**ДНІпровський національний університет   
імені Олеся Гончара  
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ  
КАФЕДРА КОМП’ЮТЕРНИХ ТЕХНОЛОГІЙ**

**КУРСОВА РОБОТА   
З ДИСЦИПЛІН ПРОФЕСІНОЇ ПІДГОТОВКИ**

на тему: Проектування та обробка бази даних «Салон краси» засобами Microsoft Acccess

Освітньо–професійна програма

Комп’ютерне моделювання та технології програмування

Спеціальність 113 Прикладна математика

Галузь знань 11 Математика і статистика

Рівень вищої освіти перший (бакалаврський)

Студентки 4 курсу групи ПА-20-1з

\_\_\_Мовсісян Л.Р.\_\_\_\_ (прізвище та ініціали)

Керівник - доцент кафедри КТ,   
канд. фіз.-мат. наук Сафронова І.А.

Кількість балів \_\_\_\_\_\_\_\_\_\_\_

Національна шкала \_\_\_\_\_\_\_

Члени комісії:

Степанова Н.І.

(підпис) (прізвище та ініціали)

Дзюба П.А.

(підпис) (прізвище та ініціали)

Зайцев В.Г.

(підпис) (прізвище та ініціали)

м. Дніпро, 2023 р.

Зміст

[ВСТУП 3](#_Toc152083734)

[ПОСТАНОВКА ЗАДАЧІ 5](#_Toc152083735)

[1. Бази даних 6](#_Toc152083736)

[1.1. Реляційні бази даних 7](#_Toc152083737)

[1.2. Модель сутність-звязок 7](#_Toc152083738)

[1.3. Архітектура клієнт-сервер 7](#_Toc152083739)

[1.4. SQL 7](#_Toc152083740)

[2. Системи керува ння базами даних. MS Access 8](#_Toc152083741)

[2.1. Microsoft Access 8](#_Toc152083742)

[3. Створення бази даних для салону краси 9](#_Toc152083743)

[4.Аналіз отриманих результатів реалізації бази даних 14](#_Toc152083744)

[ВИСНОВКИ 15](#_Toc152083745)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 16](#_Toc152083746)

# ВСТУП

*Актуальність*. Проектування та обробка баз даних є важливою темою в галузі інформаційних технологій. Багато ключових аспектів демонструють актуальність цієї теми. Завдяки зростанню обсягів даних, технології обробки даних стають все важливішими. Багато підприємств та організацій вживають різноманітні заходи для збереження, оновлення та оптимізації своїх баз даних. Зі зростанням кількості підключених пристроїв IoT (або англійською Internet of Things-інтернет речей) важливість ефективного зберігання та обробки великої кількості даних збільшується. Бази даних відіграють ключову роль у забезпеченні надійного функціонування систем IoT.

Також, разом із сталою тенденцією до використання баз даних спостерігається зростання популярності хмарних сервісів, що визначає нові вимоги до проектування та управління базами даних у хмарних середовищах. Здатність ефективно обробляти та аналізувати дані стає ключовою для застосування штучного інтелекту та аналітичних інструментів в бізнес- та наукових задачах. З поглибленням конфіденційності даних та зростанням загроз кібербезпеки, належне проектування та управління базами даних стають важливими аспектами захисту конфіденційності та цілісності інформації. Запит на розробку мобільних та веб-додатків зростає, і вони часто використовують бази даних для зберігання та обробки даних. Компанії активно використовують дані для прийняття стратегічних рішень. Правильно спроектовані та ефективно управляючі бази даних є основою для успішного аналізу та вивчення даних.

Тому тема проектування та обробки баз даних залишається актуальною і відіграє ключову роль у сучасному світі інформаційних технологій.

*Метою роботи* є створення продуктивної та комфортної бази даних для працівників та клієнтів, забезпечення швидкого та зручного доступу до інформації та її захисту. Дослідження включає в себе розгляд різноманітних типів баз даних, дослідження та аналіз реляційних баз даних, створення бази даних для організації роботи салона краси.

*Об'єктом дослідження* є система баз даних та їх використання в сучасному бізнесі та технологічному середовищі, зокрема в салоні краси. Дослідження включає в себе ретельний аналіз різноманітних типів баз даних, дослідження реляційних баз даних та їх оптимальне використання у сфері послуг салону краси.

*Предмет дослідження*: проектування та розробки бази даних прикладного напрямку засобами Microsoft Access, розробка та впровадження даних для користувачів бази, розробка та вдосконалення структури бази, що забезпечує ефективну взаємодію користувача з інформацією.

*Методами дослідження* є аналіз потреб користувачів, дослідження відповідного ринку, аналіз деяких з існуючих реляційних СУБД, дослідження технологій обробки даних їх можливості та особливості, які впливають на використання їх у різних галузях, безпосередньо створення бази даних для салону красу.

