

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА
ФРАНКА

ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ

Кафедра Цифрової економіки та бізнес-аналітики

КУРСОВА РОБОТА

з навчальної дисципліни

Проектування та адміністрування БД та СД

на тему:

Інформаційна система для салону краси

спеціальність: _____ 051 «Економіка» _____

спеціалізація: _____ «Інформаційні технології в бізнесі» _____

освітній ступінь: _____ бакалавр _____

Науковий керівник:

Доцент Депутат Б. Я.

_____ “___” _____ 20__ р.

Виконавець:

студентка групи УФЕ-31с

Бабій А. М.

_____ “___” _____ 20__ р.

Загальна кількість балів _____

ЛЬВІВ 2022

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. АНАЛІЗ ВИМОГ	5
1.1 Постановка завдання.....	5
1.2 Use Case Diagram	6
РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ	9
2.1 Опис моделі даних.....	9
2.3 Визначення типів даних	14
2.4 Обмеження цілісності даних	16
2.5 Реалізація SQL-скрипту.....	18
ВИСНОВКИ	22
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	23
ДОДАТКИ	25

ВСТУП

Актуальність теми дослідження. Технології впливають майже на всі аспекти життя 21-го століття, від ефективності та безпеки транспорту до доступу до їжі та охорони здоров'я, соціалізації та продуктивності. Потужність Інтернету дозволила глобальним спільнотам формуватися, а ідеями та ресурсами стало легше ділитися. Неабияку роль в процесі поширення ресурсами відводиться інформації. Звичайно, інформацією володіє кожен користувач інтернету, починаючи від студента та закінчуючи відвідувачем ресторану, котрий поцікавився меню онлайн. В такому випадку варто відвести важливе місце інформаційним системам. Інформаційна система — інтегрований набір компонентів для збору, зберігання та обробки даних, а також для надання інформації, знань та цифрових продуктів. Бізнес-фірми та інші організації покладаються на інформаційні системи для здійснення та управління своїми операціями, взаємодії зі своїми клієнтами та постачальниками та для конкуренції на ринку. Інформаційні системи використовуються для запуску міжорганізаційних ланцюгів поставок та електронних ринків. Завдяки інформаційній системі в наш час стало можливим скористатись спектром різних послуг, а саме: бронювання номера в готелі, заповнення страхових бланків, купівля квитків в кінотеатр та ін. У випадку салону краси, користувач має доступ до широкого асортименту можливих косметичних послуг, вибору майстра та можливість обрати зручний для нього час. Така інформаційна система набирає обертів на просторах інтернету та користується великою популярністю, чому сприяє відкриття все новіших салонів та бажанням людей ставати кращими версіями самих себе. Тому варто сказати, що така система є актуальною та її розробка допоможе відвідувачам скористатись послугами салону.

Мета та завдання дослідження. Метою даної курсової роботи є опис та вивчення теоретичних основ баз даних та реалізація інформаційної системи салону краси.

Для того, щоб досягти мету даної роботи, варто окреслити такі завдання:

1. Проаналізувати структуру інформаційної системи для салону краси;
2. Розробити Use Case діаграму для кращого розуміння бізнес-моделі даної інформаційної системи;
3. Створити ER-діаграму, окреслити її сутності та зв'язки;
4. Спроекувати та описати структуру БД.

Об'єкт дослідження. Особливість розробки інформаційної системи для салону краси за допомогою системи управління базами даних MsSQL.

Предмет дослідження. Застосування різних технологій та ресурсів задля розробки структурованої бази даних націленої для перспективного використання веб-сайту салону краси.

Структура роботи. Курсова робота складається з вступу, двох розділів — «Аналіз вимог» та «Розробка бази даних» та підрозділів, а також висновків, списку використаних джерел та додатків.

Загальний обсяг роботи — 37 сторінок.

РОЗДІЛ 1. АНАЛІЗ ВИМОГ

1.1 Постановка завдання

Неможливо недооцінити важливості вивчення та використання інформаційних систем. Варто сказати, що автоматизована ІС — сукупність інформації, різних методів і моделей, апаратних, програмних, організаційних, технологічних засобів і відповідних фахівців. Отже, інформаційна система — це організаційно впорядкована сукупність фахівців, інформаційних ресурсів та інформаційних технологій, зокрема з використанням засобів обчислювальної техніки і зв'язку, що реалізують такі інформаційні процеси, як отримання вхідних даних; обробка цих даних і/або зміна власного внутрішнього стану (внутрішніх зв'язків/відносин), видача результату або зміна свого зовнішнього стану (зовнішніх зв'язків/відносин). За допомогою ІС надається можливість встановлення зв'язку між усіма елементами бізнес-процесів підприємства, що покращує можливості планування, контролю й регулювання процесів. Інформаційна система, при формуванні якої використано принцип зворотного зв'язку на всіх рівнях управління, забезпечує зв'язок між елементами системи управління й елементами бізнес-процесів, тобто між усіма етапами прийняття рішень, а також надає можливість накопичення даних, аналізу і моделювання.

Мета даної роботи полягає у створенні бази даних для підтримання роботи салону краси. Основна задача полягає в створенні основної бази клієнтів, автоматизації запису на процедури. Використання продукту проекту допоможе салону знайти нову базу клієнтів. В перспективі, маючи готовий та під'єднаний веб-сайт, кожен користувач зможе стати клієнтом салону всього за декілька кліків. За рахунок цього, прибуток може значно вирости. Задля досягнення основної мети були визначені такі задачі: розробка бази, використання селективів, розробка ER-діаграми та Use Case діаграми.

1.2 Use Case Diagram

Діаграма варіантів використання є основною формою вимог до системи/програмного забезпечення для нової недостатньо розробленої програми. Випадки використання визначають очікувану поведінку (що), а не точний спосіб її здійснення (як). Варіанти використання після вказівки можуть бути позначені як текстовим, так і візуальним представленням (тобто діаграмою варіантів використання). Ключова концепція моделювання варіантів використання полягає в тому, що воно допомагає нам проектувати систему з точки зору кінцевого користувача. Це ефективна техніка для передачі поведінки системи в термінах користувача шляхом визначення всієї зовнішньої поведінки системи. Діаграми типу use case зазвичай розробляються на ранній стадії розробки, і люди часто застосовують моделювання варіантів використання для наступних цілей:

- Вказати контекст системи
- Врахувати вимоги системи
- Перевірити архітектуру системи
- Підтримувати впровадження та створювати тестові випадки

