

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Автоматизація управління збутом продукції для малого бізнесу
на основі SQL-запитів в СУБД Access»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти бакалавр
групи 126ІСТ_бд_2022
Кашаєв Єгор Олександрович
Керівник: Уткін Юрій Вікторович
Рецензент: Муравльов Володимир
В'ячеславович

Полтава – 2026 року

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

Освітня програма Інформаційні управляючі системи
Спеціальність 126 Інформаційні системи та технології
Рівень вищої освіти перший (бакалаврський)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Юрій УТКІН

«10» червня 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Кашасва Єгора Олександровича

1. Тема кваліфікаційної роботи:
«Автоматизація управління збутом продукції для малого бізнесу на основі SQL-запитів в СУБД Access»,
Керівник роботи: к. т. н., доцент, доцент кафедри інформаційних систем та технологій Уткін Юрій Вікторович
Затверджено наказом закладу вищої освіти від «10» квітня 2026 року № 383-ст
2. Строк подання здобувачем вищої освіти роботи «08» червня 2026 року.
3. Вихідні дані до роботи: статистичні, науково-популярні джерела мережі інтернет, дані офіційних сайтів компаній <https://www.microsoft.com/uk-ua/microsoft-365/enterprise/>, <https://newtelecom.ua/internet-v-ofis/>, середовища прикладних програм Draw.oi, MS Access, статистичні дані аналітичних компаній Work.ua
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Розділ 1 Основи проектування та застосування систем управління базами даних
Розділ 2. Етапи та зміст підготовчих робіт створення бази даних
Розділ 3. Розроблення і застосування системи бази даних на прикладі збутової діяльності підприємства
5. Перелік графічного матеріалу: схеми, рисунки, діаграми за темою та об'єктом дослідження

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання отримав
Оцінювання економічної ефективності результатів дослідження	Дивнич О. Д., к. е. н., доцент кафедри економіки та публічного управління	01.04.2026 р.	29.05.2026 р.

7. Дата видачі завдання «10» червня 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Вибір і затвердження теми роботи	02.06.2025 р. – 04.06.2025	
2	Складання і затвердження розгорнутого плану та завдання на кваліфікаційну роботу	05.06.2025 р. – 10.06.2025 р.	
3	Опрацювання джерел інформації	11.06.2025 р. – 15.06.2025 р.	
4	Збір, вивчення і обробка інформації, необхідної для виконання роботи	16.06.2025 р. – 20.06.2025 р.	
5	Виконання теоретико-методологічного розділу роботи	08.09.2025 р. – 01.11.2025 р.	
6	Виконання дослідницько-аналітичного розділу роботи	19.01.2026 р. – 14.02.2026 р.	
7	Виконання проєктно-рекомендаційного розділу роботи	16.02.2026 р. – 31.03.2026 р.	
8	Оцінювання економічної ефективності результатів дослідження	01.04.2026 р. – 29.05.2026 р.	
9	Оформлення тексту роботи	01.06.2026 р. – 06.06.2026р.	
10	Попередній захист роботи на кафедрі	08.06.2026 р.	
11	Доопрацювання роботи з урахуванням зауважень і пропозицій	09.06.2026 р. – 10.06.2026 р.	
12	Нормоконтроль	11.06.2026 р. – 15.06.2026 р.	
13	Захист кваліфікаційної роботи	16.06.2026 р. - 18.06.2026 р.	

Здобувач вищої освіти

Єгор КАШАЄВ

Керівник роботи

Юрій УТКІН

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 ОСНОВИ ПРОЄКТУВАННЯ ТА ЗАСТОСУВАННЯ СИСТЕМ УПРАВЛІННЯ БАЗАМИ ДАНИХ.....	8
1.1 Теоретичні основи організації та функціонування систем управління базами даних.....	8
1.2 Основні моделі представлення даних в системах баз даних.....	12
1.3 Роль мови SQL у створенні структурованих запитів баз даних.....	17
РОЗДІЛ 2 ЕТАПИ ТА ЗМІСТ ПІДГОТОВЧИХ РОБІТ СТВОРЕННЯ БАЗИ ДАНИХ.....	22
2.1 Аналіз предметної області та концепції створення бази даних в системі збутової діяльності підприємства.....	22
2.2 Обґрунтування принципів і моделі створення бази даних.....	26
2.3 Проєктування фізичної структури та зв'язків реляційної бази даних в середовищі СУБД MS Access.....	31
РОЗДІЛ 3 РОЗРОБЛЕННЯ І ЗАСТОСУВАННЯ СИСТЕМИ БАЗИ ДАНИХ НА ПРИКЛАДІ ЗБУТОВОЇ ДІЯЛЬНОСТІ ПІДПРИЄМСТВА.....	37
3.1 Функції та використання запитів у системі управління базою даних.....	37
3.2 Конструювання форм та звітів для управління БД.....	41
3.3 Економічна ефективність використання СУБД у збутовій діяльності підприємства.....	46
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	52
ДОДАТКИ.....	56

ВСТУП

Актуальність теми кваліфікаційної роботи полягає у зростаючій потребі малого бізнесу в ефективному управлінні процесами збуту продукції в умовах високої конкуренції та цифровізації економіки. Значна частина малих підприємств і досі використовує традиційні або частково автоматизовані методи обліку продажів, що призводить до втрати часу, виникнення помилок при обробці даних та ускладнює оперативне прийняття управлінських рішень.

Використання інформаційних систем для автоматизації управління збутом дозволяє оптимізувати процеси обліку товарів, контролю замовлень, аналізу продажів та формування звітності. Особливого значення набуває застосування систем управління базами даних, зокрема Microsoft Access, яка є доступним і функціональним інструментом для створення інформаційних систем у сфері малого бізнесу. Використання SQL-запитів у СУБД Access забезпечує швидке опрацювання даних, автоматизацію типових операцій та підвищення ефективності роботи персоналу.

Метою кваліфікаційної роботи є дослідження теоретичних аспектів роботи з базами даних та обґрунтування доцільності, функціональності та ефективності розроблення і використання баз даних в процесі здійснення збутової діяльності.

Завданнями кваліфікаційної роботи є: розробка автоматизованої системи управління збутом продукції для малого бізнесу на основі SQL-запитів у СУБД Access з метою підвищення ефективності обробки даних, оптимізації процесів обліку продажів та підтримки прийняття управлінських рішень.

Об'єктом дослідження є процес автоматизації управління збутом продукції на підприємствах малого бізнесу.

Предметом дослідження є методи та засоби автоматизації управління збутом продукції із використанням SQL-запитів та можливостей СУБД Access.

Методами дослідження обрано:

- проєктний для створення інформаційної моделі системи управління збутом та реалізації SQL-запитів;

- метод аналізу та узагальнення наукових джерел для вивчення принципів автоматизації бізнес-процесів, особливостей управління збутом продукції та можливостей СУБД Access;

- метод системного аналізу для дослідження структури процесу збуту продукції, визначення основних елементів інформаційної системи та взаємозв'язків між ними;

- моделювання для побудови структури бази даних, розробки логіки взаємодії таблиць, форм, запитів і звітів у СУБД Access;

- тестування для перевірки працездатності створеної системи, правильності виконання SQL-запитів та коректності обробки даних;

- порівняльний для оцінки ефективності автоматизованого способу управління збутом у порівнянні з традиційними методами ведення обліку.

Інформаційна база кваліфікаційної роботи зібрана на основі наукових статей, підручників та навчальних посібників, стандартів, офіційних сайтів розробників прикладного програмного забезпечення, а також даних, отриманих під час виробничих та переддипломної практик.

Практична значущість роботи полягає у розробці автоматизованої системи управління збутом продукції для малого бізнесу на основі СУБД Access, яка може бути використана для оптимізації процесів обліку продажів, контролю товарів і формування звітності. Впровадження розробленої системи сприятиме підвищенню ефективності діяльності малого підприємства, скороченню часу на ведення обліку та покращенню контролю за процесами збуту продукції. Окремі результати роботи також можуть бути використані як навчальний приклад для вивчення основ проєктування баз даних і роботи з SQL-запитами у СУБД Access.

Апробація роботи відбулася шляхом публікації тез на студентській науково-практичній конференції.

Структура і обсяг кваліфікаційної роботи: пояснювальна записка викладена на 51 сторінці й складається зі змісту, вступу, трьох розділів, списку використаних джерел та додатків. Робота включає 29 рисунків.

РОЗДІЛ 1

ОСНОВИ ПРОЄКТУВАННЯ ТА ЗАСТОСУВАННЯ СИСТЕМ УПРАВЛІННЯ БАЗАМИ ДАНИХ

1.1 Теоретичні основи організації та функціонування систем управління базами даних

Діяльність суб'єктів господарювання в сучасних умовах здійснюється під впливом численних факторів, які прямо або опосередковано впливають на функціонування підприємств і прийняття управлінських рішень щодо коригування діяльності об'єкта управління. При цьому на успішність подальшої діяльності підприємства, стабільність його розвитку та ефективність функціонування значною мірою впливають швидкість обробки інформації, її достовірність і раціональна організація інформаційних потоків.

В умовах цифровізації економіки використання окремих елементів інформаційних технологій (ІТ) для швидкої обробки інформації стає недостатнім. Це обумовлює необхідність створення та впровадження сучасних інформаційних систем (ІС), здатних забезпечити комплексне розв'язання управлінських задач. Реалізація таких систем ґрунтується на застосуванні інноваційних методів управління, економіко-математичних моделей, програмно-технічних засобів та сучасних ІТ, що забезпечують автоматизацію виконання управлінських функцій і процедур.

Термін «система» (від дав.-гр. *σύντημα* – поєднання, сполучення) означає сукупність взаємопов'язаних елементів, які утворюють певну цілісність та функціонують для досягнення визначеної мети. Будь-яка система характеризується наявністю структури, зв'язків між елементами та механізмів взаємодії із зовнішнім середовищем. У сфері управління підприємством система повинна забезпечувати збір, обробку, зберігання та передачу інформації, необхідної для прийняття рішень [1]. Оскільки прийняття управлінських рішень відбувається на основі обробленої інформації у певній предметній області, для

вирішення комплексу управлінських завдань необхідним є використання інформаційної системи (ІС).

У відповідності до визначення, наведеного у Державному Стандарті України, «...інформаційна система – це система, яка організовує накопичення і маніпулювання інформацією щодо конкретної проблемної сфери» [2].

Основу будь-якої інформаційної системи складають дані. Організація масивів даних на носіях інформації може здійснюватися двома основними способами: у вигляді системи файлів або шляхом створення баз даних. При файловій організації інформація зберігається у вигляді окремих файлів, орієнтованих переважно на розв'язання конкретної прикладної задачі. Такий підхід був характерним для ранніх етапів розвитку інформаційних систем і дозволяв забезпечити високу швидкість виконання окремих операцій. Проте файлова організація даних має низку суттєвих недоліків.

Насамперед, вузька спеціалізація файлів призводить до дублювання інформації, оскільки однакові елементи даних можуть зберігатися у різних файлах і використовуватися різними програмами. Це збільшує надмірність даних та ускладнює підтримку їх актуальності. Крім того, для виконання однотипних операцій із різними файлами використовуються окремі програмні засоби, що ускладнює контроль правильності даних і підвищує ймовірність виникнення помилок та суперечностей.

Ще одним недоліком файлової організації є обмежені можливості формування складних інформаційних запитів, які охоплюють декілька предметних областей одночасно. У багатьох випадках навіть наявність необхідних даних у різних файлах не дозволяє швидко отримати потрібну інформацію через відсутність єдиної структури їх зберігання та обробки. Такі проблеми особливо помітні в діяльності підприємств, де необхідно постійно аналізувати значні обсяги інформації щодо товарів, клієнтів, замовлень, фінансових операцій та інших бізнес-процесів.

Саме через недоліки файлового підходу виникла необхідність відокремлення даних від прикладних програм та створення універсального

механізму їх зберігання й обробки. Це сприяло появі баз даних і систем управління базами даних (СУБД), призначених для забезпечення ефективного введення, зберігання, пошуку, оновлення та обробки інформації.

За популярним визначенням, база даних це «...впорядкований набір логічно взаємопов'язаних даних, що використовуються спільно, та призначені для задоволення інформаційних потреб користувачів» [3].

Основне призначення баз даних – забезпечити надійне зберігання значних обсягів інформації та забезпечити швидкий доступ до неї для користувачів та програм. Таким чином, база даних складається з двох основних компонентів: набору даних, що зберігаються, та системи управління ними. Саме система управління базами даних забезпечує виконання всіх операцій з даними та підтримує взаємодію користувача з інформаційною системою [3].

До основних вимог, яким повинні відповідати сучасні бази даних, належать інтегрованість даних, мінімізація дублювання інформації, підтримка цілісності та актуальності даних, можливість багатокористувацької роботи, захист інформації від несанкціонованого доступу та забезпечення високої продуктивності обробки запитів. Важливими характеристиками також є масштабованість, можливість розширення структури бази даних та зручність використання для кінцевих користувачів [4].

Особливе значення має забезпечення цілісності даних, тобто підтримка правильності та узгодженості інформації у процесі її обробки. Для цього в СУБД використовуються спеціальні механізми контролю обмежень, ключових полів, зв'язків між таблицями та перевірки коректності введених даних. Крім того, сучасні СУБД підтримують механізми резервного копіювання та відновлення інформації у разі виникнення технічних збоїв.

За способом організації даних виділяють ієрархічні, мережеві, реляційні та об'єктно-орієнтовані бази даних. Найбільш поширеними на сьогодні є реляційні бази даних, у яких інформація подається у вигляді взаємопов'язаних таблиць.