Курсова робота складається з таких частин:

- вступ, в якому обґрунтовується актуальність роботи, визначено цілі проведення наукового дослідження, галузь дослідження та методи дослідження.

- постановка задачі;

- розгляд поняття бази даних

- аналіз реляційних бази даних

- дослідження моделі сутність-звязок

- вивчення архітектутри клієнт-сервер

- огляд можливостей SQL

- аналіз систем керування базами даних. MS Access

- аналіз отриманих результатів реалізації бази даних

- створення бази даних для салону краси

- висновки;

- список використаних джерел

# ПОСТАНОВКА ЗАДАЧІ

Дослідження передбачає створення бази даних для збереження, обробки і виведення даних, інтерфейсу для користувачів, а також забезпечення конфіденційності та безпеки даних.

Необхідно розробити базу даних (БД) для салону краси для збереження інформації про майстрів та співробітників салону, повного переліку послуг, які він надає, контактних даних клієнтів салону; інформацію про замовленні послу­ги та відповідний розклад майстрів, тощо. Також передбачити можливість для БД записувати, редагувати, видаляти всю необхідну інформацію; робити попередні записи клієнтів; зберігати замовлення (тобто надані послуги салону краси) тощо.

База даних має вдовольнити наступним вимогам:

1. Вимоги до функціональних характеристик

* Введення/виведення, модифікація та видалення інформації
* Пошук даних за умовою
* Фільтрування/сортування даних
* Друк вибіркової інформації

2. Вимоги до надійності

* Передбачити контроль інформації що вводиться
* Захист ключових полів від некоректного введення (автоматичне заповнення)
* Цілісність інформації в базі даних

Вхідними даними для рішення поставленої задачі розробки бази даних для салону краси є:

* дані про майстрів та співробітників;
* перелік послуг, які надаються в салоні;
* прейскурант
* дані про клієнтів салону;
* дані про замовлення;
* розклад роботи майстрів.

# 1. Бази даних

Використання баз даних є однією з характерних рис більшості сучасних інформаційних систем. По своїй суті бази даних є тим, навколо чого і будується інформаційна система будь-якого підприємства. Тому теорії створення та практиці використання баз даних приділяється достатня увага протягом періоду функціонування інформаційних систем. База даних (БД) — впорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система керування БД.

Головним завданням БД є гарантоване збереження значних обсягів інформації (т.зв. записи даних) та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин : збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних). За результатами дослідження [2, 3] розглянемо, із чого саме складаються бази даних.

Дані – це інформація, відомості, показники, необхідні для ознайомлення з ким-, чим-небудь, для характеристики когось, чогось або для прийняття певних висновків, рішень. В базах даних важливу роль відіграє інформація - Інформація — абстрактне поняття, що має різні значення залежно від контексту.

Кожна створена база даних має свою предметну область. Предметна область – це необхідний для розробки бази даних об’єкт, який має в собі дані, які будуть зберігатися в базі даних.

Модель даних — абстрактне представлення реального світу, що відображає тільки ті об'єкти, що безпосередньо стосуються програми. Це, як правило, визначає специфічну групу об'єктів, їх атрибутивне значення і відношення між ними.

Відомі два підходи до організації інформаційних масивів: файлова організація та організація у вигляді бази даних. Файлова організація передбачає спеціалізацію та збереження інформації, орієнтованої, як правило, на одну прикладну задачу, та забезпечується прикладним програмістом. Така організація дозволяє досягнути високої швидкості обробки інформації, але характеризується рядом недоліків. Характерна риса файлового підходу - вузька спеціалізація як обробних програм, так і файлів даних, що служить причиною великої надлишковості, тому що ті самі елементи даних зберігаються в різних системах. Оскільки керування здійснюється різними особами (групами осіб), відсутня можливість виявити порушення суперечливості збереженої інформації. Розроблені файли для спеціалізованих прикладних програм не можна використовувати для задоволення запитів користувачів, які перекривають дві і більше області. Крім того, файлова організація даних внаслідок відмінностей структури записів і форматів передання даних не забезпечує виконання багатьох інформаційних запитів навіть у тих випадках, коли всі необхідні елементи даних містяться в наявних файлах. Тому виникає необхідність відокремити дані від їхнього опису, визначити таку організацію збереження даних з обліком існуючих зв'язків між ними, яка б дозволила використовувати ці дані одночасно для багатьох застосувань. Вказані причини обумовили появу баз даних. База даних може бути визначена як структурна сукупність даних, що підтримуються в активному стані та відображає властивості об'єктів зовнішнього (реального) світу. В базі даних містяться не тільки дані, але й описи даних, і тому інформація про форму зберігання вже не схована в сполученні "файл-програма", вона явним чином декларується в базі.

База даних орієнтована на інтегровані запити, а не на одну програму, яку випадку файлового підходу, і використовується для інформаційних потреб багатьох користувачів. В зв'язку з цим бази даних дозволяють в значній мірі скоротити надлишковість інформації. Перехід від структури БД до потрібної структури в програмі користувача відбувається автоматично за допомогою систем управління базами даних (СУБД).