Випадки використання мають різні види відносин. Визначення зв'язку між двома варіантами використання є рішенням аналітиків програмного забезпечення діаграми варіантів використання. Відносини між двома варіантами використання в основному моделюють залежність між двома варіантами використання. Повторне використання існуючого варіанту використання за допомогою різних типів зв'язків зменшує загальні зусилля, необхідні для розробки системи. Існують такі типи зв'язків:

1. <<extend>> — варіант використання розширення складається з однієї або кількох послідовностей поведінки (сегментів), які описують додаткову поведінку, яка може поступово посилювати поведінку

базового варіанту використання. Кожен сегмент можна вставити в базовий варіант використання в іншій точці, яка називається точкою розширення;

2. <<include>> — варіант використання включає функціональні можливості, описані в іншому варіанті використання, як частину потоку його бізнес-процесу;
3. Узагальнення — узагальнення використовується, коли ви знайдете два або більше варіантів використання, які мають спільні риси поведінки, структури та призначення. Коли це станеться, ви можете описати спільні частини в новому, часто абстрактному варіанті використання, який потім спеціалізується на дочірніх варіантах використання.

У випадку моєї інформаційної системи салону краси був використаний другий тип зв'язку — include (рис. 1.1)

В даній діаграмі окреслено варіанти використання для трьох «акторів» — клієнта, працівника та адміністратора. Для клієнта створено такі use cases:

1. Перегляд послуг;
2. Створення запису;
3. Вибір бажаної послуги, вільного майстра та конкретної години;
4. Введення ім'я, прізвища, номера телефону та електронної адреси;
5. Перегляд деталей запису.

Для адміністратора визначено такі use cases:

1. Управління персоналом;
2. Редагування записів;
3. Видалення запису;
4. Додавання запису;
5. Редагування послуг;
6. Перегляд деталей запису;
7. Введення даних майстрів;

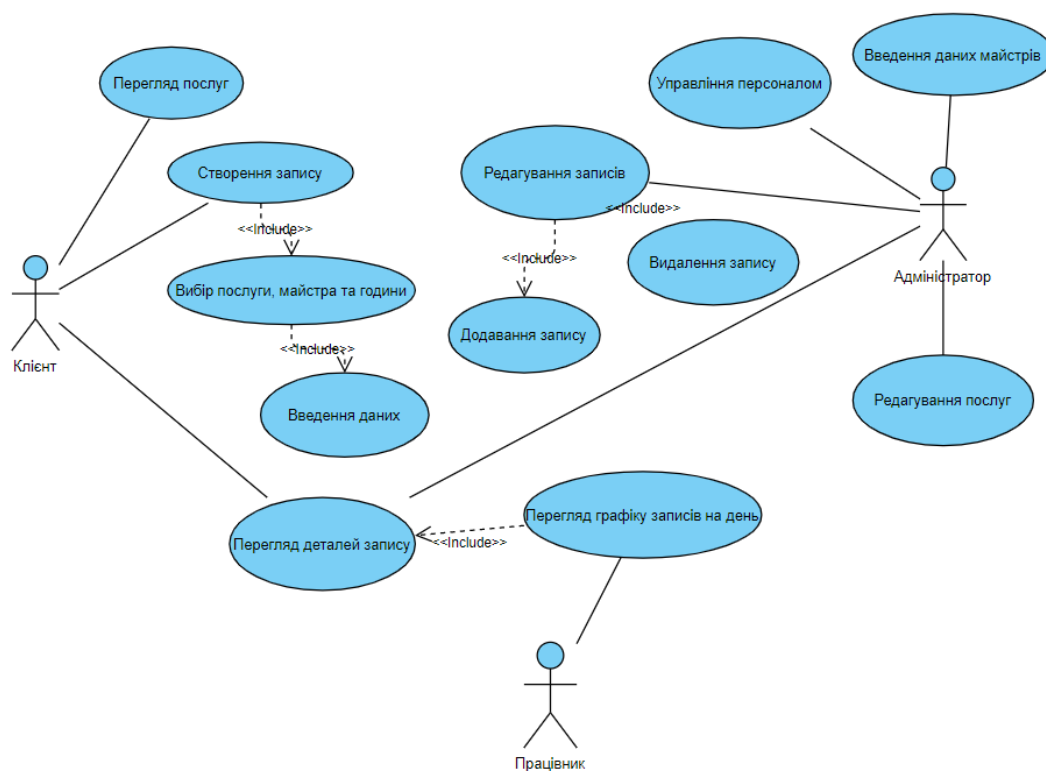


Рис. 1.1 Діаграма варіантів використання

Для працівника створено такі use cases:

1. Переглянути графік записів на день;
2. Перегляд деталей запису.

Кожен з цих кейсів детально проаналізовано та використано в бізнес-моделі моєї інформаційної системи для подальшої розробки бази даних.

РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ

2.1 Опис моделі даних

SQL, Structured Query Language, — це мова програмування, призначена для керування даними, що зберігаються в реляційних базах даних. SQL працює за допомогою простих декларативних операторів. Це забезпечує точність і безпеку даних, а також допомагає підтримувати цілісність баз даних незалежно від їх розміру.

Мова SQL сьогодні широко використовується у веб-фреймворках і додатках баз даних. Знання SQL дає вам свободу вивчення ваших даних і можливість приймати кращі рішення. Вивчаючи SQL, ви також дізнаєтесь концепції, які застосовуються майже до кожної системи зберігання даних.

Завдяки вивченню баз даних для подальшого оволодіння теоріями та методами систем баз даних, розвитку та вдосконалення вміння розробляти інформаційні системи управління з'являється можливість закласти хорошу основу для майбутнього розвитку інформаційних систем. Вивчення бази даних в основному включає в себе аналіз вимог, аналіз даних, розробку функції методу проектування логіки системи, реалізацію системи методу проектування фізичної моделі та інші методи, а потім проектування системи керування салоном краси, яка може запитувати та записувати інформацію про зустріч відповідно до певних умов, а також повне управління додавання, видалення та виправлення користувачів, бронювання послуг та інші можливості.

Основою бази даних є модель даних — фіксована система понять і правил для представлення даних структури, стану і динаміки проблемної області в базі даних. У різний час послідовне застосування одержували ієрархічна, мережева і реляційна моделі даних. У наш час усе більшого поширення набуває об'єктно-орієнтований підхід до організації баз даних.