Така модель забезпечує простоту структури, зручність обробки даних та можливість формування складних інформаційних запитів.

Основою функціонування реляційних баз даних є використання мови структурованих запитів SQL (Structured Query Language). SQL дозволяє створювати таблиці, додавати, змінювати та видаляти дані, а також здійснювати пошук і аналіз інформації. Використання SQL-запитів значно підвищує ефективність роботи інформаційних систем та забезпечує автоматизацію багатьох управлінських процесів.

Проектування баз даних є важливим етапом створення інформаційної системи. На цьому етапі визначаються структура таблиць, типи даних, ключові поля, зв'язки між таблицями та правила обробки інформації. Якісне проектування бази даних забезпечує високу продуктивність системи, зменшує надмірність даних і підвищує ефективність виконання запитів.

Для малого бізнесу системи управління базами даних мають особливе практичне значення. Їх використання дозволяє автоматизувати облік товарів, клієнтів, продажів та фінансових операцій, підвищити швидкість обробки інформації та покращити якість управлінських рішень. Автоматизація процесів управління збутом продукції сприяє зменшенню кількості помилок, скороченню часу на виконання рутинних операцій і забезпечує більш ефективний контроль діяльності підприємства.

Однією з найбільш доступних систем управління базами даних для малого бізнесу є Microsoft Access [5]. Дана СУБД поєднує засоби створення таблиць, форм, звітів, SQL-запитів та елементів автоматизації у зручному графічному середовищі. Завдяки відносній простоті використання та достатньому рівню функціональності Microsoft Access широко застосовується для створення інформаційних систем управління невеликими підприємствами, зокрема у сфері обліку та збуту продукції.

1.2 Основні моделі представлення даних в системах баз даних

Метою функціонування будь-якої ІС є збирання, зберігання, обробка та аналіз даних про об'єкти реального світу з урахуванням взаємозв'язків між ними. У теорії БД для опису таких об'єктів та їх характеристик використовуються поняття «сутність» і «атрибут». Саме ці елементи є основою побудови структури будь-якої бази даних та забезпечують формалізований опис предметної області.

Під сутністю розуміють окремий об'єкт або явище реального світу, інформація про яке повинна зберігатися та оброблятися в інформаційній системі. Сутність характеризується наявністю унікальних властивостей, що дозволяють відрізнити один об'єкт від іншого. Сутностями можуть бути люди, товари, підприємства, замовлення, працівники, транспортні засоби, документи та інші об'єкти предметної області [6].

Сукупність однотипних сутностей утворює клас або тип сутностей. Наприклад, класами можуть бути «Клієнти», «Товари», «Постачальники», «Замовлення» тощо. ІС працює саме з наборами таких сутностей, використовуючи конкретні значення їх характеристик для виконання обробки даних та формування інформаційних запитів.

Для опису властивостей сутностей використовуються атрибути. Атрибут – це характеристика або властивість об'єкта, яка набуває певного значення для конкретного екземпляра сутності. Атрибути можуть мати різні типи даних: числові, текстові, логічні, дата/час та інші. Наприклад, для сутності «Товар» атрибутами можуть бути код, назва, ціна товару, кількість на складі, виробник та категорія [6]. Кожен атрибут має множину допустимих значень, яка називається доменом. Домен визначає тип та межі можливих значень атрибута. Наприклад, для атрибута «Дата продажу» доменом будуть календарні дати, а для атрибута «Кількість товару» – додатні числові значення.

Важливим елементом організації баз даних є встановлення зв'язків між сутностями. У реальних інформаційних системах об'єкти предметної області рідко існують ізольовано, тому між ними формуються певні логічні

взаємозв'язки. Наявність таких зв'язків дозволяє формувати складні інформаційні запити та забезпечує цілісність структури бази даних.

Логічний спосіб організації даних та зв'язків між ними визначається моделлю представлення даних. Модель даних описує структуру інформації, правила її організації, способи взаємодії між елементами та операції, які можуть виконуватися над даними. Від типу моделі залежить ефективність роботи БД, швидкість обробки інформації та зручність роботи користувачів із системою.

На сучасному етапі розвитку ІТ використовується декілька основних моделей представлення даних, серед яких найбільш поширеними є ієрархічна, мережева, реляційна та об'єктно-орієнтована моделі. Крім того, все частіше застосовуються змішані та нереляційні підходи до організації даних, що дозволяє забезпечити ефективну обробку великих обсягів інформації.

Однією з перших моделей представлення даних стала ієрархічна модель, яка базується на деревоподібній структурі, коли між елементами даних встановлюються відношення типу «батько – нащадок». У такій структурі кожен дочірній елемент може мати лише одного предка, тоді як один батьківський елемент може мати декілька дочірніх. На її основі було розроблено перші СУБД [7]. Приклад схематичної побудови ієрархічної структури наведено на рис. 1.1.



Рисунок 1.1 – Схема ієрархічної моделі представлення даних

Подальшим етапом розвитку стала мережева модель даних. Вона була створена для усунення недоліків ієрархічної моделі та забезпечення більш

гнучкого представлення інформації. У мережевій моделі кожен елемент даних може бути пов'язаний одночасно з декількома іншими елементами. Таким чином реалізуються зв'язки типу «багато-до-багатьох», наприклад, рис. 1.2.

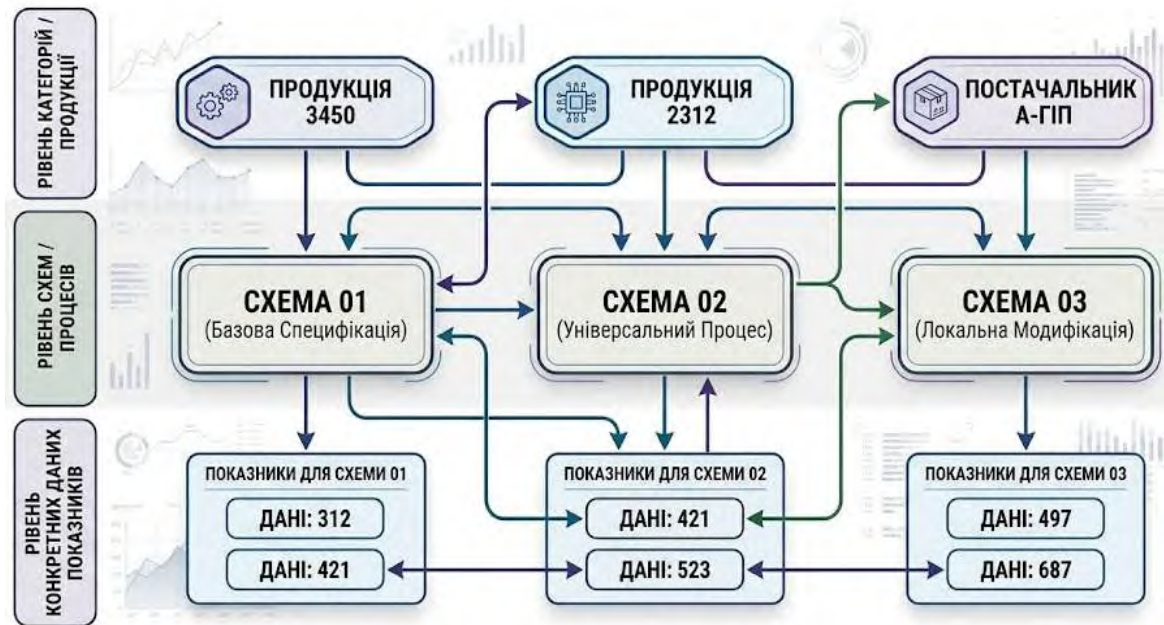


Рисунок 1.2 – Приклад мережевої моделі даних (зв'язки «багато-до-багатьох»)

Мережева модель дозволяє ефективно організовувати складні інформаційні структури та забезпечує швидкий доступ до даних. Вона характеризується високою продуктивністю при виконанні складних операцій обробки інформації. Однак складність побудови та адміністрування мережесхемних баз даних значно ускладнює їх використання. Крім того, для роботи з такими системами необхідна висока кваліфікація розробників і адміністраторів БД.

Найбільшого поширення у сучасних інформаційних системах набула реляційна модель даних. Вона була запропонована американським ученим Едгаром Коддом у 1970 р. та стала основою більшості сучасних систем управління БД. Реляційна модель базується на поданні інформації у вигляді таблиць (відношень), які складаються з рядків і стовпців [8].

У реляційній моделі кожна таблиця описує певний об'єкт предметної області, а її рядки містять окремі записи про цей об'єкт. Стовпці (поля) таблиці визначають характеристики або атрибути об'єкта. Ключові поля

використовуються для встановлення зв'язків між таблицями (рис. 1.3) [9]. Основними типами ключів є первинний ключ, який однозначно ідентифікує запис, та зовнішній ключ, який і використовується для встановлення зв'язків.

	Поле_1	Поле_2	...	Поле_n
	Назва поля 1	Назва поля 2	...	Назва поля_n
запис_1	атрибут11	атрибут12	...	атрибут1n
запис_2	атрибут11	атрибут22	...	атрибут2n
...	...	...	...	...
запис_k	атрибутk1	атрибутk2	...	атрибутkn

Рисунок 1.3 – Основні елементи відношення (таблиці)

Важливою характеристикою реляційної моделі є забезпечення цілісності даних. Для цього використовуються механізми обмежень, ключові поля та правила підтримки й оновлення зв'язків між таблицями. Це дозволяє уникнути дублювання інформації та забезпечує узгодженість даних у системі.

Основною перевагою реляційної моделі є її простота та універсальність. Таблична форма представлення даних зрозуміла користувачам і забезпечує зручність в обробці інформації. Крім того, реляційна модель підтримує використання SQL, що дозволяє виконувати складні запити до бази даних, шукати, сортувати, фільтрувати та аналізувати інформацію.

До переваг реляційної моделі належать незалежність даних від прикладних програм, можливість масштабування БД, підтримка багатокористувацької роботи та відносна простота адміністрування. Саме тому реляційні СУБД широко використовуються у фінансових системах, системах управління підприємством, електронній комерції, банківській справі та інших галузях.

Однак, реляційна модель має і певні недоліки. При роботі з дуже великими обсягами даних або складними мультимедійними об'єктами продуктивність системи знижується. Реляційна модель не завжди ефективно підтримує зберігання складних об'єктів із великою кількістю вкладених структур.

Для вирішення таких проблем була створена об'єктно-орієнтована модель даних. Вона базується на принципах об'єктно-орієнтованого програмування та передбачає представлення інформації у вигляді об'єктів. Об'єкти містять не лише дані, але й методи їх обробки [10]. Основними перевагами об'єктно-орієнтованої моделі є можливість роботи зі складними структурами даних, підтримка наслідування, інкапсуляції та поліморфізму. Це дозволяє ефективно реалізовувати системи, пов'язані з мультимедійною інформацією, графікою, інженерними розрахунками та іншими складними предметними областями.

Проте об'єктно-орієнтовані бази даних мають обмежене поширення через складність реалізації та недостатню стандартизацію. Компромісом між реляційною та об'єктно-орієнтованою моделями стала об'єктно-реляційна модель даних [11]. Вона поєднує переваги реляційного підходу з можливістю роботи зі складними об'єктами. Загальний принцип організації даних об'єктно-орієнтованої моделі наведено на рис. 1.4.

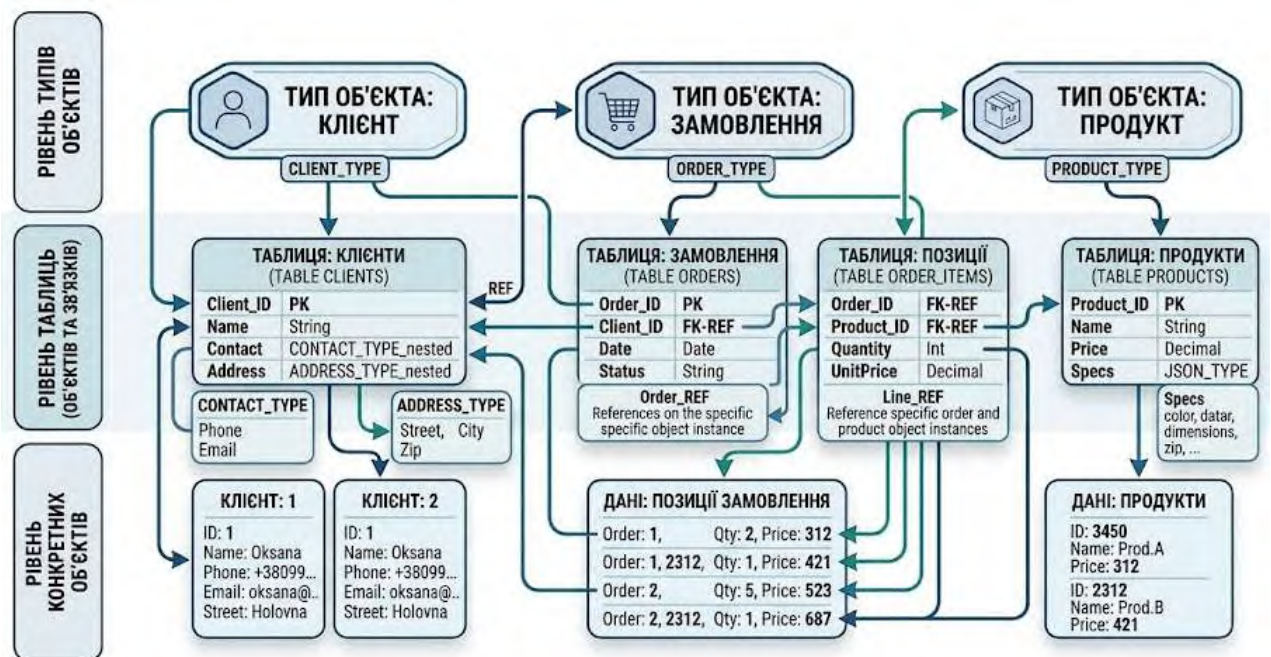


Рисунок 1.4 – Приклад організації об'єктно-реляційної моделі даних

У сучасних умовах розвитку інформаційних технологій також активно використовуються нереляційні моделі даних, відомі під загальною назвою NoSQL. Такі системи були створені для обробки великих обсягів слабо

структурованої інформації та забезпечення високої продуктивності у вебзастосунках і хмарних сервісах [12].