Системи управління базами даних – це програмні засоби, за допомогою яких можна створювати бази даних, заповнювати їх та працювати з ними. У світі існує багато різноманітних систем управління базами даних. Багато з них насправді є не закінченими продуктами, а спеціалізованими мовами програмування, за допомогою яких кожний, хто вивчить дану мову, може сам створювати такі структури, які йому потрібні, і вводити в них необхідні елементи управління. До таких мов відносяться Clipper, Paradox, FoxPro та інші.

## Реляційні бази даних

Існує кілька типів баз даних, і кожен з них має свої особливості та використання в залежності від конкретних вимог та сценаріїв. Одна з них – реляційна. Досить тривалий час основним типом були реляційні бази даних, які на сьогодні вже вважаються класичними. Реляційні бази даних (RDBMS) є одним з найпоширеніших типів баз даних. Вони використовують реляційну модель для організації та управління даними.

У [ 4 ] розглянуті наступні основні поняття реляційної бази даних, які включають:

1. **Таблиці .**

Основна структурна одиниця даних в RDBMS. Таблиці подібні до електронних таблиць і мають рядки та стовпці. Кожен рядок представляє запис, а кожен стовпець - атрибут або поле.

1. **Стовпці .**

Стовпці визначають атрибути даних, які зберігаються в таблиці. Кожен стовпець визначає конкретний тип даних (наприклад, текст, число, дата тощо).

1. **Рядки .**

Рядки в таблиці представляють конкретні записи або кортежі даних. Кожен рядок має унікальний ідентифікатор, який називається ключем.

1. **Ключі .**

Ключі використовуються для однозначної ідентифікації записів в таблиці. Один або кілька стовпців можуть бути обрані як ключі.

1. **Відносини .**

Відносини визначають зв'язки між таблицями. Це може бути один до одного (1:1), один до багатьох (1:N), або багато до багатьох (N:M). Відносини грають важливу роль в підтримці цілісності даних.

1. **SQL (Structured Query Language) .**

SQL є мовою запитань та маніпуляції даними, яка використовується для взаємодії з реляційними базами даних. SQL дозволяє створювати, зчитувати, оновлювати та видаляти дані.

1. **Транзакції .**

Транзакції визначають групу операцій, які виконуються атомарно (або в цілому), тобто або всі операції виконуються успішно, або жодна з них.

1. **Нормалізація .**

Нормалізація - це процес організації даних в базі таким чином, щоб уникнути зайвої дублювання та підтримувати цілісність даних. Нормалізація допомагає уникати аномалій даних та покращує ефективність бази даних.

Реляційні бази даних широко використовуються у бізнесі, науці, освіті та інших галузях для зберігання та управління великим обсягом структурованих даних

## Модель сутність - зв’язок

Як ми вже з’ясували, базою даних є структурована сукупність даних, які відображують стан об’єктів певної предметної області та зв’язки між ними. Розробник бази даних повинен описати певну предметну область, змоделювати її для використання у вигляді бази даних. Така модель називається моделлю сутність-зв'язок.

Ми знаємо, що найбільш поширеною є реляційна модель, в основі якої лежать таблиці з наборами однотипних об’єктів. Тобто передусім, аналізуючи предметну область, потрібно виділити об’єкти і виокремити серед них сутності – множини об’єктів, котрі мають однаковий набір параметрів, суттєвих для бази даних. Так, у школі є різні класи – це об’єкти, котрі належать до сутності Клас. У класі вчаться учні – що належать до сутності Учень.

* Модель сутність-зв'язок (ER-модель) - це методологія визначення та опису структури бази даних, яка використовується для моделювання відносин

між сутностями в системі. ER-модель (от англ. Entity-Relationship model, модель «сущность — связь») дозволяє визначити сутності, їх атрибути та взаємозв'язки між сутностями. Вона є важливим інструментом на етапі проектування бази даних.

У [5], автор зазначає такі основні елементи моделі сутність - зв'язок :

1. **Сутності .**

Сутності представляють об'єкти або концепції, які мають важливий сенс в контексті бізнес-домену. Кожна сутність має унікальний ідентифікатор, який дозволяє її відрізняти від інших.

1. **Зв'язки .**

Зв'язки вказують на взаємозв'язки між сутностями. Зв'язки можуть бути один до одного (1:1), один до багатьох (1:N) або багато до багатьох (N:M). Кожен зв'язок має асоційовану кількість та тип зв'язку.

1. **Атрибути .**

Атрибути описують характеристики сутностей. Вони визначають конкретні властивості або риси, які характеризують сутність.

1. **Ключі .**

Ключі визначають унікальні ідентифікатори сутностей. Ключі можуть бути первинними (визначають унікальність запису в сутності) та зовнішніми (вказують на відносину з іншою сутністю).