Види моделей даних:

- Реляційна;

- Ієрархічна;
- Мережева

Реляційна база даних — це база даних, яка організовує інформацію в одну або кілька таблиць. Також, реляційна база даних — це тип бази даних, який зберігає та надає доступ до точок даних, пов'язаних одна з одною. Реляційні бази даних засновані на реляційній моделі, інтуїтивно зрозумілому, простому способі представлення даних у таблицях. У реляційній базі даних кожен рядок таблиці є записом з унікальним ідентифікатором, який називається ключем. Стовпці таблиці містять атрибути даних, і кожен запис зазвичай має значення для кожного атрибута, що полегшує встановлення зв'язків між точками даних. Реляційна модель означає, що логічні структури даних — таблиці даних, представлення даних та індекси — відокремлені від фізичних структур зберігання. Це поділ означає, що адміністратори баз даних можуть керувати фізичним зберіганням даних, не впливаючи на доступ до цих даних як логічної структури. Наприклад, перейменування файлу бази даних не призведе до перейменування таблиць, що зберігаються в ньому. Розрізнення між логічним і фізичним також стосується операцій з базою даних, які є чітко визначеними діями, які дозволяють програмам маніпулювати даними та структурами бази даних. Логічні операції дозволяють програмі вказати потрібний вміст, а фізичні операції визначають, як ці дані мають бути доступні, а потім виконує завдання. Щоб дані завжди були точними та доступними, реляційні бази даних дотримуються певних правил цілісності. Наприклад, правило цілісності може вказувати, що повторювані рядки не допускаються в таблиці, щоб виключити можливість введення помилкової інформації в базу даних.

Ієрархічна структура (модель) будується у вигляді ієрархічної деревоподібної структури, у якій для кожного головного об'єкта існує кілька підлеглих, а для кожного підлеглого об'єкта може бути тільки один головний.

Концептуальна схема ієрархічної моделі являє собою сукупність типів записів, пов'язаних типами зв'язків в одним чи кількома деревами. Усі типи зв'язків цієї

моделі належать до виду «один до декількох» і зображуються у вигляді стрілок. Таким чином, взаємозв'язки між об'єктами нагадують взаємозв'язки в генеалогічному дереві, за єдиним винятком: для кожного породженого (підлеглого) типу об'єкта може бути тільки один вхідний (головний) тип об'єкта. Тобто ієрархічна модель даних допускає тільки два типи зв'язків між об'єктами: «один до одного» і «один до декількох».

У мережевій моделі один і той же об'єкт може одночасно виступати як у ролі головного, так і підлеглого елемента. Це означає, що кожний об'єкт може брати участь у довільній кількості зв'язків. Зв'язок у цьому випадку може встановлюватися явно, коли значення деяких полів є посилання на дані, що містяться в іншому файлі. Прикладом мережевої структури БД може бути структура автобусних маршрутів (із будь-якого населеного пункту існують маршрути в інші). Подібно до ієрархічної, мережеву модель також можна подати у вигляді орієнтованого графу. Але в цьому випадку граф може містити цикли, тобто вершина може мати кілька батьківських вершин.

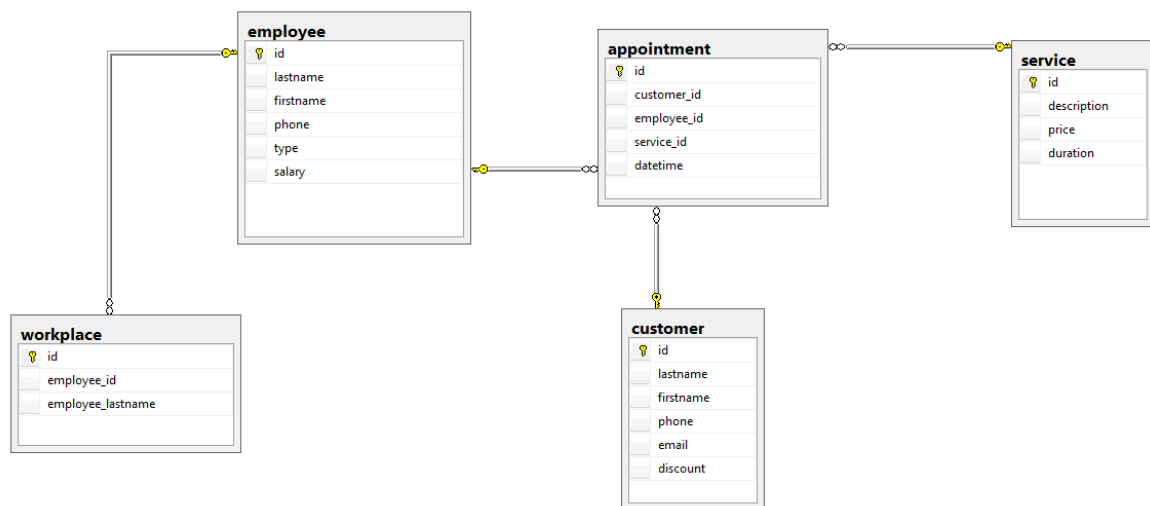


Рис. 2.1 Діаграма сутностей та зв'язків

Для розробки даного проекту було створено реляційну базу даних. Для візуального відображення структури створено ER-діаграму (рис.2.1) Діаграми ER використовуються для моделювання та проектування реляційних баз даних з

точки зору логіки та бізнес-правил (в логічній моделі даних) і з точки зору конкретної технології, яка має бути реалізована (у фізичній моделі даних). У розробці програмного забезпечення, діаграма ER часто є початковим кроком у визначенні вимог до проекту інформаційних систем. Діаграми ER складаються з сутностей, зв'язків та атрибутів.

Діаграма складається з п'яти зв'язаних таблиць: workplace, employee, appointment, service, customer.

Workplace містить інформацію про робоче місце кожного майстра, полями якої є:

- Id — ідентифікаційний код робочого місця;
- Employee_id — ідентифікаційний код працівника;
- Employee_lastname — прізвище працівника

Employee містить інформацію про кожного майстра, полями якої є:

- Id — ідентифікаційний код працівника;
- Lastname — прізвище працівника;
- Firstname — ім'я працівника;
- Phone — номер телефону працівника;
- Type — тип майстра;
- Salary — місячна ставка працівника

Appointment містить інформацію про запис, полями якої є:

- Id — ідентифікаційний код запису;
- Customer_id — ідентифікаційний код клієнта;
- Employee_id — ідентифікаційний код працівника;
- Service_id — ідентифікаційний код послуги;
- Datetime — дата та час запису

Service містить інформацію про послуги, полями якої є:

- Id — ідентифікаційний код послуги;
- Description — назва послуги;

- Price — ціна послуги;
- Duration — тривалість послуги

Customer містить інформацію про клієнтів, полями якої є:

- Id — ідентифікаційний код клієнта;
- Lastname — прізвище клієнта;
- Firstname — ім'я клієнта;
- Phone — номер телефону клієнта;
- Email — електронна пошта клієнта;
- Discount — знижка клієнта

2.2 Нормалізація відношень

Нормалізація бази даних — це процес, і його слід виконувати для кожної бази даних, яку ви проектуєте. Процес створення дизайну бази даних і застосування набору формальних критеріїв і правил називається звичайними формами.