До основних типів NoSQL-систем належать документно-орієнтовані бази даних, ключ-значення системи, колонкові бази даних та графові бази даних. Документно-орієнтовані системи зберігають інформацію у вигляді документів, зазвичай у форматах JSON або XML. Бази типу «ключ-значення» забезпечують надзвичайно швидкий доступ до даних за унікальним ключем. Графові бази даних орієнтовані на ефективне представлення складних зв'язків між об'єктами.

Попри активний розвиток NoSQL-технологій, реляційна модель продовжує залишатися основою більшості інформаційних систем малого та середнього бізнесу. Це пояснюється її надійністю, простотою використання, підтримкою стандартної мови SQL та широкими можливостями для автоматизації бізнес-процесів.

У межах даної роботи особливий інтерес становить саме реляційна модель даних, оскільки система автоматизації управління збутом продукції реалізовується на основі СУБД Microsoft Access. Дана система підтримує реляційний підхід до організації інформації та забезпечує ефективну роботу з таблицями, запитам, формами і звітами. Використання реляційної моделі дозволяє забезпечити цілісність даних, мінімізувати дублювання інформації та створити ефективну систему управління збутом продукції для малого бізнесу..

1.3 Роль мови SQL у створенні структурованих запитів баз даних

Ефективне функціонування сучасних систем управління базами даних неможливе без використання спеціалізованих засобів роботи з інформацією. Ефективне функціонування сучасних систем управління базами даних значною мірою забезпечується використанням спеціалізованих мов обробки інформації. Основною мовою взаємодії користувача та програмних засобів із реляційними базами даних є SQL (Structured Query Language – мова структурованих запитів).

Вона використовується для створення структури БД, введення та редагування інформації, організації доступу до даних, формування запитів і забезпечення функціонування інформаційних систем різного призначення [13].

Мова SQL була розроблена у 1970-х роках компанією IBM на основі реляційної моделі даних Едгара Кодда. Головною метою стало створення універсального інструменту для роботи з реляційними БД незалежно від конкретної системи управління БД. На сьогодні SQL є міжнародним стандартом і підтримується більшістю сучасних СУБД, зокрема Microsoft Access, MySQL, PostgreSQL, Oracle Database та Microsoft SQL Server [13].

Мову SQL традиційно відносять до декларативних або описових мов програмування. Це означає, що користувач описує, яку інформацію необхідно отримати або яку операцію потрібно виконати, але не визначає безпосередній алгоритм реалізації цих дій. Основна увага в SQL приділяється результату виконання запиту, а не способу його досягнення. Сучасні реалізації SQL підтримують окремі елементи процедурного програмування, зокрема оператори умовного переходу, циклів, змінних та процедур, тому SQL поєднує можливості описового та процедурного підходів до обробки даних.

Функціональні можливості SQL умовно поділяються на декілька основних компонентів: DDL, DML, DCL та TCL. Компонент DDL (Data Definition Language – мова визначення даних) використовується для створення та зміни структури бази даних. За допомогою команд цієї групи можна створювати таблиці, змінювати їх структуру, визначати типи даних, ключові поля та видаляти об'єкти бази даних. Основними командами DDL є CREATE, ALTER та DROP. Саме ці оператори забезпечують формування логічної структури бази даних відповідно до вимог предметної області [14].

Компонент DCL (Data Control Language – мова керування даними) забезпечує організацію захисту інформації та контроль доступу користувачів до бази даних. За допомогою команд цієї групи визначаються права користувачів на перегляд, зміну або видалення інформації. Основними командами DCL є GRANT та REVOKE.

Компонент TCL (Transaction Control Language – мова управління транзакціями) використовується для керування транзакціями. Транзакція являє собою логічно завершену послідовність операцій із базою даних, яка повинна бути виконана повністю або не виконана зовсім. Це дозволяє забезпечити цілісність та узгодженість інформації. Основними командами TCL є COMMIT, ROLLBACK та SAVEPOINT.

Усі ключові слова SQL є зарезервованими, тому їх не рекомендується використовувати як імена таблиць, полів або змінних. SQL-запити можуть записуватися як великими, так і малими літерами, а також розміщуватися в одному або декількох рядках. Завершення SQL-виразу зазвичай позначається крапкою з комою (;), хоча в окремих системах при введенні одного запиту цей символ може бути необов'язковим.

Важливою складовою SQL є система типів даних. Вона визначає формат і спосіб зберігання інформації в таблицях бази даних. Для текстових даних SQL підтримує два основних типи: CHARACTER та CHARACTER VARYING.

Тип CHARACTER або CHAR використовується для зберігання рядків фіксованої довжини. При визначенні такого типу задається максимальна кількість символів у рядку. Якщо введенне значення містить менше символів, ніж передбачено форматом, решта позицій автоматично заповнюється пробілами.

Тип CHARACTER VARYING або VARCHAR застосовується для зберігання текстових даних змінної довжини. У цьому випадку зберігається лише фактична кількість введених символів, що дозволяє ефективніше використовувати пам'ять. Саме тип VARCHAR найбільш часто використовується при створенні таблиць баз даних, оскільки значення текстових полів у більшості випадків мають різну довжину [15].

Крім стандартних символьних типів, у багатьох СУБД використовуються спеціальні типи для зберігання великих текстових даних, наприклад TEXT або MEMO. Такі типи застосовуються для роботи з описами, коментарями, примітками та іншою текстовою інформацією значного обсягу.

Основним засобом роботи з даними в SQL є SQL-запит. У результаті виконання SQL-запиту формується результатна таблиця, яка містить дані, що відповідають умовам запиту. Якщо потрібних записів не знайдено, створюється порожня таблиця результатів. Результатна таблиця існує лише під час виконання запиту та не зберігається фізично у базі даних, тому її часто називають віртуальною таблицею.

Базовий SQL-запит для вибірки даних має вигляд:

SELECT списокСтовпців FROM списокТаблиць.

Оператори *SELECT* та *FROM* є обов'язковими елементами запиту. Оператор *SELECT* визначає перелік стовпців, які необхідно отримати у результаті виконання запиту, а оператор *FROM* вказує таблиці, з яких здійснюється вибірка даних.

Для уточнення умов вибірки використовуються додаткові оператори SQL. Найбільш поширеними серед них: *WHERE*, *GROUP BY*, *HAVING* і *ORDER BY*.

Оператор *WHERE* використовується для відбору записів за певними умовами та виконує функцію фільтрації даних. Оператор *GROUP BY* призначений для групування записів за значеннями окремих полів. Це дозволяє виконувати агрегатні обчислення, наприклад визначати суму, середнє значення або кількість записів у групі.

Оператор *HAVING* застосовується для відбору груп записів після виконання групування. Він використовується лише разом із *GROUP BY*. Оператор *ORDER BY* забезпечує сортування результатів запиту за одним або декількома полями у зростаючому або спадному порядку.

SQL-запит із використанням усіх основних операторів уточнення має вигляд:

SELECT списокСтовпців FROM Ім'яТаблиці;

WHERE УмоваПошуку;

GROUP BY полеГрупування;

HAVING УмоваПошуку;

ORDER BY УмоваСортування.

Хоча оператори у SQL-запиті записуються у визначеному порядку, виконуються вони системою управління базами даних в іншій послідовності. Спочатку обробляється оператор FROM, потім WHERE, GROUP BY, HAVING, SELECT і лише після цього ORDER BY. Такий механізм виконання запитів забезпечує ефективну обробку інформації та формування результатної таблиці.

У системі Microsoft Access SQL використовується як основний механізм формування запитів до бази даних. Хоча Access підтримує графічний конструктор запитів, усі операції фактично реалізуються за допомогою SQL-команд. Це дозволяє користувачам поєднувати простоту візуального проектування із широкими можливостями мови структурованих запитів.

Для автоматизації управління збутом продукції SQL має особливе значення, оскільки дозволяє швидко виконувати пошук товарів, аналіз продажів, формування звітів, визначення залишків продукції та контроль діяльності підприємства. Використання SQL-запитів забезпечує підвищення швидкості обробки інформації, зменшення кількості помилок та покращення якості прийняття управлінських рішень [16].

Таким чином, SQL є універсальним та ефективним інструментом роботи з реляційними базами даних. Її використання забезпечує створення сучасних інформаційних систем, автоматизацію обробки даних та підтримку управлінських процесів у діяльності підприємств різних сфер.

РОЗДІЛ 2

ЕТАПИ ТА ЗМІСТ ПІДГОТОВЧИХ РОБІТ СТВОРЕННЯ БАЗИ ДАНИХ

2.1 Аналіз предметної області та концепції створення бази даних в системі збутової діяльності підприємства

У сучасних умовах функціонування підприємств особливого значення набуває ефективна організація процесів збуту продукції. Саме збутова діяльність забезпечує реалізацію готової продукції, формування прибутку підприємства та підтримку стабільних взаємовідносин із клієнтами. Для підприємств малого бізнесу питання автоматизації збутової діяльності є особливо актуальним, оскільки обмежені фінансові та кадрові ресурси потребують оптимізації бізнес-процесів і підвищення ефективності управління інформацією.

Виконання робіт зі створення інформаційного продукту передбачає проведення аналізу предметної області, вивчення інформаційних потоків, потреб користувачів системою, алгоритмів розв'язання задач, програмування процедур обробки даних. База даних із системою запитів, що розробляється, призначена для управління процесами збуту продукції, яка надходить і обліковується на складах підприємства. До основних складових збутової діяльності належать облік товарів, реєстрація клієнтів, оформлення замовлень, контроль реалізації продукції, ведення інформації про постачання, аналіз продажів та формування звітності. Всі зазначені процеси супроводжуються постійним накопиченням та обробкою значних обсягів інформації, що зумовлює необхідність використання сучасних інформаційних систем.

Всебічному дослідженню економічної сутності збутової діяльності присвячено значну кількість наукових праць як українських, так і зарубіжних учених. У сучасних умовах господарювання збут розглядається як одна з ключових складових діяльності підприємства, оскільки саме він забезпечує доведення продукції до кінцевого споживача та формування фінансових результатів діяльності підприємства [17]. Згідно з одним із поширених

визначень, збут – це «сфера діяльності підприємства, спрямована на реалізацію продукції на певних ринках» [17].

Таке трактування підкреслює практичну спрямованість збутової діяльності та її безпосередній зв'язок із процесами реалізації товарів і взаємодії з покупцями. У працях зарубіжних учених поняття збуту розглядається у двох основних аспектах [18]:

– у широкому розумінні збут трактується як процес руху товару від виробника до споживача, який починається після завершення виробництва продукції та завершується її передачею покупцю;

– у вузькому розумінні збут розглядається як завершальний етап передачі права власності на товар від продавця до кінцевого споживача або посередника.

Схематично мета, цілі, стратегія збутової діяльності показані на рис. 2.1.



Рисунок 2.1 – Концептуальна схема управління збутом продукції підприємства малого бізнесу (дизайн на основі [19])

Для цілей автоматизації управління збутом продукції важливим є комплексний підхід, який поєднує облік товарних операцій, інформаційний супровід продажів та контроль виконання операцій реалізації продукції. У структурі збутової діяльності підприємства можна виділити декілька основних елементів, які формують предметну область створюваної ІС.

1. Товар або продукція підприємства. Саме товар виступає основним об'єктом обліку в системі збуту. Для кожної одиниці продукції необхідно зберігати інформацію про назву товару, код, категорію, ціну, кількість на складі, виробника та інші характеристики.

2. Клієнти або покупці, контрагенти. Система управління збутом повинна забезпечувати облік інформації про фізичних чи юридичних осіб, які здійснюють придбання продукції. Дані про клієнтів використовуються для формування історії продажів, аналізу попиту та підтримки взаємовідносин із покупцями.

3. Замовлення або операції продажу, які відображають факт реалізації продукції та містять інформацію про дату продажу, перелік товарів, кількість продукції, суму замовлення, спосіб оплати та відповідального працівника.

4. Складський облік та контроль товарних залишків. У процесі реалізації продукції система повинна забезпечувати автоматичне оновлення інформації про кількість товарів на складі для уникнення дефіциту або надлишку продукції.

5. Фінансові розрахунки з клієнтами, які включають контроль оплати, облік заборгованостей, фіксацію способів оплати та формування фінансової звітності щодо реалізації продукції.

Важливим складником збутової діяльності є формування аналітичної інформації та звітності. Аналіз обсягів продажів, визначення найбільш популярних товарів, оцінювання активності клієнтів та контроль ефективності реалізації продукції є необхідними умовами прийняття управлінських рішень.

Усі зазначені елементи збутової діяльності тісно пов'язані між собою та потребують централізованого зберігання й обробки інформації. Реалізація такої системи у середовищі Microsoft Access дозволяє забезпечити структуроване зберігання даних, автоматизацію облікових операцій, швидкий пошук

інформації та формування необхідної звітності для підтримки діяльності малого бізнесу. Специфічною особливістю Microsoft Access є те, що вся інформація бази даних зберігається в одному файлі, розширення якого *.mdb. Основними об'єктами СКБД Access є наступні [20].