1. **Кардинальність .**

Кардинальність визначає кількість екземплярів однієї сутності, які можуть бути пов'язані з екземпляром іншої сутності через зв'язок.

1. **Асоціації .**

Асоціації визначають взаємодію між сутностями та їхніми атрибутами через зв'язки.

Модель сутність-зв'язок допомагає визначити основну структуру бази даних, забезпечуючи чітке розуміння відносин між різними частинами інформації. Це дозволяє покращити ефективність проектування бази даних і сприяє підтримці цілісності та ефективного використання даних в системі.

## Архітектура клієнт-сервер

Сьогодні використовуються фактично дві основні моделі огранізації обчислень та обробки інформації: модель централізованих обчислень та модель розподілених обчислень Модель централізованих обчислень виникла і використовується з самого початку комерційного використання комп'ютерів з 1950-тих років. Цю модель можна уявити собі як аплікацію, що працює на одному (або кількох не з'єднаних між собою) комп'ютерах. Модель розподілених обчислень виникає з широким використанням мережевих з'єднань — тобто у 1980-тих роках. Цю модель можна уявити собі як аплікацію, що працює на кількох з'єднаних між собою комп'ютерах одночасно. Якщо описати обидві моделі з точки зору теорії систем, то різниця між ними є в наступному:

– модель централізованих обчислень має лише один пристрій обробки і один пристрій збереження даних, хоча присторїв вводу-виводу (терміналів) може мати і декілька.

– Модель розподілених обчислень х-зується вже як мінімум двома пристроями обробки і мін. двома пристроями збереження даних.

Однією з найбільш поширених версій моделі розподілених обчислень є архітектура клієнт-сервер

Архітектура клієнт-сервер — це структурний підхід до дизайну та побудови розподілених систем, де функціональність системи розділена між двома основними компонентами: клієнтом і сервером. Кожен з цих компонентів виконує конкретні функції, і вони взаємодіють між собою через мережу.

Основні характеристики архітектури клієнт-сервер (6):

1. **Сервер .**

Це центральна частина системи, яка забезпечує обробку та зберігання даних, виконання операцій та обслуговування запитів від клієнтів.

1. **Клієнт .**

Це складова системи, яка відповідає за інтеракцію з користувачем, введення та відображення даних, а також відправку запитів до сервера.

1. **Комунікація .**

Взаємодія між клієнтом і сервером здійснюється через мережу (наприклад, Інтернет або локальну мережу). Клієнти висилають запити на сервер, а сервери відповідають на ці запити.

1. **Розділення обов'язків .**

Розділення функціональності між клієнтом і сервером. Зазвичай, сервер відповідає за обробку даних та виконання бізнес-логіки, тоді як клієнт — за представлення даних та взаємодію з користувачем.

1. **Синхронна та асинхронна обробка .**

Варіанти взаємодії можуть бути синхронними (чекання на відповідь) або асинхронними (продовження виконання після відправки запиту без очікування відповіді).

1. **Масштабованість .**

Архітектура клієнт-сервер дозволяє легко масштабувати систему шляхом додавання нових клієнтів або серверів.

1. **Надійність .**

Система може бути більш надійною, оскільки збої в одному компоненті (клієнті або сервері) не обов'язково призводять до повного відмови системи.

Архітектура клієнт-сервер є дуже поширеною в інформаційних технологіях і використовується в багатьох різних областях, таких як веб-розробка, корпоративні системи, мобільні додатки тощо.

## Структурована мова запитів (SQL)

Мова SQL (Structured Query Language – структурована мова запитів) в даний час є промисловим стандартом, який більшою чи меншою мірою підтримує будь-яка СУБД, яка претендує на звання «реляційної». У той же час SQL піддається суворій критиці за недостатню відповідність реляційним принципам.

На початку 70-х років у компанії IBM була розроблена експериментальна СУБД System R на основі мови SEQUEL (Structured English Qeury Language – структурована англійська мова запитів), яку можна вважати безпосереднім попередником SQL. Метою розробки було створення простої непроцедурної мови, якою міг скористатися будь-який користувач, що навіть не має навичок програмування. У 1981 році IBM оголосила про свій перший, заснований на SQL програмному продукті, SQL/DS. Трохи згодом до неї приєдналися Oracle та інші виробники. Перший стандарт мови SQL був прийнятий Американським національним інститутом стандартизації (ANSI) у 1987 (так званий SQL level /рівень/ 1) та дещо уточнений у 1989 році (SQL level 2). Подальший розвиток мови постачальниками СУБД вимагав прийняття в 1992 нового розширеного стандарту (ANSI SQL-92 або просто SQL-2). В даний час проводиться робота з підготовки третього стандарту SQL, який повинен включати елементи об'єкто-орієнтованого доступу до даних. Необхідно сказати, що хоча SQL і замислювався як засіб роботи кінцевого користувача, зрештою він став настільки складним, що перетворився на інструмент програміста.