Процес нормалізації бази даних класифікується на такі типи:

1. Перша нормальна форма (1 NF)
2. Друга нормальна форма (2 NF)
3. Третя нормальна форма (3 NF)
4. Нормальна форма Бойса Кодда або Четверта нормальна форма (4 NF)
5. П'ята нормальна форма (5 NF)
6. Шоста нормальна форма (6 NF)

Однією з рушійних сил нормалізації бази даних є оптимізація даних за рахунок зменшення надлишкових даних. Надлишковість даних означає, що в одній базі даних є кілька копій однієї і тієї ж інформації, розповсюджених у кількох місцях.

До недоліків резервування даних можна віднести:

1. Обслуговування даних стає втомливим — видалення даних і оновлення даних стають проблематичними;

2. Це створює невідповідність даних;
3. Аномалії вставки, оновлення та видалення стають частими;
4. Надлишкові дані збільшують розмір бази даних і займають непомірно багато місця на диску.

2.3 Визначення типів даних

Реляційна база даних — це база даних, яка організовує інформацію в одну або кілька таблиць. У свою чергу, таблиця — це набір даних, організованих у рядки та стовпці. Стовпець — це набір значень даних певного типу. Рядок — це один запис у таблиці. З цих тверджень випливає, що усі дані, що зберігаються в реляційній базі даних, мають певний тип. Деякі з найпоширеніших типів даних:

- Int (integer) — ціле число, додатне або від’ємне ціле число
- TEXT — текстовий рядок
- DATE — дата у форматі РРРР-ММ-ДД
- REAL — десяткове значення

Серед менш поширених також присутні: CHAR (size), VARCHAR (size), BOOLEAN, FLOAT (p), DECIMAL (size, d) та інші.

Типи даних, які використовувались для розробки бази даних моєї інформаційної системи:

- INT;
- VARCHAR (size) — рядок змінної довжини (може містити літери, цифри та спеціальні символи). Параметр size визначає максимальну довжину стовпця в символах — може бути від 0 до 65535, в даному випадку — 500;
- DateTime — комбінація дати та часу. Формат: РРРР-ММ-ДД год:мм:сс. Підтримуваний діапазон — від «1000-01-01 00:00:00» до «9999-12-31 23:59:59»

Використання типів даних проілюстровано в наступних таблицях:

Таблиця 2.1

Складові таблиці «workplace»

№	Ім'я атрибуту	Тип даних
1	Id	INT
2	Employee_id	INT
3	Employee_lastname	VARCHAR(50)

Таблиця 2.2

Складові таблиці «service»

№	Ім'я атрибуту	Тип даних
1	Id	INT
2	Description	VARCHAR(500)
3	Price	INT
4	Duration	INT

Таблиця 2.3

Складові таблиці «employee»

№	Ім'я атрибуту	Тип даних
1	Id	INT
2	Lastname	VARCHAR(50)
3	Firstname	VARCHAR(50)
4	Phone	VARCHAR(50)
5	Type	VARCHAR(50)
6	Salary	INT

Таблиця 2.4

Складові таблиці «customer»

№	Ім'я атрибуту	Тип даних
1	Id	INT
2	Lastname	VARCHAR(50)
3	Firstname	VARCHAR(50)
4	Phone	VARCHAR(50)
5	email	VARCHAR(100)
6	discount	INT

Таблиця 2.5

Складові таблиці «appointment»

№	Ім'я атрибуту	Тип даних
1	Id	INT
2	Customer_id	INT
3	Employee_id	INT
4	Service_id	INT
5	Datetime	Datetime

2.4 Обмеження цілісності даних

Цілісність даних — це загальна точність, повнота та узгодженість даних. Коли цілісність даних захищена, інформація, що зберігається в базі даних, залишатиметься повною, точною та надійною незалежно від того, як довго вона зберігається та як часто до неї звертаються. Важливість цілісності даних для захисту від втрати даних або витоку даних неможливо переоцінити: щоб захистити ваші дані від зовнішніх сил зі зловмисними намірами, ви повинні спочатку переконатися, що внутрішні користувачі правильно обробляють дані. Застосовуючи відповідну перевірку даних та перевірку помилок, ви можете гарантувати, що конфіденційні дані ніколи не будуть неправильно класифіковані або не збережені, що наражає вас на потенційний ризик. Цілісність даних можна підтримувати за допомогою обмежень. Ці обмеження визначають правила, згідно з якими такі операції, як оновлення, видалення, вставки тощо, мають виконуватися для підтримки цілісності даних.

Існує декілька обмежень цілісності даних:

- NOT NULL — обмеження для правил, пов'язаних з нульовими значеннями в стовпці;
- UNIQUE — ключові обмеження для правила, пов'язаного з унікальними значеннями стовпців;

- PRIMARY KEY — обмеження для правила, пов'язаного з первинними ідентифікаційними значеннями;
- FOREIGN KEY — обмеження для правил, пов'язаних з цілісністю посилань;
- CHECK — обмеження для складних правил цілісності.

Первинні ключі мають кілька вимог:

- Жодне зі значень не може бути NULL;
- Кожне значення має бути унікальним (тобто ви не можете мати двох клієнтів з однаковим customer_id в таблиці клієнтів);
- Таблиця не може мати більше одного стовпця первинного ключа

Коли первинний ключ для однієї таблиці з'являється в іншій таблиці, він називається зовнішнім ключем (**foreign key**).

В розробці даної бази даних було використано primary та foreign keys, а саме:

1. Primary key — id в таблиці workplace;
2. Primary key — id в таблиці employee;
3. Primary key — id в таблиці appointment;
4. Primary key — id в таблиці service;
5. Primary key — id в таблиці customer;
6. Foreign key — employee_id в таблиці workplace;
7. Foreign key — employee_id в таблиці appointment;
8. Foreign key — service_id в таблиці appointment;
9. Foreign key — customer_id в таблиці appointment.

Всі створені зв'язки проілюстровано на рисунку 2.2.