1. Таблиця є головним об'єктом БД і використовуються для безпосереднього зберігання інформації про об'єкти предметної області. Передбачена можливість розподілу доступу до всієї БД, а також до ядра БД.

2. Запити призначені для пошуку, відбору, сортування, групування та аналізу інформації, що міститься у таблицях БД. Саме запити забезпечують реалізацію SQL-механізмів у середовищі Access, можуть створюватися як у графічному режимі за допомогою конструктора, так і шляхом безпосереднього написання SQL-коду. За допомогою запитів користувач може отримувати вибіркові дані, які відповідають певним умовам, виконувати обчислення, створювати підсумкові таблиці та аналізувати результати діяльності підприємства. Існує декілька основних типів запитів: запити на вибірку, запити на оновлення для змінювання даних в таблицях, запити на додавання (внесення нових записів). Крім того, Access підтримує перехресні запити та параметричні запити, які дозволяють формувати складні аналітичні вибірки.

3. Форми призначені для введення, перегляду та редагування інформації у зручному для користувача вигляді. Вони забезпечують графічний інтерфейс взаємодії з базою даних та дозволяють працювати з інформацією без безпосереднього доступу до структури таблиць.

4. Звіти використовуються для формування результатної інформації у вигляді документів, придатних для друку або електронного подання. За допомогою звітів у Microsoft Access можна створювати документи про реалізацію продукції, обсяги продажів, фінансові результати, товарні залишки та інші показники діяльності підприємства.

Пподальша робота зосереджена в СУБД MS Access із використанням функцій загальносистемного середовища, мови програмування VBA, майстрів, конструкторів таблиць, форм, звітів, запитів.

2.2 Обґрунтування принципів і моделі створення бази даних

Будь-яка БД у поєднанні із СУБД є інформаційною системою, основним призначенням якої є організація ефективного зберігання, обробки та використання інформації. Такі системи забезпечують цілісність даних, оптимізацію інформаційних потоків та підтримку процесів прийняття управлінських рішень.

Процес створення БД – це складний багатокomпонентний процес, який часто передбачає перегляд існуючих методів організації інформаційної діяльності підприємства. Розробка БД супроводжується частковим або повним реінжинірингом ІС економічного об'єкта, переосмисленням та вдосконаленням процесів роботи з інформацією з метою підвищення ефективності діяльності підприємства [20]. Проектування бази даних передбачає послідовне виконання комплексу взаємопов'язаних завдань, які представлені по крокам.

1. Визначаються основні характеристики майбутньої системи, її функціональне призначення, склад інформації та категорії користувачів..
2. Здійснюється побудова моделей предметних областей та визначення логічної структури даних..
3. Організація взаємодії людини з машиною, тобто створення зручного механізму роботи користувачів із системою.
4. Детальне проектування окремих компонентів бази даних, тестування та затвердження програмного продукту.
5. Впровадження системи в експлуатацію.

Одним з ключових етапів створення бази даних є визначення доцільності її розробки. На цьому етапі аналізуються потреби підприємства, оцінюються інформаційні потоки, виявляються проблеми існуючої системи обліку та обґрунтовується необхідність автоматизації процесів обробки інформації, оцінюються технічні, організаційні та економічні можливості БД (рис. 2.2).

Під час створення ІС необхідно дотримуватися загальновизнаних принципів проектування. Одним з основних є принцип системності, згідно з

яким база даних розглядається як цілісна система взаємопов'язаних елементів. Це дозволяє забезпечити узгоджену роботу окремих компонентів та ефективну організацію інформаційних процесів.



Рисунок 2.2 – Послідовність визначення характеристик БД через обстеження

Під час створення інформаційної системи необхідно дотримуватися загальновизнаних принципів проектування. Одним з основних є принцип системності, згідно з яким база даних розглядається як цілісна система взаємопов'язаних елементів. Це дозволяє забезпечити узгоджену роботу окремих компонентів та ефективну організацію інформаційних процесів [21].

Принцип сумісності передбачає можливість взаємодії інформаційної системи з іншими програмними продуктами та інформаційними ресурсами.

Принцип ефективності передбачає досягнення максимальної ефективності роботи системи при мінімальному споживанні ресурсів. База даних повинна забезпечувати швидкий доступ до інформації, оптимальне використання пам'яті та мінімізацію дублювання даних.

Не менш важливим є принцип стандартизації та уніфікації. Його дотримання забезпечує використання єдиних підходів до структури даних, методів їх обробки та організації взаємодії користувача з системою.

Під час розробки баз даних важливу роль відіграє концепція життєвого циклу (ЖЦ) програмного забезпечення. ЖЦ програмного продукту охоплює весь

період його існування – від моменту виникнення ідеї створення системи до завершення її використання та виведення з експлуатації. Кожен етап ЖЦ включає певний набір робіт, пов'язаних із аналізом вимог, проєктуванням, впровадженням, тестуванням, реалізацією та обслуговуванням системи. За визначенням «...життєвий цикл ПЗ – певна послідовність фаз або стадій від моменту прийняття рішення про необхідність створення ПЗ до повного вилучення ПЗ з експлуатації» [23].

У сучасній практиці розробки програмного забезпечення використовуються різні моделі життєвого циклу. До найвідоміших належать каскадна, ітеративна, інкрементальна та спіральна моделі [24].

Каскадна або водоспадна модель передбачає послідовне виконання всіх етапів розроблення, коли перехід до наступного етапу здійснюється лише після завершення попереднього. Перевагою такої моделі є чітка структурованість процесу, однак її недоліком є недостатня гнучкість при зміні вимог до програмного продукту.

Спіральна модель поєднує елементи послідовного та ітеративного підходів і передбачає поступове уточнення вимог та функціональних можливостей системи. Вона орієнтована на мінімізацію ризиків при створенні складних інформаційних систем.

Для розроблення баз даних найбільш доцільною вважається інкрементально-ітеративна модель життєвого циклу. Такий підхід передбачає поетапне створення та вдосконалення системи шляхом реалізації окремих функціональних компонентів. Кожна ітерація являє собою завершений етап розроблення, результатом якого є працездатна версія програмного продукту з певним набором функцій.

Основною перевагою інкрементального підходу є можливість поступового нарощування функціональності системи та адаптації програмного продукту до змін вимог користувачів. Це особливо важливо при створенні баз даних для малого бізнесу, оскільки дозволяє поступово вдосконалювати систему без необхідності повного перепроеєктування.

Найчастіше таку змішану еволюційну модель називають просто ітеративною (говорячи про процес) та/або інкрементальною (говорячи про нарощування функціональності продукту). Схематично дія моделі показана на рис. 2.3.

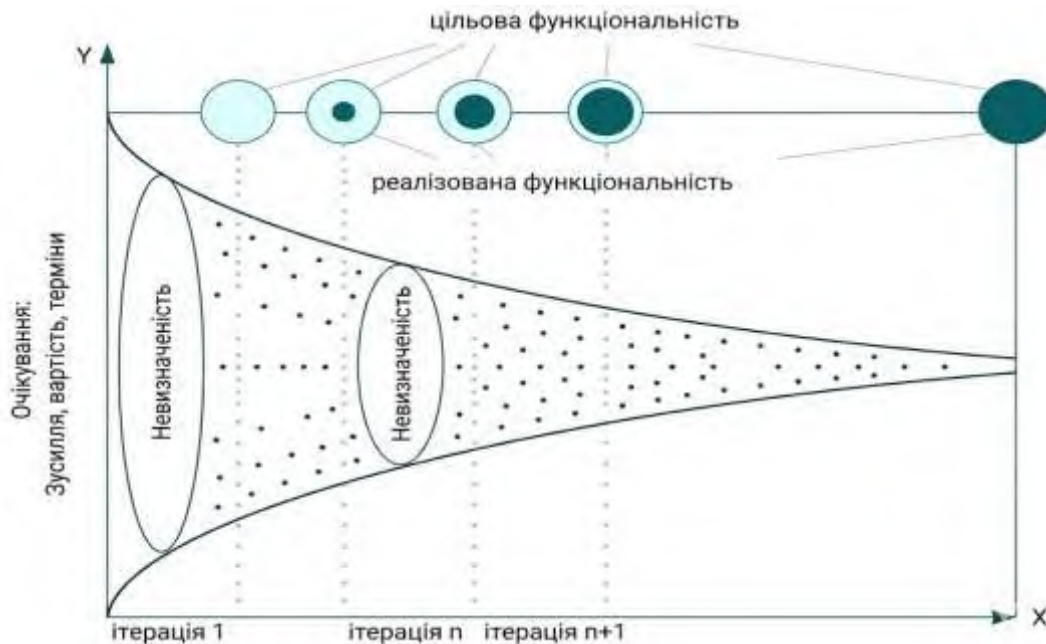


Рисунок 2.3 – Зниження невизначеності та інкрементальне розширення функціональності при ітеративній організації життєвого циклу [23]

У процесі проектування БД необхідно враховувати вимоги до структури інформації, швидкості обробки даних, захисту інформації, зручності інтерфейсу та можливості подальшого розширення системи. Крім того, важливими є вимоги щодо забезпечення цілісності даних, підтримки багатокористувацького режиму роботи та автоматизації виконання типових операцій.

Концептуальний підхід до створення бази даних передбачає формування цілісної моделі предметної області, визначення основних сутностей, атрибутів та зв'язків між ними. Саме концептуальне проектування є основою побудови логічної та фізичної структури бази даних і забезпечує ефективність подальшої реалізації інформаційної системи.

Функціональна структура комплексу бази даних, що розроблена в даній роботі, показана на рис. 2.4. Структура регламентує послідовність

функціонування програмної системи, формує схему рішення задач, визначає схему використання довідкових даних, схему взаємодії між різними об'єктами бази даних, забезпечує діалогову взаємодію з користувачем і коректне маніпулювання даними.

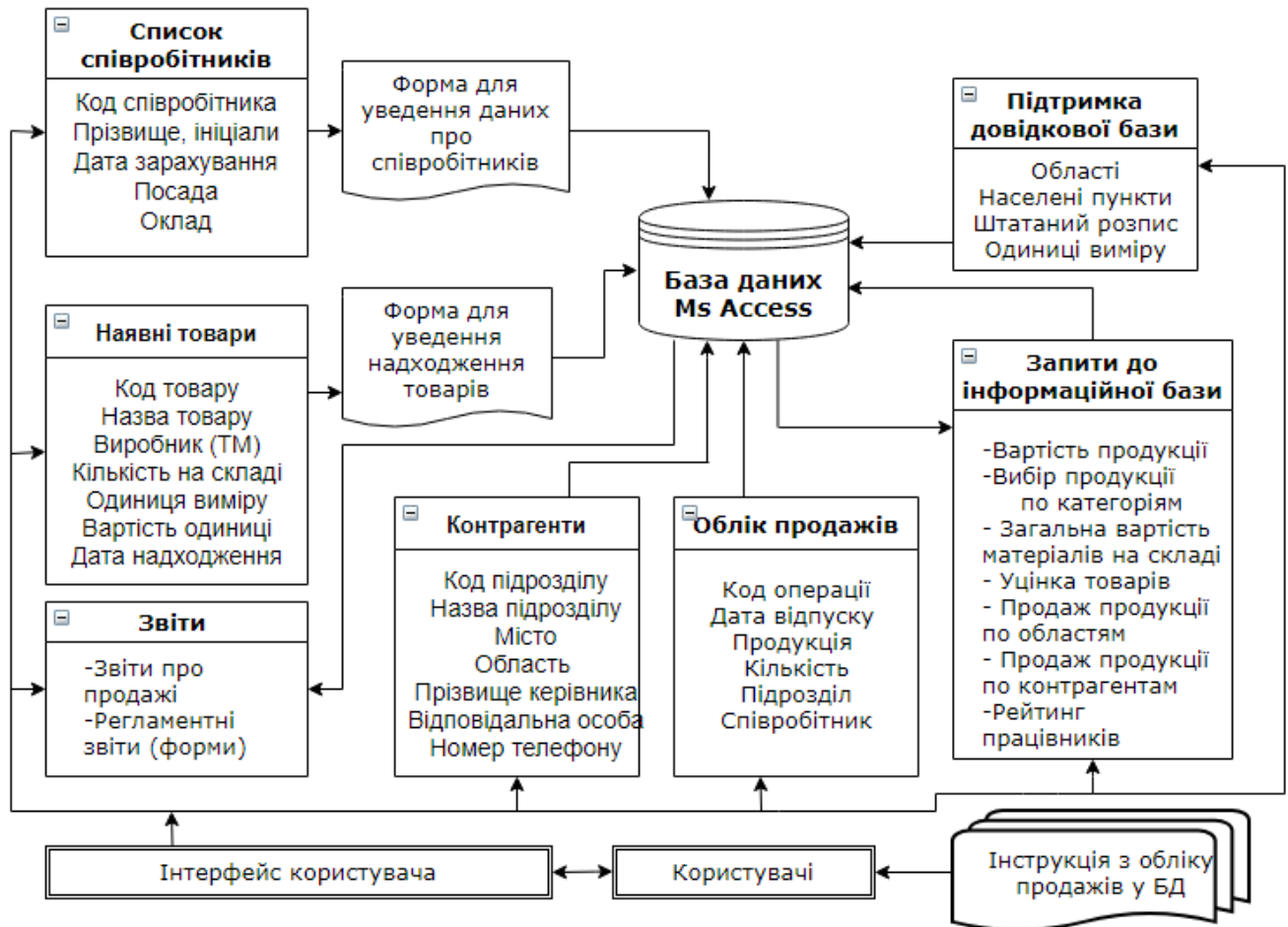


Рисунок 2.4 – Концептуальна структура функціонування бази даних з автоматизації збуту продукції в сфері малого бізнесу

Процес проєктування БД являє собою послідовність переходів від неформального словесного опису інформаційної структури предметної області (ведення товарного обліку, реєстру продажів, даних про співробітників кожного відділу, контрагентів) до формалізованого опису об'єктів предметної області в термінах реляційної моделі та фізичної реалізації бази даних в середовищі СУБД MS Access, побудові кожного з об'єктів із задаванням властивостей, реляційних зв'язків тощо.