У SQL визначено два підмножини мови:

• SQL-DDL (Data Definition Language) – мова визначення структур та обмежень цілісності баз даних. Сюди відносяться команди створення та видалення баз даних; створення, зміни та видалення таблиць; керування користувачами і т.д.

• SQL-DML (Data Manipulation Language) - мова маніпулювання даними: додавання, зміна, видалення та вилучення даних, управління транзакціями.

SQL є стандартизованою мовою, і різні системи управління базами даних (наприклад, MySQL, PostgreSQL, Microsoft SQL Server, Oracle) використовують SQL для взаємодії з даними. SQL є потужним інструментом для роботи з базами даних та забезпечує зручний і ефективний спосіб управління даними в реляційних структурах.

# 2. Системи керування базами даних. MS Access

## 2.1. Microsoft Access

Наразі існує багато альтернативних платформ, але я для проектування та створення своєї бази даних обрала Microsoft Access . В університеті є ліцензія на використання продукції MS365, відповідно завдання в університеті ми виконуємо на цій платформі. Для того щоб, за власний рахунок не організовувати ліцензію на LibreOffice Base , Kexi або NuBuilder Forte , я обрала Microsoft Access , тому що у них нього ,більшість функцій свбодному ,в той час як на інших лише частина функцій безкоштовні, решта – платні.

Microsoft Access - це програма для створення та управління реляційними базами даних. Вона є частиною пакету офісних програм Microsoft Office. Access дозволяє користувачам легко створювати бази даних, розробляти форми та звіти, виконувати запити та автоматизувати деякі завдання за допомогою макросів.

Як вказано у [10, 11, 12 ] , Microsoft Access включає в себе настпуні основні функції та характеристики:

Створення таблиць: Користувачі можуть створювати таблиці для зберігання даних. Access надає редактор таблиць для визначення полів, типів даних та відносин між таблицями.

Форми і звіти: Використовуючи Access, можна легко створювати користувацькі форми для введення та відображення даних, а також звіти для виведення інформації в зручному форматі.

Запити: Access дозволяє користувачам створювати SQL-подібні запити для вибірки, оновлення, вставки та видалення даних з таблиць.

Макроси та вбудований код VBA: Для автоматизації завдань та додаткової функціональності можна використовувати макроси.

Інтеграція з іншими продуктами Microsoft: Access легко інтегрується з іншими програмами Microsoft Office, такими як Excel, Word та Outlook.

Безпека та управління правами доступу: Access надає можливості управління правами доступу до бази даних, включаючи визначення рівнів доступу для користувачів та груп.

Інтеграція з іншими джерелами даних: Access може підключатися до інших джерел даних, таких як таблиці SQL Server, Excel або інші бази даних, за допомогою зовнішніх посилань.

Розглянемо, переваги та недоліки Microsoft Access:

**Переваги Microsoft Access:**

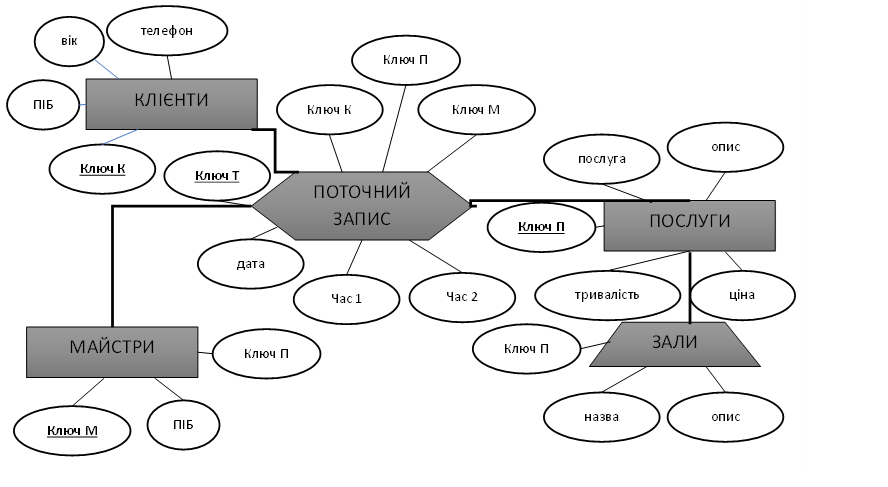
* Зручний інтерфейс: Microsoft Access має інтуїтивно зрозумілий інтерфейс, що робить його доступним для користувачів з різним рівнем досвіду у роботі з базами даних.
* Швидкий старт: Завдяки шаблонам та інструментам для швидкого створення баз даних, Access дозволяє швидко розпочати роботу над проектом.
* Інтеграція з іншими продуктами Microsoft: Access добре інтегрований з іншими продуктами Microsoft Office, такими як Excel, Word та Outlook.
* Можливості розробки без програмування: Користувачі можуть створювати бази даних, форми та звіти без необхідності програмування, використовуючи лише інтерфейс.
* Макроси та VBA: Для більш складних завдань ви можете використовувати макроси або VBA для автоматизації рутинних задач та розширення функціональності.