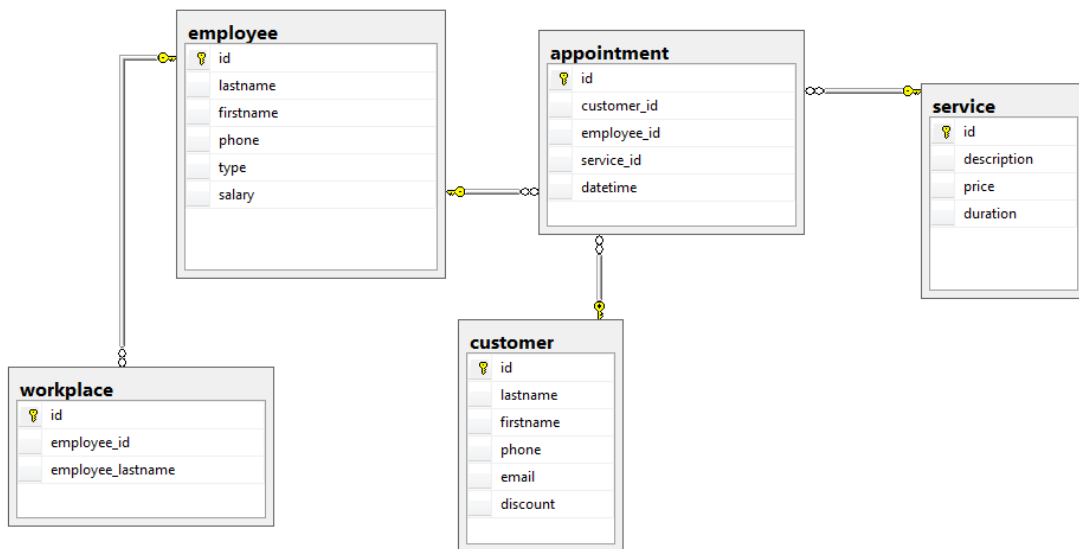


Рис. 2.2 Діаграма сутностей і зв'язків з використанням primary та foreign keys

2.5 Реалізація SQL-скрипту

В базі даних виконується реалізація таких запитів:

`SELECT * FROM customer` — вибір всієї інформації з таблиці customer

`SELECT * FROM appointment` — вибір всієї інформації з таблиці appointment

`SELECT * FROM workplace` — вибір всієї інформації з таблиці workplace

`SELECT * FROM employee` — вибір всієї інформації з таблиці employee

`SELECT * FROM service` — вибір всієї інформації з таблиці service

Звіт за день:

```

SELECT  appointment.customer_id,  customer.lastname,  service.description,
service.price,customer.discount AS 'discount,%',
(service.price*customer.discount)/100 AS 'sum of discount', (service.price-
(service.price*customer.discount)/100) AS 'total service sum'
FROM appointment
JOIN customer on customer.id=appointment.customer_id
JOIN service ON service.id=appointment.service_id
WHERE datetime BETWEEN '2022-01-01 00:00' AND '2022-01-01 23:59';

```

Сума зароблених коштів за день+к-сть сервісів:

```

SELECT SUM(service.price-(service.price*customer.discount)/100) AS 'зароблене
за день', COUNT(service.id) AS 'к-сть послуг'
FROM appointment
JOIN customer on customer.id=appointment.customer_id
JOIN service ON service.id=appointment.service_id
WHERE datetime BETWEEN '2022-01-01 00:00' AND '2022-01-01 23:59';

```

Сума зароблених коштів за місяць:

```

SELECT SUM(service.price-(service.price*customer.discount)/100) AS 'зароблене
за місяць', COUNT(service.id) AS 'к-сть послуг'
FROM appointment
JOIN customer on customer.id=appointment.customer_id
JOIN service ON service.id=appointment.service_id
WHERE datetime BETWEEN '2022-01-01 00:00' AND '2022-01-31 23:59';

```

Відсоток від послуги:

```

SELECT employee.lastname, service.description, service.price, (service.price/2) as
'відсоток від послуги'
FROM appointment
join employee on employee.id=appointment.employee_id
join service on service.id=appointment.service_id
order by employee.lastname

```

Звіт по зарплаті:

```

SELECT employee.lastname, SUM(service.price/2) as 'зарплата без ставки ',
employee.salary as 'ставка',(SUM(service.price/2)+employee.salary) as 'зарплата за
місяць '
FROM appointment
join employee on employee.id=appointment.employee_id
join service on service.id=appointment.service_id
group by employee.lastname,employee.salary
order by employee.lastname

```

Хто заслуговує на знижку:

```

SELECT customer.lastname, COUNT(appointment.id) as 'к-сть відвідувань за
січень'
from appointment
join customer on customer.id= appointment.customer_id
where datetime between '2022-01-01 00:00' and '2022-01-31 23:59'
group by customer.lastname
having COUNT(appointment.id)>=3;

```

Яку послугу замовляють найчастіше:

```
SELECT service.description, COUNT(appointment.id) as 'найпопулярніші послуги'
from appointment
join service on service.id=appointment.service_id
group by service.description
order by COUNT(appointment.id) desc;
```

До якого майстра найчастіше записуються:

```
SELECT     employee.lastname,employee.type,     COUNT(appointment.id)     as
'найпопулярніші майстри '
from appointment
join employee on employee.id=appointment.employee_id
group by employee.lastname,employee.type
order by COUNT(appointment.id) desc;
```

Графік роботи на день для візажистів:

```
select employee.lastname, employee.type,customer.lastname, service.description,
datetime, service.duration
from appointment
join employee on employee.id=appointment.employee_id
join customer on customer.id=appointment.customer_id
join service on service.id=appointment.service_id
where datetime>='2022-01-01 10:00' and datetime<= '2022-01-01 18:00'
and employee.type='візажист'
order by employee.lastname
```

ВИСНОВКИ

В ході виконання завдання проаналізовано використання SQL, також розроблено базу даних для салону краси, яка б у перспективі дозволяла майбутнім користувачам веб-сайту зручно та легко ставати клієнтами салону, а також з включенням в базу різноманітних таблиць з використанням варіаційних типів даних та зв'язків.

Розроблено Use Case діаграму для трьох користувачів інформаційної системи, а саме: клієнта, адміністратора та працівника. Така діаграма дозволила б розробникові швидше орієнтуватись в бізнес-моделі інформаційної системи, покращила хід виконання розробки.

Однією з найважливіших ланок даної праці стала розробка ER-діаграми, в котрій присутні п'ять таблиць з різним наповненням та інформацією. Така діаграма ілюструє, як «суб'єкти», такі як люди, об'єкти або поняття, пов'язані один з одним у системі. Також було визначено її сутності, атрибути та зв'язки.