2.3 Проектування фізичної структури та зв'язків реляційної бази даних в середовищі СУБД MS Access

Процес реалізації бази даних починається зі створення файлу бази даних, наприклад «Збут_продукції.accdb», а також створення всіх об'єктів БД, передбачених попередньо функціональною структурою (див. рис. 2.4). В якості даних були використані загальноживані назви різноманітних промислових товарів, назви посад працівників, інші дані, які дозволяють моделювати процеси реальних відносин на основі узагальнень.

На першому етапі створюються довідкові таблиці «Співробітники», «Наявні товари», «Контрагенти», «Відпуск товарів» в режимі конструктора із визначення переліку полів, властивостей полів, а також визначення поля первинного ключа в кожній таблиці. Для кожного поля визначається тип даних, який встановлює формат зберігання інформації. У Microsoft Access використовуються текстові, числові, логічні, грошові типи даних, а також типи дата/час, МЕМО, лічильник та інші. Правильний вибір типів даних забезпечує ефективне зберігання інформації та коректне виконання операцій обробки даних. На рис. 2.5 наведено етап встановлення типів та властивостей даних таблиці «Співробітники».

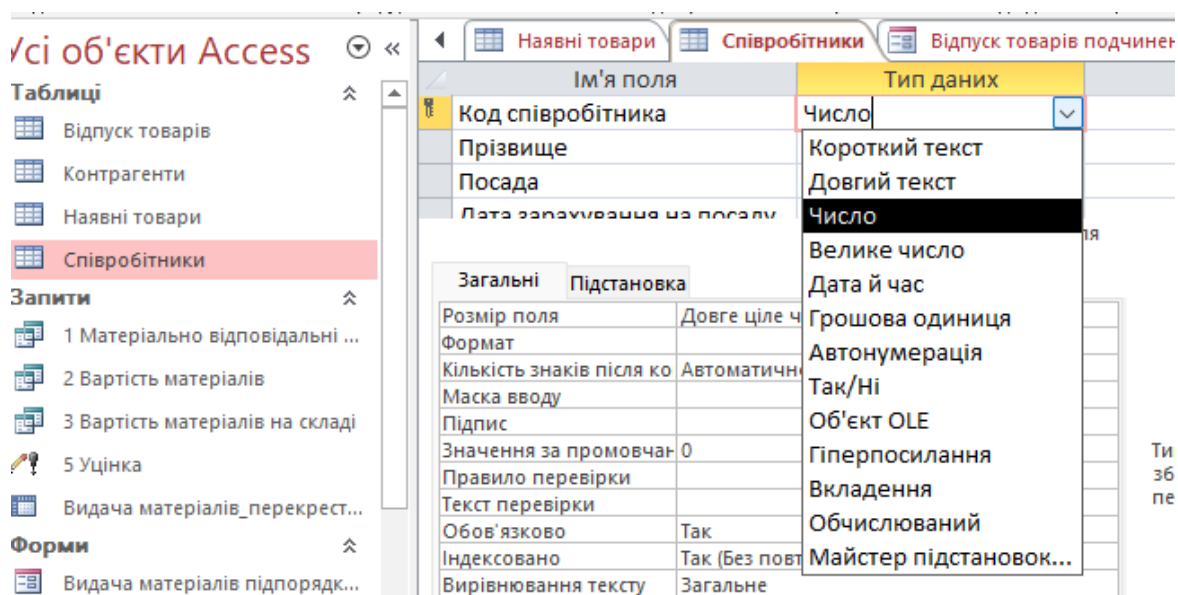


Рисунок 2.5 – Встановлення типів даних та властивостей полів таблиці БД

Списки підстановки формуються під час створення проекту таблиці в режимі конструктора через Майстра підстановки, що викликається зі списку «Тип даних». Приклад використання фіксованого переліку зі списку наведено на рис. 2.6 під час уведення даних по одиницях виміру одиниці товару в таблиці «Наявні товари». Такі списки, по суті, відграють роль довідників.

Наявні товари							
Код товару	Назва товару	Виробник (ТМ)	Кількість на складі	Одиниця виміру	Вартість одиниці	Дата надходження	
11	Сірки	УСФ	1000	уп	20,00грн	17.09.	
12	Сіль	Карпати-сіль	150	кг	18,50грн	18.09.	
13	Мило господарське	ХЗ-86	600	шт	11,40грн	26.09.	
14	Цукор	Хуторок	500	уп	38,20грн	27.09.	
15	Цукор ваговий	Лохвицький с/з	468	уп 1х6х1л	31,40грн	28.09.	
16	Гречка	Хуторок	271	уп 1х4х2л	74,20грн	01.10.	
17	Горох	Форум	525	уп 1 кг	24,60грн	02.10.	
18	Вода питна	Миргородська	760	уп 0.8 кг	108,00грн	15.10.	
19	Борошно пшеничне	Хуторок	950	кг	26,00грн	25.10.	
20	Вода мінеральна	Оболонь	70	уп	88,60грн	14.02.	
21	Пшоно	Маяк	80	кг	28,40грн	13.03.	

Рисунок 2.6 – Заповнення даних таблиці «Наявні товари» з використанням фіксованого переліку (списку підстановки)

Зв'язування даних окремих таблиць відбувається автоматично під час проектування окремих полів таблиць в режимі «Конструктор» і створення списків підстановки на основі використання існуючих полів інших таблиць БД (рис. 2.7).

Ім'я поля	Тип даних	Опис
Ідентифікатор	Автонумерація	
Назва компанії	Короткий текст	
Марка автомобіля	Короткий текст	
Вантажопідйомність	Число	
Замовник	Короткий текст	

Майстер підстановок

Цей майстер створює поле підстановки, яке відображає список значень, які можна вибрати. Яким чином поле підстановки має отримати значення?

☒ поле підстановки має отримати значення з іншої таблиці або запиту

☐ Я самостійно введу потрібні значення

Рисунок 2.7 – Робота з «Майстер підстановок» для вибору підстановки даних

Основні переваги СУБД реалізуються при роботі не з окремими таблицями, а з групами взаємопов'язаних таблиць. На наступних етапах необхідно обрати таблиці, з яких і підставляються дані при заповненні таблиць. Полів для підстановки може бути обрано не одне, як видно з рис. 2.8.

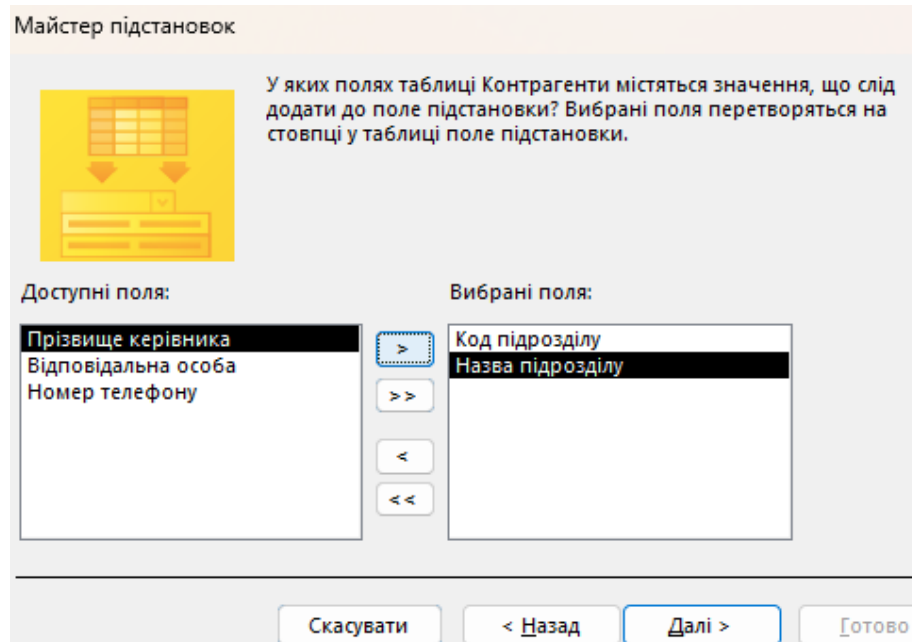


Рисунок 2.8 – Вибір полів з таблиць для списку підстановки у таблиці «Логістичні компанії» для поля «Замовник»

При заповненні більшості таблиць, наприклад, «Відпуск матеріалів», використовуються підстановки даних зі сформованих раніше списків у поля «Співробітник» і «Матеріали». Ключове поле при цьому приховане (рис. 2.9).

Наявні товари		Логістичні компанії		Відпуск товарів			
Код операції	Дата видачі	Матеріал	Кількість	Підрозділ	Співробітник	Кликайть, щоб додати	
101	01.02.2026	12	35	10011	Антонов С.В.		
102	15.03.2026	13	74	10014	Коваленко А.С.	Керівник відділу	
103	13.10.2025	14	40	10011	Антонов С.В.	Менеджер з продажів	
104	24.11.2025	16	21	10013	Михайлова О.І.	Бухгалтер	
105	12.01.2026	17	10	10012	Кравченко І.В.	Менеджер з продажів	
106	09.04.2026	11	54	10011	Скриль В.С.	Менеджер з продажів	
107	16.02.2026	12	45	10014	Власенко Р.О.	Вантажник	
108	03.01.2026	13	37	10011	Кириленко О.І.	Вантажник	
109	10.11.2025	15	59	10017	Міркунов О.Є.	Інженер	
110	15.03.2026	13	48	10013	Мірова А.П.	Касир	
111	26.03.2026	16	14	10012	Скриль В.С.		
112	18.01.2026	11	35	10014	Антонов С.В.		

Рисунок 2.9 – Приклад виконання підстановки зі списку для заповнення поля

За кроками, що описані на рис.2.5-2.9, створюються й інші сутності. Зміст заповненої таблиці БД «Відпуск товарів» представлено в додатку А. Схема зв'язків між таблицями бази даних показано у вікні «Зв'язки» на рис. 2.10.

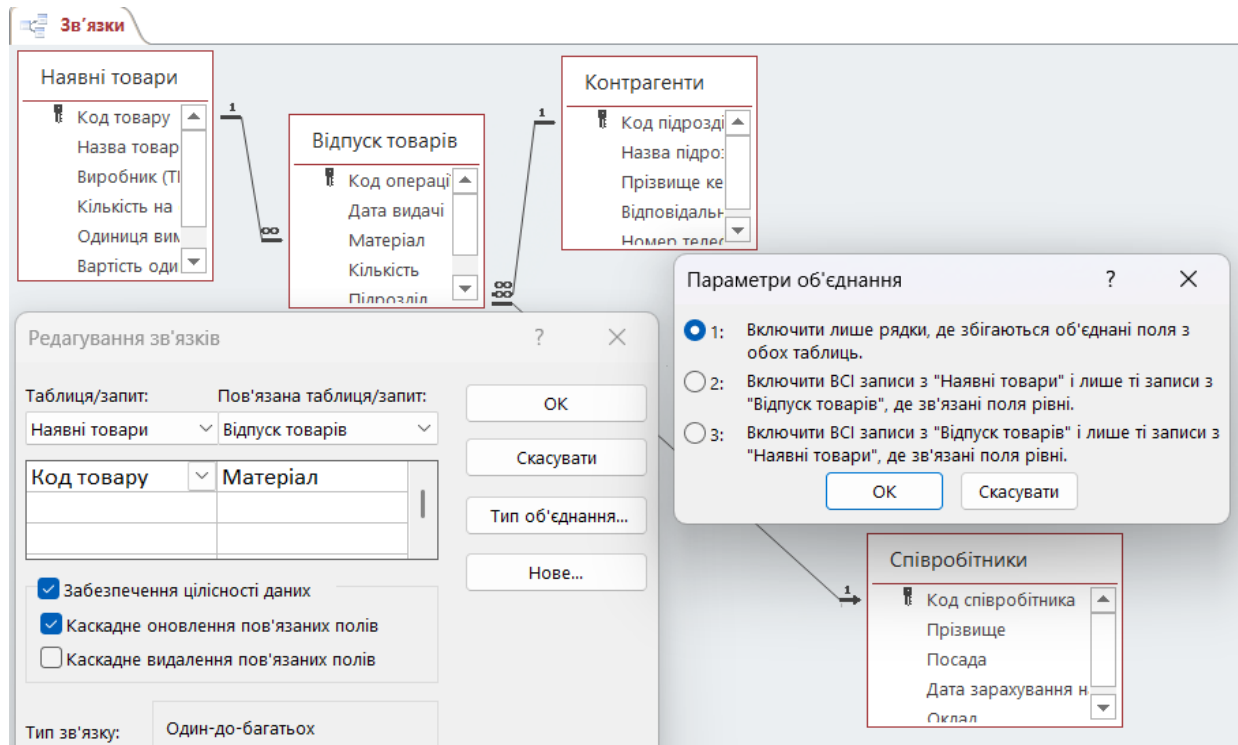


Рисунок 2.10 – Схема даних бази даних Збут_продукції.accdb із вікном налаштування/зміни властивостей зв'язків та параметрів об'єднання

У вікні «Зв'язки» середовища Microsoft Access типи міжтабличних зв'язків визначаються автоматично на основі властивостей пов'язаних полів, а їх налаштування здійснюється за допомогою діалогових вікон «Редагування зв'язків» та «Параметри об'єднання» (рис. 2.10). Використання зв'язків між таблицями забезпечує цілісність даних, усуває дублювання інформації та дозволяє організувати ефективну взаємодію між окремими об'єктами бази даних. Типи зв'язків є відображенням логічних взаємозв'язків між даними, що містяться в різних таблицях [25]. У середовищі Microsoft Access найчастіше використовуються такі типи зв'язків:

– «один-до-одного» (1:1) – одному запису в першій таблиці відповідає лише один запис у другій таблиці. Такий тип зв'язку застосовується порівняно

рідко та використовується переважно для розділення великих таблиць або обмеження доступу до окремих даних;

- «один-до-багатьох» (1:∞) – одному запису головної таблиці може відповідати декілька записів у пов'язаній таблиці. Це найбільш поширений тип зв'язку в реляційних базах даних;

- «багато-до-багатьох» (∞:∞) – одному запису однієї таблиці можуть відповідати декілька записів іншої таблиці і навпаки. Такий зв'язок реалізується через проміжну таблицю, яка містить зовнішні ключі обох таблиць.