**Недоліки Microsoft Access:**

* Обмеження щодо масштабування: Access не є ідеальним вибором для великих підприємств або проектів з великим обсягом даних, оскільки він може швидко стати неефективним у великій базі даних.
* Необхідність в Microsoft Office: Для використання Access потрібно мати установлений пакет офісних програм Microsoft Office, що може бути вартісним.
* Орієнтованість на одного користувача: Access зазвичай використовується для проектів з одним користувачем, і може виникнути проблеми при спробі одночасної роботи більше ніж одного користувача.
* Обмежені можливості для веб-застосунків: Хоча Access має можливості для створення веб-застосунків, це не його сильна сторона, і інші платформи можуть бути більш підходящими для цього.
* Безпека: У порівнянні з іншими СУБД, Access може мати менші можливості в області безпеки та керування доступом.

# 3. Створення бази даних для салону краси

Розробку бази даних я почала з побудови інфологічної моделі бази даних для салону краси, де має зберігатися інформація про співробітників салону, про послуги і процедури, які він надає та їх типи, про клієнтів салону і їх поточні записи, і т.ін. Інформація може бути використана у менеджером та адміністратором невеликого салону краси, його співробітниками, а за потреби і надана відвідувачам.

  
рис. 1.1 – Інфологічна модель бази даних для Салону краси

Створення моделі включає в себе визначення об'єктів та їх характеристик (атрибутів), які будуть відстежуватися та зберігатися в базі даних. Першим чином визначимо сутності і їх атрибути.

Нижче наведено кілька основних сутностей та атрибутів для проектування бази даних салону краси:

**Сутності**:

Клієнти

Майстри

Послуги

Поточний запис

***Атрибути***:

**Клієнти (сутність)**

*Атрібути:* *Код клієнта-Прізвище-Ім'я-По-батькові-Телефон-Вік*

**Майстри: (сутність)**

*Атрібути: Код майстра-Прізвище-Ім'я-Дата народження-Тип послуги*

**Послуги: (сутність)**

*Атрібути: Код послуги-Найменування-Вартість-Номер Залу-Тип послуги*

**Поточний запис: (сутність)**

*Атрібути: Дата запису-Код клієнту-Код майстра-Код послуги-Час*

Вихідними даними є перегляд послуг, які надаються в салоні краси відповідно заданому критерію, можливість оформлення послуги, пошук потрібної інформації.

Після виділення сутностей і атрибутів я зробила нормалізацію для таблиці “Інформація щодо записів”.

Cутність :“Інформація щодо записів”

Атрибути:

Найменування;

- Назва;

- Ім’я;

- Телефон;

- Прізвище;

- Дата запису;

- Майстри прізвище;

- Час.

Для визначення первиного ключа таблиці "Інформація щодо записів" я обрала той набір атрибутів, який унікально ідентифікує кожен запис у таблиці. З урахуванням наданих атрибутів, можливі кілька варіантів.

Варіант 1: Назва та Дата запису

Можливий первинний ключ:

Назва

Дата запису

Цей варіант передбачає, що комбінація назви та дати запису є унікальною для кожного запису.

Варіант 2: Телефон та Дата запису

Можливий первинний ключ:

Телефон

Дата запису

Цей варіант передбачає, що комбінація телефону та дати запису є унікальною для кожного запису.

Варіант 3: Найменування, Прізвище, та Дата запису

Можливий первинний ключ:

Найменування

Прізвище

Дата запису

Цей варіант передбачає, що комбінація найменування, прізвища та дати запису є унікальною для кожного запису.

Вибір конкретного варіанту залежить від того, як саме ви хочете ідентифікувати кожен запис в таблиці та від конкретних умов використання вашої бази даних.

Усі інші таблиці вже є нормалізованими.

Всі дані розбито на таблиці “Клієнти”, “Співробітники”, «Послуги», «Поточний запис», «Інформація щодо записів», «Тип послуги» та «Тип залу». Перелік інформації, що вводиться:

Про клієнтів (рис 1.2) :

- Код клієнта;

- Назва або ПІБ клієнта;

- Телефон.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, число, линия

Автоматически созданное описание рис. 1.2 – Таблиця “ Клієнти ”

Про співробітників (рис. 1.3) :

- Код співробітника;

- Прізвище;

- Ім'я;

- Тип наданої послуги;

- Дата народження.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

рис. 1.3 - Таблиця “ Майстри”

Про послуги (1.4) :

- Код послуги;

- Тип послуги;

- Найменування послуги;

- Вартість послуги;

- Номер залу.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание Изображение выглядит как текст, число, Шрифт, программное обеспечение