Було спроектовано та описано структуру БД, з реалізацією різного роду запитів та наповнення таблиць. Запити та наповнення описані в розділах курсової роботи та додатках відповідно.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інформаційні системи [Електронний ресурс]. — Режим доступу: https://pidru4niki.com/1222090547713/informatika/informatsiyni_sistemi
2. Моделі баз даних [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%B%D1%96_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85
3. НОРМАЛІЗАЦІЯ ВІДНОШЕНЬ [Електронний ресурс]. — Режим доступу: https://life-prog.ru/ukr/1_331_normalizatsiya-vidnoshen.html
4. Що таке діаграма варіантів використання [Електронний ресурс]. — Режим доступу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>
5. Що таке нормалізація бази даних у SQL Server? [Електронний ресурс]. — Режим доступу: <https://www.sqlshack.com/what-is-database-normalization-in-sql-server/>
6. Beauty salon management system for database curriculum design practice [Електронний ресурс]. — Режим доступу: <https://blog.fearcat.in/a?ID=01600-90f46b10-1b8c-4f00-b126-a602aaf5cbb5>
7. Entity relational diagram. [Електронний ресурс]. — Режим доступу: <https://www.smartdraw.com/entity-relationship-diagram/>
8. Information system [Електронний ресурс]. — Режим доступу: <https://www.britannica.com/topic/information-system>
9. Introduction To SQL [Електронний ресурс]. — Режим доступу: <https://www.codecademy.com/courses/learnsql/lessons/manipulation/exercises/sql>
10. Relational Database [Електронний ресурс]. — Режим доступу: <https://www.techtarget.com/searchdatamanagement/definition/relational-database>

- 11.The 7 main ways technology impacts your daily life [Электронный ресурс]. — Режим доступа: <https://tech.co/vpn/main-ways-technology-impacts-daily-life>
12. Use Case Diagram [Электронный ресурс]. — Режим доступа: <https://www.lucidchart.com/pages/uml-use-case-diagram>
- 13.What is a relational database? [Электронный ресурс]. — Режим доступа: <https://www.oracle.com/database/what-is-a-relational-database/>
- 14.What is data integrity [Электронный ресурс]. — Режим доступа: <https://afteracademy.com/blog/what-is-data-integrity>
- 15.What is data integrity and why is it important [Электронный ресурс]. — Режим доступа: <https://ua.talend.com/resources/what-is-data-integrity/>
- 16.What is use case diagram [Электронный ресурс]. — Режим доступа: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>
- 17.What is an Entity Relationship Diagram (ERD)? [Электронный ресурс]. — Режим доступа: <https://www.lucidchart.com/pages/er-diagrams>

ДОДАТКИ

Додаток А

Лістинг А.1 Наповнення таблиці workplace

```
INSERT INTO workplace(id,employee_id) VALUES(1,1)
INSERT INTO workplace(id,employee_id) VALUES(2,2)
INSERT INTO workplace(id,employee_id) VALUES(3,3)
INSERT INTO workplace(id,employee_id) VALUES(4,4)
INSERT INTO workplace(id,employee_id) VALUES(5,5)
INSERT INTO workplace(id,employee_id) VALUES(6,6)
INSERT INTO workplace(id,employee_id) VALUES(7,7)
INSERT INTO workplace(id,employee_id) VALUES(8,8)
INSERT INTO workplace(id,employee_id) VALUES(9,9)
INSERT INTO workplace(id,employee_id) VALUES(10,10)
INSERT INTO workplace(id,employee_id) VALUES(11,11)
INSERT INTO workplace(id,employee_id) VALUES(12,12)
```

Лістинг А.2 Наповнення таблиці customer

```
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(1,'Дудинська', 'Яна', '0504897410', 'dudynskayana@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(2,'Вербицька', 'Карина', '0508520369', 'verbitskakaryna@gmail.com',10)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(3,'Стринадюк', 'Аліна', '0957895623', 'strynadukalina@gmail.com',10)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(4,'Третяк', 'Ольга', '0956302154', 'tretyakolga@gmail.com',10)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(5,'Онкович', 'Дарина', '0507894563', 'onkovychdaruna@gmail.com',0)
```

```

INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(6,'Загаєвич', 'Марина', '0957841023', 'zagaevychmaryna@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(7,'Покропивна', 'Анастасія', '0661470258', 'pokropyvnanastya@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(8,'Єднак', 'Мирослава', '0504445263', 'yednakmuroslava@com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(9,'Савчук', 'Сніжана', '0956663322', 'savchuksnizhana@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(10,'Медведюк', 'Лідія', '0507770412', 'medvedyuklidiya@com',10)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(11,'Алчевська', 'Алла', '0505554411', 'alchevskaalla@com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(12,'Візерські', 'Валентина', '0507412589', 'vizerskivalentyna@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(13,'Арсенич', 'Анна', '0669630258', 'arsenychanna@com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(14,'Бутник', 'Христина', '0669874512', 'bytnukkhrystyna@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(15,'Суховерська', 'Маргарита', '0957411478',
'sukhoverskamargaryta@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(16,'Крижанівська', 'Ярослава', '0502223366',
'kryzhanivskayaroslava@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(17,'Франчук', 'Єлизавета', '0667770000', 'franchykliza@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(18,'Палій', 'Ольга', '0958889966', 'paliyolga@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(19,'Дрозденко', 'Діана', '0501110033', 'drozdenkodian@gmail.com',0)

```

```

INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(20,'Заярна', 'Олеся', '0957854123', 'zayarnaolesya@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(21,'Михальська', 'Любов', '0954563622', 'mukhalskalyubov@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(22,'Лاپинська', 'Яна', '0954441200', 'lapynskayana@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(23,'Гнатишин', 'Марія', '0971110101', 'hnatyshynmariya@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(24,'Семенко', 'Наталія', '0630233366', 'semenkonataliya@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(25,'Собчук', 'Софія', '0630002132', 'sobchyksophiya@gmail.com',10)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(26,'Лисяк', 'Катерина', '0957771144', 'lusiakkaterunagmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(27,'Шеремета', 'Руслана', '0504447788', 'sheremetaruslanagmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(28,'Кульженко', 'Ольга', '0500123335', 'kulzhenkoolga@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(29,'Сапеляк', 'Олександра', '0631478565', 'sapelyaksasha@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(30,'Якобовська', 'Наталія', '0957488844', 'yakobovskanataliya@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(31,'Ялова', 'Юлія', '0951445500', 'yalovayuliya@gmail.com',10)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(32,'Лупій', 'Іванна', '0951110000', 'lupiyivanna@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(33,'Гуц', 'Тетяна', '0959986633', 'hutstetyana@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(34,'Демчишин', 'Софія', '0970223562', 'demchyshynsophiya@gmail.com',0)