У розробленій БД використано переважно зв'язки типу «один-до-багатьох», оскільки саме вони найбільш повно відповідають структурі предметної області та забезпечують ефективну організацію даних.

Редагування зв'язків у вікні «Редагування зв'язків» дозволяє керувати правилами взаємодії між таблицями та підтримувати цілісність інформації. Для цього використовуються відповідні параметри налаштування:

- «Забезпечення цілісності даних» унеможливорює створення записів у підпорядкованій таблиці без відповідного запису в головній таблиці, а також запобігає видаленню ключових записів, які є у зв'язаних таблицях;

- «Каскадне оновлення зв'язаних полів» забезпечує автоматичне оновлення значень зовнішніх ключів у пов'язаних таблицях у разі зміни значення первинного ключа;

- «Каскадне видалення зв'язаних записів» автоматично видаляє всі пов'язані записи у підпорядкованих таблицях при видаленні такого запису головної таблиці.

Використання каскадних операцій дозволяє підтримувати узгодженість даних та уникати появи так званих «висячих» записів, які не мають відповідного зв'язку з основною таблицею. При створенні та редагуванні міжтабличних зв'язків необхідно дотримуватись певних правил:

Перед встановленням або зміною зв'язків таблиці повинні бути закриті, оскільки відкриті об'єкти можуть блокувати внесення змін до структури бази даних.

Поля, між якими створюється зв'язок, повинні мати однаковий тип даних та сумісні властивості. Наприклад, поле первинного ключа типу «Лічильник» повинно відповідати зовнішньому ключу типу «Числовий».

Зв'язок створюється між первинним ключем головної таблиці та зовнішнім ключем підпорядкованої таблиці. При цьому значення первинного ключа повинні бути унікальними, тоді як у зовнішньому ключі допускається повторення значень. Після створення зв'язків їх можна редагувати, видаляти або доповнювати новими параметрами відповідно до потреб інформаційної системи.

Особливе значення при налаштуванні зв'язків мають параметри об'єднання таблиць. У Microsoft Access доступні різні варіанти об'єднання даних, які визначають спосіб відображення записів у запитах. Параметри об'єднання другого та третього типу дозволяють включати до результатів запиту всі записи однієї або обох зв'язаних таблиць незалежно від наявності відповідних значень у пов'язаній таблиці. Це забезпечує гнучкість під час формування вибірок та аналітичних запитів.

Таким чином, створення міжтабличних зв'язків є одним із ключових етапів проєктування реляційної бази даних. Саме зв'язки забезпечують логічну цілісність інформації, ефективну взаємодію між таблицями та можливість реалізації складних SQL-запитів. Після завершення етапів створення таблиць, встановлення зв'язків і внесення початкових даних база даних стає готовою до подальшого використання, формування запитів, розроблення користувацького інтерфейсу та створення форм і звітних документів.

РОЗДІЛ 3

РОЗРОБЛЕННЯ І ЗАСТОСУВАННЯ СИСТЕМИ БАЗИ ДАНИХ НА ПРИКЛАДІ ЗБУТОВОЇ ДІЯЛЬНОСТІ ПІДПРИЄМСТВА

3.1 Функції та використання запитів у системі управління базою даних

Для отримання необхідної інформації з БД у заданому вигляді в Microsoft Access використовуються запити. Запит є спеціальним об'єктом бази даних, призначеним для вибірки, обробки, аналізу та зміни даних, що зберігаються у таблицях. У Microsoft Access підтримується декілька основних типів запитів [20]:

- запит на вибірку – призначений для отримання інформації з однієї або декількох таблиць відповідно до заданих умов відбору;
- запит із параметром – дозволяє вводити значення параметрів під час виконання запиту, що забезпечує гнучкість формування вибірки даних;
- підсумковий запит – використовується для виконання статистичних та підсумкових обчислень, зокрема визначення суми, середнього значення або ін.;
- перехресний запит – забезпечує подання результатів у вигляді таблиці з групуванням даних за рядками і стовпцями для аналітики інформації;
- запит на оновлення – використовується для автоматичної зміни значень у таблицях бази даних;
- запит на додавання – дозволяє переносити або копіювати записи до іншої таблиці;
- запит на видалення – забезпечує вилучення записів за умовами.

При виконанні запиту вихідні дані можна сортувати, фільтрувати, об'єднувати, змінювати. На відміну від простого фільтру запит може використовувати вибірккові дані багатьох таблиць, пов'язаних між собою відношеннями. Крім стандартних функцій, запити дозволяють використовувати власні вирази та функції мови SQL для виконання складніших розрахунків. Під час розроблення бази даних запити створюються у режимі Конструктора, який забезпечує графічне формування SQL-виразів.

У результаті виконання запиту формується спеціальна динамічна таблиця результатів (Dynaset), яка містить інформацію відповідно до умов запиту. Така таблиця є тимчасовою і не змінює вміст самих таблиць, за винятком спеціальних запитів на оновлення, додавання або видалення даних. Запити готуються в «Конструкторі» у спеціальному бланку. Кожному запиту присвоюється ім'я. На рис. 3.1 представлені етапи формування запиту із додатковим підсумковим полем про вартість відпущених матеріалів.

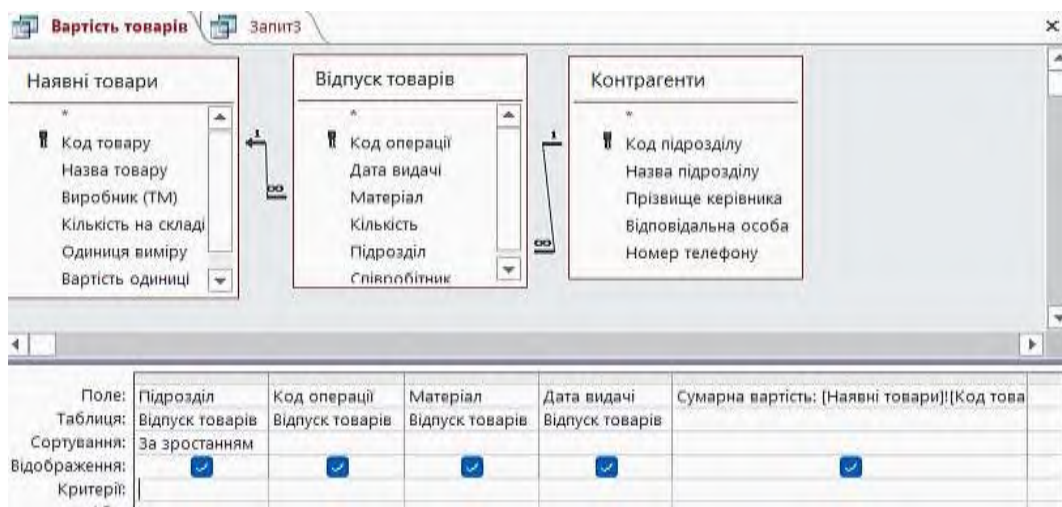


Рисунок 3.1 – Створення запиту в режимі конструктора

У бланку запиту створюється додаткове поле «вартість матеріалів» із обчисленням за груповою операцією (див. рис. 3.1). Формулу можна редагувати за допомогою інструменту «Побудовник виразів», де представлені елементи виразу – структура бази даних та вбудовані функції (рис. 3.2).

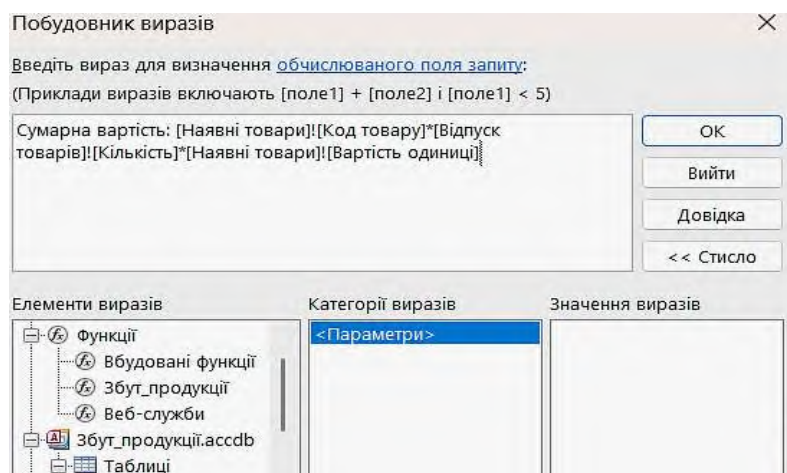


Рисунок 3.2 – Побудови формули запиту у вікні «Побудовник виразів»

Ефективним засобом редагування структури запиту та його логіки є мова обробки запитів SQL [26]. Перехід у режим SQL із режиму Конструктора дозволяє безпосередньо вносити зміни до тексту SQL-виразу, уточнювати умови вибірки, параметри сортування, групування та обчислення даних (рис. 3.3).

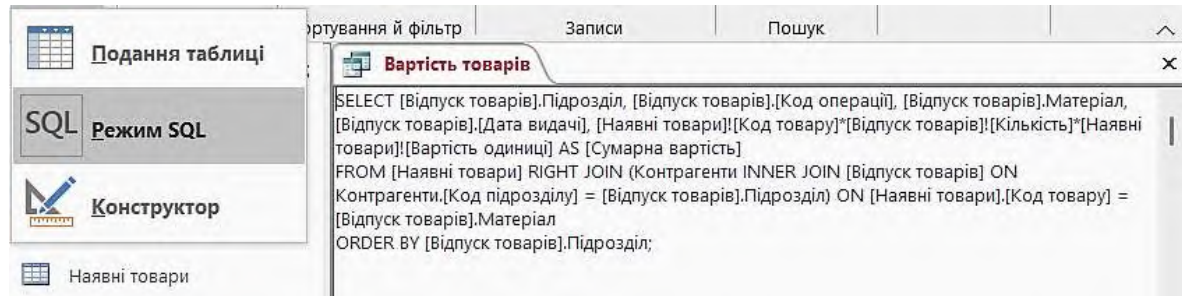


Рисунок 3.3 – Повний запис виразу запиту з підсумками з операторами SQL

Результатом виконання запиту є тимчасова віртуальна таблиця, яка відображає результати обробки відповідно до поточного стану даних у базі (додаток Б). Додатковими умовами вибірки можуть бути обмеження за певним періодом часу, конкретними контрагентами, видами продукції тощо.

У процесі збутової діяльності іноді необхідно переглядати ціни на певні категорії товарів. Для цього використовують запити «!Оновлення», під час виконання яких оновлюються окремі дані в таблицях. На рис. 3.4 представлено елементи побудови запиту на оновлення ціни (уцінка) наявних товарів.

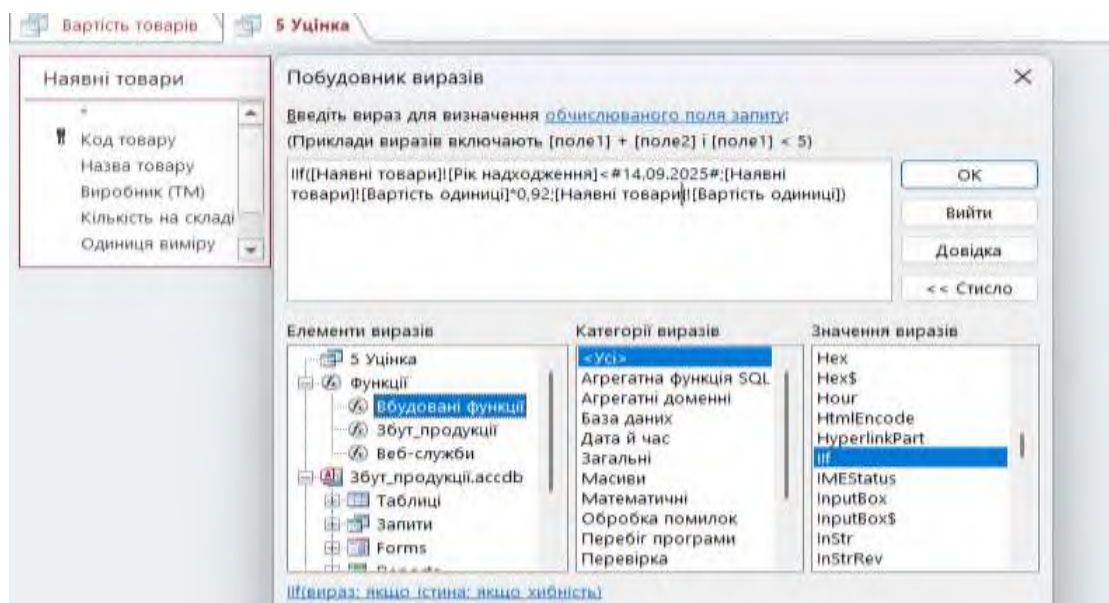


Рисунок 3.4 – Побудова виразу для оновлення даних в полі таблиці

У верхній частині бланку запиту обрано таблицю «наявні матеріали» (див. рис. 3.4) в нижній частині бланку в рядку «Оновлення до:» записана функція, згідно якої відбувається перерахунок даних поля «Вартість одиниці». Вираз побудований у вікні «Побудовник виразів». Обрано вбудовану функцію If із категорії «Перевірка». У полі для введення виразу записано умову для обрання даних, які будуть перераховані. Наприклад, уцінюються на 8 % товари, що надійшли раніше зазначеного терміну (див. рис. 3.4). При запуску кнопкою «!Запуск» з'являється вікно попередження про зміну даних в таблиці (рис. 3.5).