Автоматически созданное описание

рис. 1.4 – Таблиця “ Послуги ”

Інформація про отримані послуги/замовлення ( рис. 1.5 ) :

- Код послуги;

- Код співробітника;

- Код клієнта;

- Дата;

- Час витрачений на послугу.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

рис. 1.5 – Таблиця “ Поточний запис ”

Інформація щодо записів (1.6) :

- Найменування;

- Назва;

- Ім’я;

- Телефон;

- Прізвище;

- Дата запису;

- Майстри прізвище;

- Час.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, число, снимок экрана

Автоматически созданное описание

рис. 1.6 – Таблиця “ Інформація щодо записів”

Тип залу ( 1.7) :

- Код залу;

- Назва.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

рис. 1.7 – Таблиця “ Тип залу”

Тип послуги (1.8 ) :

* Код типу послуги;
* Назва;
* Опис.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

рис. 1.8 – Таблиця “ Тип послуги ”

Другим етапом створення бази даних був її аналіз і встановлення зв'язків.

( рис. 1.9 )

Зв'язки:

Один клієнт може мати багато записів на прийом (один до багатьох).

Один співробітник може обслуговувати багато записів приймання (один до багатьох).

Один запис на прийом пов'язаний з одним клієнтом, одним співробітником та однією послугою.

Изображение выглядит как текст, снимок экрана, диаграмма, Параллельный

Автоматически созданное описание

рис. 1.9 – Схема зав’язків даних

На наступному етапі, я представила базу на мові інфологічного моделювання, вона має такий вигляд:

Клієнти (Кл, Код клієнта-Прізвище-Ім'я-По-батькові-Телефон-Вік)

Майстри (Мс, Код майстра-Прізвище-Ім'я-Дата народження-Тип послуги)

Послуги (Пос, Код послуги-Найменування-Вартість-Номер Залу-Тип послуги) [Поточний запис]

Поточний запис [Клієнт М, Майстри 1, послуги 1 ] (Мс,Кл, Дата запису-Код клієнту-Код майстра-Код послуги-Час)

Послуги [Тип залу 1, Тип послуги 1 ] ( Пос,Пот.зап., Код послуги-Найменування-Вартість-Номер Залу-Тип послуги)

Тип залу (Пос,Код залу, Назва) { послуги}

Тип послуги (Пос, код типу послуги Код, Назва) { послуги}

В цих моделях Клієнт, Поточний запис і Послуги – найменування, а Кл, Пот.зап. и Пос. – коди кліентів, поточних записів і послуг, які надає цей салон.

На цьому етапі я сформулювала і відтворила діаграмму ER-моделювання (рис. 1.9 ) :

Після створення і заповнення таблиць у середовищі Microsoft Access я створила декілька SQL запитів для більш ефективної комунікаціїї між співробітниками і відвідувачами салону краси.

Перший запит (рис. 1.10 ) дає змогу вивести і переглянути вік усіх майстів, що працюють в салоні.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

рис. 1.10 – запит “Вік майстрів”

Інтерпретація у SQL:

SELECT Майстри.Прізвище, Майстри.[дата народження], Int((Date()-Майстридата народження])/365) AS Возраст

FROM Майстри

Другий запит ( рис. 1.11 ) дає змогу зробити його за наступними параметрами:

* Ціна послцги;
* Максимальну ціну послуги;
* Код типу послуги.

![Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

рис. 1.11 – запит “Запит за ціною послуги”

Інтерпретація у SQL:

SELECT Послуги.Найменування, Послуги.Вартість, [Тип послуги].Назва, [Тип залу].Назва

FROM [Тип послуги] INNER JOIN ([Тип залу] INNER JOIN Послуги ON [Тип залу].[Код залу] = Послуги.[Номер Залу]) ON [Тип послуги].[Код типа послуги] = Послуги.[Тип послуги]

WHERE ((([Введіть вашу ціну за послугу])<Послуги!Вартість) And (([Введіть вашу мах ціну за послуги])>Послуги!Вартість));

Третій запит ( рис. 1.12 ) дає змогу вивести вартість послуги за наступними параметрами:

* Назва послуги.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

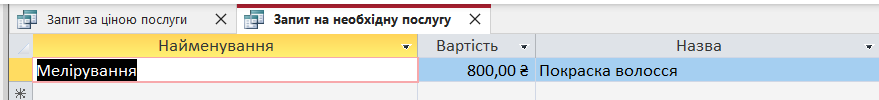


рис. 1.12 – запит “Запит за необхідну послугу”

Інтерпретація у SQL:

SELECT Послуги.Найменування, Послуги.Вартість, [Тип послуги].Назва

FROM [Тип послуги] INNER JOIN Послуги ON [Тип послуги].[Код типу послуги] = Послуги.[Тип послуги]

WHERE (((Послуги.Найменування)=[Введіть назву потрібної послуги]));