```

```

INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(35,'Бугрова', 'Анастасія', '0957844400', 'bugrovanastya@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(36,'Мисько', 'Аліна', '0630213398', 'muskoalina@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(37,'Бабляк', 'Тетяна', '0503216547', 'bablyaktetiana@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(38,'Слободянюк', 'Ольга', '0956330011', 'slobodyanukolga@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(39,'Авраменко', 'Ірина', '0632117788', 'avramenkoiryna@gmail.com',0)
INSERT INTO customer(id,lastname,firstname,phone,email,discount) VALUES
(40,'Бродович', 'Наталія', '0974412506', 'brodovychnataliya@gmail.com',0)

```

Лістинг А.3 Наповнення таблиці service

```

INSERT INTO service(id, description,price,duration) VALUES (1, 'Макіяж Express',
350, 80)
INSERT INTO service(id, description,price,duration) VALUES (2, 'Макіяж Day',
500, 65)
INSERT INTO service(id, description,price,duration) VALUES (3, 'Макіяж Night',
650, 80)
INSERT INTO service(id, description,price,duration) VALUES (4, 'Макіяж
Hollywood', 750, 90)
INSERT INTO service(id, description,price,duration) VALUES (5, 'Макіяж
Wedding', 850, 120)
INSERT INTO service(id, description,price,duration) VALUES (6, 'накладні вії', 200,
20)
INSERT INTO service(id, description,price,duration) VALUES (7, 'пучок вій', 20, 10)
INSERT INTO service(id, description,price,duration) VALUES (8, 'Манікюр
класичний', 230, 65)

```

INSERT INTO service(id, description,price,duration) VALUES (9, 'Манікюр комбінований', 230, 65)

INSERT INTO service(id, description,price,duration) VALUES (10, 'Манікюр апаратний', 230, 65)

INSERT INTO service(id, description,price,duration) VALUES (11, 'Зняття старого матеріалу', 80, 25)

INSERT INTO service(id, description,price,duration) VALUES (12, 'Масаж рук', 80, 20)

INSERT INTO service(id, description,price,duration) VALUES (13, 'Ремонт нігтя', 40, 10)

INSERT INTO service(id, description,price,duration) VALUES (14, 'Нарощування нігтя', 55, 15)

INSERT INTO service(id, description,price,duration) VALUES (15, 'Педикюр (апаратна чистка+покриття)', 320, 65)

INSERT INTO service(id, description,price,duration) VALUES (16, 'Покриття гель-лаком', 200, 40)

INSERT INTO service(id, description,price,duration) VALUES (17, 'Комбінована чистка ', 250, 30)

INSERT INTO service(id, description,price,duration) VALUES (18, 'Масаж ніг', 80, 20)

INSERT INTO service(id, description,price,duration) VALUES (19, 'Фарбування брів', 130, 60)

INSERT INTO service(id, description,price,duration) VALUES (20, 'Фарбування брів хною', 150, 60)

INSERT INTO service(id, description,price,duration) VALUES (21, 'Ламінування брів', 400, 30)

INSERT INTO service(id, description,price,duration) VALUES (22, 'Ламінування+фарбування', 550, 90)

INSERT INTO service(id, description,price,duration) VALUES (23, 'Накрутка "Голівудські локони"', 400, 60)

```
INSERT INTO service(id, description,price,duration) VALUES (24, 'Афро накрутка', 350, 60)
```

```
INSERT INTO service(id, description,price,duration) VALUES (25, 'Вечірня укладка', 600, 80)
```

```
INSERT INTO service(id, description,price,duration) VALUES (26, 'Плетіння', 550, 50)
```

```
INSERT INTO service(id, description,price,duration) VALUES (27, 'Вирівнювання утюжком без миття голови', 150, 20)
```

```
INSERT INTO service(id, description,price,duration) VALUES (28, 'Весільна зачіска', 600, 60)
```

```
INSERT INTO service(id, description,price,duration) VALUES (29, 'Фарбування коренів', 500, 120)
```

```
INSERT INTO service(id, description,price,duration) VALUES (30, 'Фарбування тон в тон', 1200,150)
```

```
INSERT INTO service(id, description,price,duration) VALUES (31, 'Тонування', 1000, 120)
```

```
INSERT INTO service(id, description,price,duration) VALUES (32, 'Балаяж', 2300, 180)
```

```
INSERT INTO service(id, description,price,duration) VALUES (33, 'Стрижка 1 довжина', 300, 60)
```

```
INSERT INTO service(id, description,price,duration) VALUES (34, 'Стрижка 2 довжина', 400, 60)
```

```
INSERT INTO service(id, description,price,duration) VALUES (35, 'Стрижка 3 довжина', 500, 60)
```

```
INSERT INTO service(id, description,price,duration) VALUES (36, 'Стрижка кінчиків(без миття)', 200, 40)
```

```
INSERT INTO service(id, description,price,duration) VALUES (37, 'Стрижка чілки(без миття)', 150, 40)
```

```
INSERT INTO service(id, description,price,duration) VALUES (38, 'Дитяча стрижка', 250, 35)
```

```
INSERT INTO service(id, description,price,duration) VALUES (39, 'Миття голови',
100, 40)
```

```
INSERT INTO service(id, description,price,duration) VALUES (40, 'Мезотерапія
очей', 2150, 60)
```

```
INSERT INTO service(id, description,price,duration) VALUES (41, 'Мезотерапія
шкіри голови', 1400, 60)
```

```
INSERT INTO service(id, description,price,duration) VALUES (42,
'Ботулінотерапія', 100, 40)
```

```
INSERT INTO service(id, description,price,duration) VALUES (43, 'Контурна
пластика губ', 7300, 60)
```

```
INSERT INTO service(id, description,price,duration) VALUES (44, 'Механічна
чистка обличчя', 800, 60)
```

```
INSERT INTO service(id, description,price,duration) VALUES (45, 'Пілінг', 750, 60)
```