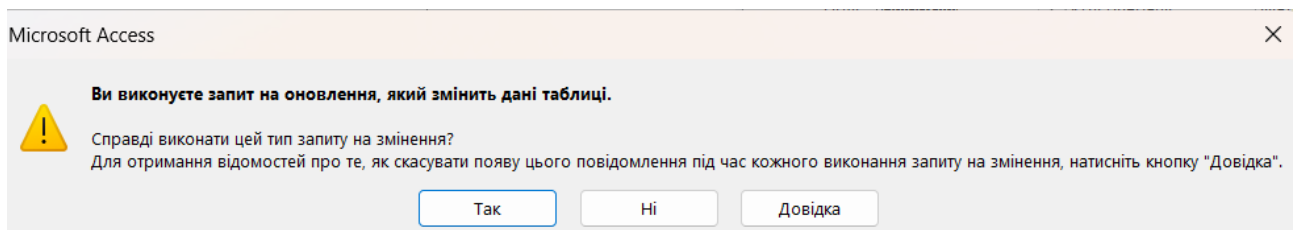


Рисунок 3.5 – Діалогове вікно-попередження про внесення змін до таблиці

Результат виконання запиту можна продивитися безпосередньо в таблиці та усіх пов'язаних із нею об'єктах – формах, інших таблицях і запитах, в яких, можливо, фігурує зв'язане поле. До однієї таблиці можна створювати різні запити на оновлення із різними умовами.

Умови запиту можна відкоригувати за потреби. Існує особливий інструмент – вікно параметру для уведення певних умов запиту безпосередньо в момент виконання запиту (рис. 3.6).

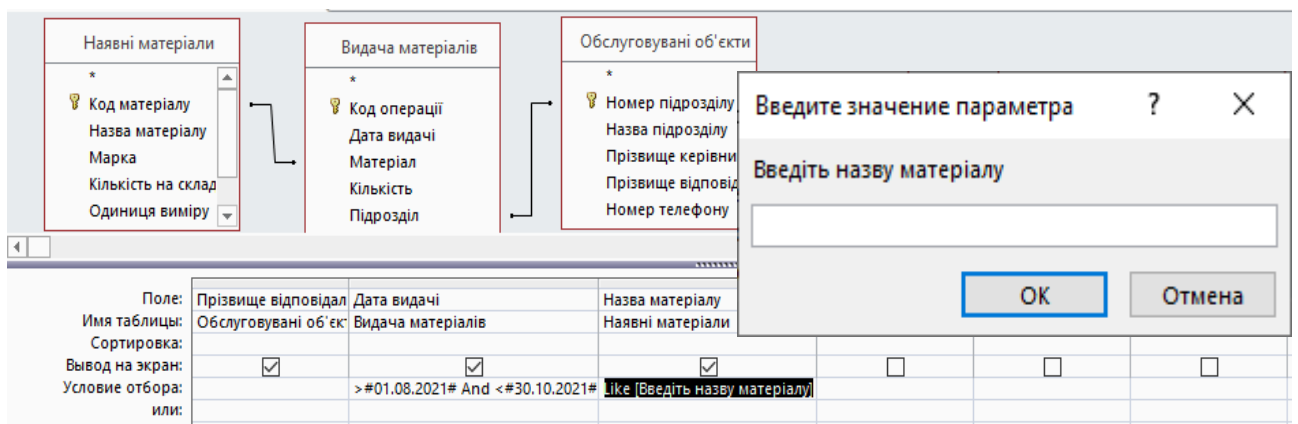


Рисунок 3.6 – Конструктор запиту з параметром і вікно параметра при запуску

У прикладі, що наведено на рис. 3.6, показано синтаксис запису вікна параметра із текстом запитання (Like [Введіть назву матеріалу]), яке з'являється над текстовим полем. Для наочності у запит було введено умову відбору в полі «Дата видачі» для матеріалів, які будуть обрані по такому запиту – проміжок часу між двома конкретними датами і логічна функція AND, яка приймає істинне значення при вірності обох логічних виразів формули.

Отже, за допомогою запитів СУБД Access здійснює достатньо великий спектр операцій із даними, надає можливість ефективного управління ними за рахунок точного відбору, групування за різними атрибутами, внесення змін у поля та видачі актуальних результатів. Кількість запитів у створеній базі даних не обмежена. Оскільки таблиці і запити є основними об'єктами бази даних, які забезпечують результативність її роботи, то значну увагу приділяють безпеці і цілісності структури таблиць і звітів. Досягається, зокрема, за рахунок організації інтерфейсу користувача.

3.2 Конструювання форм та звітів для управління БД

Форми є важливою складовою інтерфейсу користувача в середовищі Microsoft Access. Вони використовуються для введення, перегляду, редагування та пошуку даних у базі даних. Форма являє собою електронний бланк, який забезпечує зручне відображення інформації з таблиць або запитів та спрощує взаємодію користувача із системою. Зв'язок між формою та джерелом даних реалізується за допомогою спеціальних елементів керування [27]: поля, кнопки, списки, перемикачі, написи та інші компоненти інтерфейсу. Найчастіше для введення та відображення інформації використовуються поля, у яких відображаються значення відповідних атрибутів таблиці.

Для однієї таблиці може бути створено декілька різних форм залежно від функціонального призначення та категорії користувачів. Водночас одна форма може забезпечувати доступ до даних із декількох взаємопов'язаних таблиць.

Форми в Microsoft Access можуть створюватися різними способами у вкладці «Створити»: за допомогою автоформи, майстра форм або конструктора форм. Автоматичне створення форм забезпечує швидке формування стандартного інтерфейсу, тоді як майстер форм дозволяє налаштовувати склад полів та структуру форми. Режим Конструктора використовується для детального налаштування форми: редагувати підписи полів, змінювати структуру та дизайн форми, додавати кнопки керування, обчислювані поля та інші елементи.

Наприклад, для автоформи «Товари» в конструкторі додано додаткове поле із формулою, яке обчислює загальну вартість партії товару (рис. 3.7).

Код товару	Код товару
Назва товару	Назва товару
Виробник (ТМ)	Виробник (ТМ)
Кількість на складі	Кількість на ск
Одиниця виміру	Одиниця виміру
Вартість одиниці	Вартість одиниці
Дата надходження	Дата надходже
Вартість партії, грн	= [Вартість одиниці] * [Кількість на складі]

Рисунок 3.7 – Розробка форми в режимі «Конструктор» із полем для обчислень

Вигляд такої форми в режимі форми представлено на рис. 3.8. У полі «Вартість партії, грн» виводиться результат обчислення за формулою.

Товари	
Код товару	19
Назва товару	Борошно пшеничне
Виробник (ТМ)	Хуторок
Кількість на складі	950
Одиниця виміру	кг
Вартість одиниці	26.00грн
Дата надходження	25.10.2021
Вартість партії, грн	24700

Рисунок 3.8 – Екранна форма перегляду записів із полем для обчислення

При управління збутом продукції важливо відслідковувати як відвантаження продукції, так і мати дані про її наявність на складі. Побудова підпорядкованої форми надає можливість поєднати перегляд даних одразу двох таблиць (рис. 3.9).

Склад

Код матеріалу:

Назва матеріалу:

Марка:

Кількість на складі:

Одиниця виміру:

Вартість одиниці:

Видача матеріалів

Код операції	Дата видачі	Підрозділ	Кількість
127	06.01.2022	Кишеньки	100
112	18.01.2022	Фуршет	35
117	09.03.2022	Продукти	30

Залишок: 1000 | 1 з 4 | Відкрити | Додати | Відмінити | Поиск

Рисунок 3.9 – Складена форма збуту продукції на основі двох таблиць

Для створення підпорядкованої форми варто скористатися майстром побудови форм на стрічці «Створити». На першому етапі визначають склад форми, обирають таблиці та поля з них (рис. 3.10).

Майстер форм

Які поля слід використати у формі?
Вибирати можна з кількох таблиць або запитів.

Таблиці та запити
Таблиця: **Контрагенти**

Доступні поля:

- Код підрозділу
- Назва підрозділу
- Прізвище керівника
- Відповідальна особа
- Номер телефону

Вибрані поля:

- Код операції
- Дата видачі
- Матеріал
- Кількість
- Підрозділ
- Співробітник

Скасувати < Назад Далі > Готово

Рисунок 3.10 – Вибір таблиць для підпорядкованої форми у вікні майстра

На наступних кроках розподіляють ролі головної та підпорядкованої таблиць, враховуються типи зв'язку між ними, задається назва форм (рис. 3.11).

Рисунок 3.11 – Завершення створення форми з підпорядкованою формою

У результаті буде отримана форма, яка наведена в додатку В. Додаткове редагування форми здійснюється в режимі конструктора у будь-який момент налаштування форм. Фрагмент вікна редагування із використанням властивостей показано на рис. 3.12.

Рисунок 3.12 – Засоби редагування форми БД в режимі «Конструктор»

Звіт – з числа об’єктів СУБД Access, який призначений для друку даних і форматування[28]. Робота зі створення звітів схожа на роботу з формами. Звіти можна створювати різними способами, але, як і у випадку зі створенням форми, можна використати «Майстер звітів» і оформити за власним бажанням. Автоматичний звіт найпростіше створити за даними певної таблиці, наприклад, перелік товарів на складі на дату (рис. 3.13).

Код товару	Назва товару	Кількість на складі
20	Вода мінеральна	70
21	Пшона	80
11	Сірки	1000
12	Сіль	150
13	Мило господарське	600
14	Цукор	500
15	Цукор ваговий	468
16	Гречка	271
17	Горох	525
18	Вода питна	760
19	Борошно пшеничне	950

20 травня 2025 р. Стр. 1 из 1

Рисунок 3.13 – автоматично створений звіт про наявність товарів

Складні звіти, до яких висуваються певні вимоги щодо розрахунків, для подання інформації до контрольних органів (регламентні звіти), створюють в режимі «Конструктор» (рис.3.14).

Контрагенти

Верхний колонтитул

Код подразделения	Назва підрозділу	Прізвище керівника	Дата видачі	Код операції	Матеріал	Вартість матеріалів
Номер підроз	Назва підроз	Прізвище кері				

Область данных

Дата видачі	Код операції	Матеріал	Вартість матеріалів
Примечание группы 'Номер подразделения'			
="Итого для " & "Номер подразделения" & " " & [Номер подразделения] & " (" & Count(*) & " " & If(Count(*)=1,"запись","записей") &			
Загальна сума			=Sum([Вартість

Нижний колонтитул

Примечание отчета

УСЬОГО

=Sum([Вартість

Рисунок 3.14 – Створення розрахункових полів та виразів в конструкторі звіту

Фрагмент сформованого звіту наведено на рис. 3.15. Повна сторінка представлена в додатку Г.

Контрагенти						
Код підрозділу	Назва підрозділу	Прізвище керівника	Дата видачі	Код операції	Матеріал	Вартість матеріалів
11	Сільпо	Сидоров	13.10.2021	103	Цукор	276,00 Р
			14.12.2021	113	Сіль	78,89 Р
			28.12.2021	126	Гречка	454,00 Р
			03.01.2022	108	Мило	96,20 Р
			01.02.2022	101	Сіль	78,89 Р
			09.04.2022	106	Срники	183,60 Р
Ітоги для 'Номер підрозділу' = 11 (6 записей)				Загальна сума		1 164,58р.
12	ПП Горобець	Горобець	14.12.2021	123	Сіль	67,62 Р
			12.01.2022	105	Горох	52,44 Р
			26.03.2022	111	Гречка	114,80 Р
			04.04.2022	128	Цукор вагов	380,00 Р
Ітоги для 'Номер підрозділу' = 12 (4 записей)				Загальна сума		614,86р.

Рисунок 3.15 – Приклад звіту по сумарним продажам різним контрагентам

Проаналізувавши результати створення форм і запитів, можна дійти висновку, що один раз створений запит може багаторазово використовуватися для отримання актуальної інформації без повторного налаштування параметрів вибірки. Це суттєво підвищує ефективність роботи користувачів. Завдяки використанню форм і спеціальної мови запитів SQL, можливо досягти достатньо високих результатів обробки даних, забезпечити ефективність цих процесів.

3.3 Економічна ефективність використання СУБД у збутовій діяльності підприємства

Використання систем управління базами даних у сфері збутової діяльності дозволяє автоматизувати значну кількість операцій, підвищити швидкість обробки інформації та зменшити вплив людського фактора на результати роботи підприємства. Саме тому оцінювання економічної ефективності впровадження

баз даних є важливою складовою процесу розроблення інформаційної системи [29]. Під економічною ефективністю інформаційної системи розуміють співвідношення між отриманими результатами від її використання та витратами на створення, впровадження й експлуатацію системи. Для підприємств малого бізнесу питання економічної доцільності є особливо важливим, оскільки фінансові ресурси таких підприємств часто є обмеженими.