Четвертий запит ( рис. 1.13 ) дає змогу вивести кошти витрачені кожним клієнтом в сумі за усі придбані послуги у салоні краси:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

рис. 1.13 - запит “Запит на сумарну вартість з одного клієнта”

Інтерпретація у SQL:

SELECT Клієнти.Прізвище, Sum(Послуги.Вартість) AS [Sum-Вартість]

FROM Послуги INNER JOIN (Клієнти INNER JOIN [Поточний запис] ON Клієнти.[Код клієнта] = [Поточний запис].[Код клієнту]) ON Послуги.[Код послуги] = [Поточний запис].[Код послуги]

GROUP BY Клієнти.Прізвище;

# 4.Аналіз отриманих результатів реалізації бази даних

Отже, я досягла поставленої мети дослідження . Мною була спроектована і створена база даних у просторі Microsoft Access, з урахуванням вимог організації та користувачів.

Також, я розробила структуру бази даних простору, включаючи ієрархію таблиць та ключів, базу даних та форми для введення даних. У останньому етапі я реалізувала простір з інфоррмацією засобами Microsoft Access, використовуючи стандартні компоненти та налаштовуючи їх відповідно до вимог проекту.

У подальшому можна вдосконалити інтерфейс для користувачів, додати більше сутностей, функціоналу, форм, додати розрахунки, що передбачали систему знижок і клубні карти клієнтів.

Можна покращити безпеку даних і конфіденційнсть через ведення двухфакторної аутентификації.

Для більшої зручності співвробітників або відвідувачів салону краси, можна створити більшу кількість форм, де вони зможуть побачити перзоналізовані детальну інформацію щодо записів і наданих послуг салоном краси. У форми можна додати доступ до розкладу усіх майстрів, або відповідних послуг ,що надає салон. Створити додаткові форми для взаємодії відвідувачів і співпрацівників салону краси.

Крім того, можна забезпечити можливість подання заявок на відвідування відповідних послуг, а також організацію діяльності в салоні краси, зокрема проведення спільних процедур та послуг універсального типу.

# ВИСНОВКИ

У роботі були детально розглянуті та проаналізовані системи баз даних та досліджено технологіх обробки даних з точки зору зручності та практичної доцільності їх використання для створення бази даних в середовищі від Microsoft 365.

Засобами Microsoft Access розроблено базу даних для співробітників, яка надає їм можливість безпечного та зручного доступу до усіх поточних записів відвідувачів і інформаціїї про них , що є в доступі у салона краси та у майбутньому зможе забезпечити ефективну комунікацію і співпрацю між майстрами і клієнтами закладу.

Було створено зручну та інтуїтивно зрозумілу базу, яка дозволяє легко та швидко знаходити необхідну інформацію. Крім того, були розроблено різноманітні запити і форми, які дозволяють ефективно взаємодіяти з базою та забезпечувати високу продуктивність роботи майстрів.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке база даних URL: <https://hostiq.ua/wiki/ukr/database/> (дата звернення: 21.11.2023).
2. Моделі даних та їх класифікація. URL: <https://wiki.cuspu.edu.ua/index.php/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%96_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85_%D1%82%D0%B0_%D1%97%D1%85_%D0%BA%D0%BB%D0%B0%D1%81%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F>.

(дата звернення: 21.11.2023).

1. Види інформаційних масивів URL: <https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/260038/index.html> (дата звернення: 21.11.2023).
2. Основні поняття реляційних БД: нормалізація, зв’язок та ключі URL: <https://bondarenko.dn.ua/osnovni-ponyattya-relyatsijnih-bd-normalizatsiya-zv-yazok-ta-klyuchi/> (дата звернення: 21.11.2023).
3. Короткий Курс по Теорії БД (1)Глава 2-4 (дата звернення: 20.11.2023).
4. Сімонович С. В. «Базовий курс», с.13-40 (дата звернення: 20.11.2023).
5. Симонович С. В., Євсєєв Г. А., Олексїїв А. Г. Спеціальна інформатика: Навчальний посібник. - М.: АСТ-ПРЕС: Інфорком-Прес, 2019 (дата звернення: 21.11.2023).
6. Гончаров А. Ю. Access 2020. Самовчитель с прикладами, Київ, 2017 р. с.23-40 (дата звернення: 20.11.2023).
7. Основні оператори мови SQL URL: <https://allreferat.com.ua/uk/informatuka_kompyuterni_nayku/referat/6286> (дата звернення: 22.11.2023)
8. Довідка та навчання по Access URL: <https://support.microsoft.com/ru-ru/access> (дата звернення: 21.11.2023).
9. «Практикум за комп'ютерною технологією» Єфімова, Моїсеєва (дата звернення: 21.11.2023).
10. Бакаревич Ю.Б., Ярмолюк Н.В. Самовчитель Microsoft Access 2012. – Харків, 2002 (дата звернення: 21.11.2023).