Лістинг А.4 Наповнення таблиці employee

```
INSERT INTO employee(id,lastname,firstname,phone,type,salary) VALUES (1,
'Білоус', 'Катерина', '0506998745', 'візажист', 10000)
```

```
INSERT INTO employee(id,lastname,firstname,phone,type,salary) VALUES (2,
'Романюк', 'Ангеліна', '0668552312', 'візажист', 11000)
```

```
INSERT INTO employee(id,lastname,firstname,phone,type,salary) VALUES (3,
'Петришин', 'Ольга', '0502331474', 'майстер манікюру', 15000)
```

```
INSERT INTO employee(id,lastname,firstname,phone,type,salary) VALUES (4,
'Кузьменко', 'Олександра', '0668595421', 'майстер манікюру', 15000)
```

```
INSERT INTO employee(id,lastname,firstname,phone,type,salary) VALUES (5,
'Труш', 'Христина', '0503698520', 'майстер педикюру', 10000)
```

```
INSERT INTO employee(id,lastname,firstname,phone,type,salary) VALUES (6,
'Сірик', 'Дарина', '0506998746', 'майстер педикюру', 10000)
```

```
INSERT INTO employee(id,lastname,firstname,phone,type,salary) VALUES (7,
'Таранець', 'Діана', '0507894152', 'бровист', 12000)
```

```
INSERT INTO employee(id,lastname,firstname,phone,type,salary) VALUES (8,
'Гуцал', 'Марина', '0661478965', 'бровист', 12000)
```

```
INSERT INTO employee(id,lastname,firstname,phone,type,salary) VALUES (9,
'Хомик', 'Ярослава', '0660251436', 'стиліст по зачісках', 15000)
```

```
INSERT INTO employee(id,lastname,firstname,phone,type,salary) VALUES (10,
'Гевко', 'Олена', '0669630201', 'колорист', 16000)
```

```
INSERT INTO employee(id,lastname,firstname,phone,type,salary) VALUES (11,
'Бойчук', 'Ірина', '0504715823', 'перукар', 12000)
```

```
INSERT INTO employee(id,lastname,firstname,phone,type,salary) VALUES (12,
'Стефанко', 'Марія', '0669632585', 'косметолог', 20000)
```

Лістинг А.5 Наповнення таблиці appointment

```
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (1, 1, 11, 34, '01-01-2022 10:00')
```

```
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (2, 2, 11, 36, '01-01-2022 11:00')
```

```
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (3, 3, 11, 38, '01-01-2022 13:00')
```

```
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (4, 4, 11, 36, '01-01-2022 14:00')
```

```
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (5, 5, 11, 34, '01-01-2022 15:00')
```

```
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (6, 6, 11, 37, '01-01-2022 17:00')
```

```
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (7, 7, 8, 22, '01-01-2022 12:00')
```

```
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (8, 8, 8, 22, '01-01-2022 14:00')
```



```
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (9, 9, 8, 21, '01-01-2022 16:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (10, 10, 3, 8, '01-01-2022 10:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (11, 11, 3, 11, '01-01-2022 11:30')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (12, 12, 3, 10, '01-01-2022 12:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (13, 13, 3, 8, '01-01-2022 13:30')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (14, 14, 3, 8, '01-01-2022 14:45')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (15, 15, 3, 9, '01-01-2022 16:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (16, 16, 3, 9, '01-01-2022 17:30')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (17, 17, 1, 1, '01-01-2022 10:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (18, 18, 1, 2, '01-01-2022 12:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (19, 19, 1, 2, '01-01-2022 14:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (20, 20, 1, 2, '01-01-2022 16:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (21, 21, 4, 12, '01-01-2022 10:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (22, 22, 4, 8, '01-01-2022 11:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (23, 23, 4, 10, '01-01-2022 12:30')
```

```
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (24, 24, 4, 10, '01-01-2022 14:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (25, 25, 4, 13, '01-01-2022 15:30')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (26, 26, 4, 8, '01-01-2022 16:30')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (27, 27, 5, 15, '01-01-2022 10:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (28, 28, 5, 15, '01-01-2022 11:30')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (29, 29, 5, 16, '01-01-2022 13:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (30, 30, 5, 15, '01-01-2022 14:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (31, 31, 9, 28, '01-01-2022 10:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (32, 32, 9, 27, '01-01-2022 11:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (33, 33, 9, 26, '01-01-2022 11:30')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (34, 34, 9, 26, '01-01-2022 12:30')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (35, 35, 9, 23, '01-01-2022 14:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (36, 36, 9, 23, '01-01-2022 15:30')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (37, 37, 9, 26, '01-01-2022 17:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (38, 38, 12, 45, '01-01-2022 10:00')
```

```
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (39, 39, 12, 44, '01-01-2022 12:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (40, 40, 12, 40, '01-01-2022 16:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (41, 13, 12, 45, '01-01-2022 17:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (42, 1, 10, 30, '01-01-2022 11:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (43, 12,10 , 32, '01-01-2022 14:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (44, 38, 6, 15, '01-01-2022 11:00')
INSERT INTO appointment(id, customer_id,employee_id,service_id,datetime)
VALUES (45, 38, 2, 2, '01-01-2022 12:10')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (46, 20, 5, 18, '2022-01-02 10:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (47, 4, 1, 2, '2022-01-03 12:00' )
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (48, 25, 9,23, '2022-01-04 16:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (49, 10, 3, 9, '2022-01-05 12:30')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (50, 10 ,7, 20, '2022-01-06 11:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (51, 10, 8, 19, '2022-01-07 14:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (52, 14, 4,10, '2022-01-08 17:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (53, 21, 6,15,'2022-01-09 15:00')
```

```
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (54, 32, 12,45, '2022-01-10 13:30')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (55, 40, 11, 37, '2022-01-11 16:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (56, 37, 4, 8, '2022-01-12 14:30')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (57, 37, 12, 43, '2022-01-13 16:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (58, 33, 9,26, '2022-01-14 10:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (59, 12, 10, 30, '2022-01-15 15:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (60, 17, 5,17, '2022-01-16 12:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (61, 17, 1, 3,'2022-01-17 17:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (62, 28 , 2,4, '2022-01-18 16:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (63, 36, 6,15, '2022-01-19 16:30')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (64, 21, 10,32, '2022-01-20 14:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (65, 14, 11, 38, '2022-01-21 11:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (66, 11, 11, 35, '2022-01-22 14:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (67, 21,11, 36,'2022-01-23 16:00')
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (68, 4, 4 ,10,'2022-01-24 10:00')
```

```
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (69, 17, 11, 33,'2022-01-25 14:00')
```

```
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (70, 17, 12, 45, '2022-01-26 16:30')
```

```
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (71, 29, 10, 31, '2022-01-27 15:00')
```

```
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (72, 16, 7,19, '2022-01-28 14:30')
```

```
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (73, 11, 12, 44, '2022-01-29 11:00')
```

```
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (74, 11, 12, 45, '2022-01-30 13:00')
```

```
INSERT INTO appointment (id, customer_id,employee_id,service_id,datetime)
VALUES (75, 16, 9, 23, '2022-01-31 15:30')
```