок 3.17 – Вікно кошику

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Рисунок 3.18 – Сформований чек

Додаток має вікно показу статистики, яке відкривається після натискання кнопки "Показати статистику". Вікно статистики має свій фільтр, який дозволяє вибрати кількість відображених стовпців в діаграмі. Вікно статистики зображено на рисунку 3.19.

Зображення, що містить текст, знімок екрана, число, Графік

Автоматично згенерований опис

Рисунок 3.19 – Вікно статистику

# ВИСНОВКИ

У процесі виконання курсової роботи на тему «Проектування та розробка бази даних «Магазин взуття»» було досягнуто кілька важливих результатів, які свідчать про успішне виконання поставлених завдань.

По-перше, проведено детальний аналіз вимог до системи. На основі цього аналізу визначено основні функціональні та нефункціональні вимоги, які дозволили сформувати чітке уявлення про необхідну структуру бази даних та її функціональні можливості.

По-друге, було спроектовано структуру бази даних, яка включає таблиці для зберігання інформації про товари, бренди, клієнтів, замовлення та співробітників. Створено схему даних, яка забезпечує ефективне зберігання та обробку інформації, а також забезпечує цілісність та взаємозв'язок між даними.

По-третє, розроблено основні елементи інтерфейсу користувача, що забезпечують зручний доступ до функцій системи. Це включає вікна логіну, додання або зміни об'єктів, а також вікно кошика, яке дозволяє користувачам легко управляти своїми покупками та оформляти замовлення.

По-четверте, особливу увагу приділено питанням безпеки даних та їх цілісності. Розроблено механізми контролю доступу та перевірки коректності введених даних, що забезпечує надійність та безпеку зберігання інформації в базі даних.

Насамкінець, реалізована база даних дозволяє ефективно управляти інформацією в магазині взуття, оптимізувати процеси обробки замовлень та підвищити якість обслуговування клієнтів. Впровадження цієї системи сприятиме покращенню роботи магазину, підвищенню задоволеності клієнтів та збільшенню обсягів продажів.

Отже, виконання даної курсової роботи показало, що правильне проектування та розробка бази даних є ключовими факторами успішного функціонування інформаційної системи у роздрібній торгівлі. Розроблена база даних повністю відповідає поставленим вимогам і готова до впровадження в роботу магазину взуття.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Microsoft Docs. (2023). Windows Presentation Foundation (WPF). Доступно за адресою:

<https://learn.microsoft.com/en-us/dotnet/desktop/wpf/>

1. Microsoft Docs. (2023). Документація Visual Studio. Доступно за адресою:

<https://learn.microsoft.com/en-us/visualstudio/>

1. MySQL Documentation. (2023). Довідковий посібник MySQL. Доступно за адресою:

<https://dev.mysql.com/doc/>

1. Документація по бібліотеці LiveCharts доступна за адресою: <https://v0.lvcharts.com/App/examples/Wpf/start>
2. Вільямс, Б., Лейн, Б., & Венц, Ч. (2018). Професійна Visual Studio 2017. Wrox.
3. Дайк, В., Веллінг, Л., & Томсон, Л. (2017). PHP і MySQL: Веб-розробка. Addison-Wesley Professional.
4. Скраггс, Д. (2015). MySQL для початківців: Довідник. Wiley.
5. Мюррей, М., Вейтер, Л., & Коллінз, Р. (2020). MySQL 8: Довідник для адміністраторів баз даних. Addison-Wesley Professional.
6. Зейбель, П. (2016). MySQL Cookbook: Рецепти для програмування та адміністрування. O'Reilly Media.
7. Форда, К. (2018). MySQL: Основи, Поглиблені концепції та Програмування. McGraw-Hill Education.
8. Ліберті, Дж. (2009). Програмування C# 4.0: Створення додатків для Windows, Web та RIA для .NET 4.0 Framework. O'Reilly Media.