Використання систем управління базами даних у збутовій діяльності забезпечує низку економічних переваг. Насамперед це скорочення часу на обробку інформації, автоматизація виконання рутинних операцій, підвищення точності обліку товарів і замовлень, зменшення кількості помилок у документації та покращення контролю за товарними залишками. Крім того, автоматизація дозволяє підвищити оперативність прийняття управлінських рішень та покращити якість аналітичної інформації [30].

Для оцінювання ефективності впровадження інформаційних систем у бакалаврських роботах доцільно використовувати спрощені методики економічного аналізу, які не потребують складних фінансових розрахунків, але дозволяють обґрунтувати доцільність використання програмного продукту. Однією з найпростіших методик є оцінювання економії робочого часу. Її суть полягає у порівнянні витрат часу на виконання певних операцій до та після впровадження бази даних. Розрахунок економії часу можна подати у вигляді:

$$E_t = (T_1 - T_2) \cdot N, \quad (3.1)$$

де E_t – економія часу;

T_1 – час виконання операції до автоматизації;

T_2 – час виконання операції після автоматизації;

N – кількість операцій за певний період.

Наприклад, до автоматизації працівник витрачав у середньому 20 хвилин на оформлення одного замовлення, включаючи пошук інформації про товар, перевірку залишків та формування документації. Після впровадження бази даних цей процес скоротився до 7 хвилин.

Якщо підприємство оформлює 25 замовлень на день, то економія часу становитиме: $E_t = (20 - 7) \cdot 25 = 325$ хв. Отже, за один робочий день підприємство економить 325 хв. або понад 5 год. робочого часу.

Ще одним простим методом є оцінювання економії витрат на оплату праці. Якщо скорочення часу виконання операцій дозволяє зменшити навантаження на персонал, можна визначити умовний економічний ефект:

$$E_p = E_t \cdot C, \quad (3.2)$$

де E_p – економічний ефект;

E_t - зекономлений час;

C - артість однієї години роботи працівника.

Наприклад, якщо середня вартість години роботи працівника становить 170 грн, а економія часу за місяць 110 годин, то в грошовому вираженні:

$E_p = 110 \cdot 167 = 18700$ грн з розрахунку на 1 працівника. Це еквівалентно заробітній платі персоналу вищої кваліфікації [31].

Для оцінки ефективності також можна використовувати показник терміну окупності системи. Він показує, за який період часу витрати на створення інформаційної системи компенсуються отриманим економічним ефектом:

$$T = \frac{K}{E}, \quad (3.3)$$

де T – термін окупності;

K – витрати на створення системи;

E – середній економічний ефект за певний період.

Наприклад, якщо розроблення та впровадження бази даних коштувало підприємству 98600 грн, а щомісячний економічний ефект становить 18700 грн, то: $T = 98600 \div 18700 \approx 5,3$ міс.

Отже, система окупиться приблизно за півроку роботи. Крім кількісних показників, при оцінюванні ефективності баз даних важливо враховувати і якісні результати впровадження системи. До таких результатів належать:

- підвищення точності обліку даних;
- зменшення кількості помилок у документації;
- покращення контролю за товарними залишками;
- прискорення пошуку інформації;
- підвищення якості обслуговування клієнтів;
- спрощення формування звітності;
- покращення умов праці персоналу.

Для малого бізнесу важливим фактором економічної ефективності є також використання доступних програмних засобів. Застосування Microsoft Access дозволяє створити функціональну базу даних без значних фінансових витрат на придбання дорогого серверного програмного забезпечення або складної ІТ-інфраструктури. Крім того, система не потребує високих технічних вимог та спеціалізованого адміністрування.

У процесі оцінювання ефективності необхідно враховувати не лише прямий економічний ефект, але й перспективи подальшого розвитку підприємства. Автоматизація збутової діяльності створює основу для масштабування бізнесу, підвищення конкурентоспроможності та вдосконалення управлінських процесів.

Таким чином, використання систем управління базами даних у збутовій діяльності підприємства забезпечує як економічні, так і організаційні переваги. Навіть застосування найпростіших методик оцінювання дозволяє підтвердити доцільність впровадження бази даних та обґрунтувати ефективність автоматизації процесів управління збутом продукції малого підприємства.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було досліджено теоретичні та практичні аспекти автоматизації управління збутом продукції малого підприємства на основі використання SQL-запитів у середовищі Microsoft Access. Проведене дослідження дозволило сформулювати такі основні висновки.

1. Встановлено, що в умовах цифровізації економічної діяльності ефективність функціонування підприємств значною мірою залежить від рівня автоматизації процесів обробки інформації. Для малого бізнесу особливого значення набуває використання доступних інформаційних систем, здатних забезпечити оперативний облік, контроль та аналіз збутової діяльності..

2. У процесі дослідження розкрито теоретичні основи функціонування систем управління базами даних, визначено їх роль у забезпеченні цілісності, структурованості та ефективної обробки інформації. Проаналізовано основні моделі представлення даних та обґрунтовано доцільність використання реляційної моделі для реалізації системи управління збутом продукції.

3. Досліджено можливості мови SQL як основного інструменту роботи з даними у сучасних СУБД. Визначено особливості використання SQL-запитів для вибірки, сортування, групування, оновлення та аналізу інформації. Встановлено, що застосування SQL-запитів дозволяє значно підвищити ефективність обробки даних і автоматизувати виконання типових операцій у системі збутової діяльності підприємства.

4. Проведено аналіз предметної області збутової діяльності малого підприємства та визначено основні інформаційні об'єкти системи: товари, клієнти, замовлення, постачальники, операції продажу та складський облік. На основі проведеного аналізу сформовано концептуальний підхід до проектування структури бази даних.

5. Обґрунтовано вибір Microsoft Access як програмного середовища для реалізації бази даних. Визначено переваги даної СУБД для малого бізнесу, серед

яких простота використання, підтримка SQL-запитів, наявність засобів створення форм, звітів та механізмів автоматизації роботи з інформацією.

6. Розглянуто принципи та концептуальні підходи до проєктування баз даних, зокрема принципи системності, відкритості, сумісності, стандартизації та ефективності. Встановлено доцільність використання інкрементально-ітеративного підходу при створенні інформаційної системи управління збутом продукції.

7. Проведене оцінювання економічної ефективності підтвердило доцільність впровадження автоматизованої системи управління збутом продукції. Використання бази даних дозволяє скоротити витрати часу на обробку інформації, зменшити кількість помилок у документації, покращити контроль за товарними залишками та підвищити оперативність формування звітності.

Практичне значення роботи полягає у можливості використання розроблених підходів і структури бази даних для автоматизації збутової діяльності підприємств малого бізнесу. Реалізація системи сприяє підвищенню ефективності управління інформаційними ресурсами та створює основу для подальшого розвитку інформаційної інфраструктури підприємства. розроблені теоретичні та практичні положення підтверджують ефективність використання SQL-запитів і систем управління базами даних для автоматизації збутової діяльності малого підприємства.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Система. Словник іншомовних слів. URL: <https://www.jnsm.com.ua/cgi-bin/u/book/sis.pl?Qry=%D1%E8%F1%F2%E5%EC%E0> (дата звернення: 26.04.2026).
2. ДСТУ 2481-94. Системи оброблення інформації. Інтелектуальні інформаційні технології. Терміни та визначення. Чинний від 1995-01-01. Вид. офіц. Київ, 1994.
3. Загальні відомості про системи управління базами даних (субд). *StudFiles*. URL: <https://studfile.net/preview/9410526/> (дата звернення: 19.05.2026).
4. СУБД і основні вимоги до них. URL: <http://um.co.ua/4/4-15/4-159701.html> (дата звернення: 19.05.2026).
5. Microsoft Access. *Microsoft*: офіційний вебсайт. URL: <https://www.microsoft.com/uk-ua/microsoft-365/access> (дата звернення: 19.05.2026).
6. Що таке об'єкт бази даних? - визначення з техопедії. Бази даних 2026. *Icy Science*. URL: <https://uk.theastrologypage.com/database-object> (дата звернення: 19.05.2026).
7. Схеми бази даних: компоненти, типи та проєктування. *FoxmindEd*. URL: <https://foxminded.ua/skhemy-bazy-danyh/> (дата звернення: 19.05.2026)..
8. Реляційні бази даних. *Дія.Освіта – IT-студії*. URL: <https://it-osvita.diia.gov.ua/task/item/c1d56d2b-718f-4490-8498-ee209eac30d8> (дата звернення: 19.05.2026).
9. Реляційна модель даних. URL: https://pidru4niki.com/13560615/bankivska_sprava/relyatsiyna_model_danih (дата звернення: 19.05.2026).
10. Лекція 30. Об'єктно-орієнтовані бази даних. URL: <http://moodle.ipk.kpi.ua/moodle/mod/resource/view.php?id=39256> (дата звернення: 19.05.2026).

11. Об'єктно-реляційна модель даних. *Scribd*. URL: <https://surl.li/mdunnm> (дата звернення: 19.05.2026).
12. Стівен Твен. Підручник NoSQL: що таке, типи баз даних NoSQL і приклад. *GURU99*: бібліотека підручників. URL: <https://www.guru99.com/uk/nosql-tutorial.html> (дата звернення: 19.05.2026).
13. Робота з запитами SQL. URL: <https://www.databasetour.net/ua/documentation/working-with-sql-queries.htm> (дата звернення: 19.05.2026).
14. Fang J., Lychagin D., Carey M. J., Tsotras V. J. A new window Clause for SQL++ . *The VLDB Journal*. 2024. Vol. 33. P. 595–623.
15. Barczak K. The Examination of SQL Queries Efficiency in Chosen IT System . *Journal of Computer Sciences Institute*. 2023. Vol. 28.
16. Tomova M., Hofmann M., Hütterer C., Mäder P. *Assessing the utility of text-to-SQL approaches for satisfying software developer information needs* // *Empirical Software Engineering*. 2024. Vol. 29.
17. Ямкова О. М. Зміст формування збутової політики виробничих підприємств. *Економіка і управління*. 2012. № 2. С. 52-59.
18. Радіонова Н. Й. , Терещук В. В. Розрахункові операції за допомогою дистанційних банківських послуг: переваги та недоліки. *Ефективна економіка*. № 2. 2016. 8 с. URL: <http://www.economy.nayka.com.ua/?op=1&z=4784> (дата звернення: 19.05.2026).
19. Терент'єва Н. В. Управління збутовою діяльністю в системі управління підприємством. *Ефективна економіка*. 2016. №2. URL: <http://www.economy.nayka.com.ua/?op=1&z=4783> (дата звернення: 19.05.2026).
20. Основні відомості про бази даних. URL: <https://surl.li/cqioqn>
21. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій). [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського; уклад.: О. С. Коваленко, Л. М. Добровська. Київ: КПІ ім. Ігоря Сікорського, 2020. 192с.

URL: https://ela.kpi.ua/bitstream/123456789/33651/1/PIS_KL.pdf (дата звернення: 19.05.2026).

22. Марченко А.В. Проектування інформаційних систем: курс лекцій. URL: http://kist.ntu.edu.ua/textPhD/PIS_Marchenko.pdf (дата звернення: 19.05.2026).

23. Лавріщева К. М. Програмна інженерія: підручник. К.: НАН України. 2008. 319 с.

24. Моделі життєвого циклу, принципи і методології розробки програмного забезпечення (ПЗ): стаття. URL: <https://evergreens.com.ua/ua/articles/software-development-metodologies.html> (дата звернення: 19.05.2026).

25. Бази даних. ER-модель. Поняття зв'язку. Потужність зв'язку. Типи зв'язків. Приклади. URL: <https://www.bestprog.net/uk/2019/01/27/er-model-the-concept-of-relationship-the-relationship-capacity-types-of-relationships-examples-ua/> (дата звернення: 19.05.2026).

26. Кашаєв Є. Роль мови SQL у створенні структурованих запитів баз даних. *Сучасні інформаційні технології та інноваційні методики в економіці, менеджменті та бізнесі*: Матеріали XXI щорічної студ. наукової конференції, ПДАУ, м. Полтава, 22 квітня 2026 р. Полтава: ПДАУ, 2026 р. С. 86-88.

27. Технологія проектування та адміністрування баз даних і систем даних: метод. рекомендації для практичних робіт. /Уклад. Ю. В. Волосюк. Миколаїв: МНАУ, 2017. 72 с.

28. Методичні вказівки до виконання лабораторної роботи з дисципліни «Організація баз даних» для студентів спец. 123 «Комп'ютерна інженерія» заочної форми навчання /Уклад. Т.О. Паромова. Запоріжжя: НУЗП, 2020. 22 с.

29. Цапук О. Ю. Система критеріїв та показників оцінювання ефективності впровадження інноваційних розробок. *Маркетинг і менеджмент інновацій*. 2013. №3. С.137-145. URL: https://mmi.fem.sumdu.edu.ua/sites/default/files/mmi2013_3_137_145.pdf (дата звернення: 19.05.2026).

30. Терещенко Л. О. Оцінка економічного ефекту від впровадження управлінських інформаційних систем: загальні принципи оцінки. *Економіка та суспільство*. 2021. Вип. №29. С. 1-7. DOI: <https://doi.org/10.32782/2524-0072/2021-29-44>.

31. Інженер-програміст: середня зарплата в Україні. URL: <https://www.work.ua/> (дата звернення: 19.05.2026).