**РОЗДІЛ 4. ПРОЕКТУВАННЯ БАЗИ ДАНИХ (БД)**

**4.1. Розробка і створення концептуальної моделі даних**

Початковий етап розробки моделі включає побудову інформаційної моделі для автоматизації управління мережею барбершопів, яка може бути виражена різними знаковими формами. На заключному етапі ці знакові форми втілюються в комп'ютерну модель. Таким чином, інформаційна модель відображається в певній знаковій формі, наприклад, у вигляді таблиць бази даних.

Таблиця "users" призначена для зберігання даних про користувачів системи управління барбершопами.

1. id - унікальний ідентифікатор користувача, який автоматично збільшується.
2. first\_name - ім'я користувача.
3. last\_name - прізвище користувача.
4. email - електронна пошта користувача.
5. role - роль користувача в системі (адміністратор, менеджер, майстер).
6. password - зашифрований пароль користувача.
7. created\_at - час створення облікового запису.
8. updated\_at - час останнього оновлення облікового запису.

Таблиця "shops" містить локації барбершопів в наступних стовпцях:

1. id - унікальний ідентифікатор барбершопу.
2. title - назва барбершопу.
3. address - адреса розташування.
4. time\_from - час початку роботи.
5. time\_to - час закінчення роботи.
6. image - зображення барбершопу.
7. content - опис барбершопу.
8. created\_at - час створення запису.
9. updated\_at - час останнього оновлення запису.

Ця таблиця "shops" призначена для зберігання інформації про різні барбершопи. Кожен запис таблиці відповідає окремому філіалу і містить всі дані про нього.

Таблиця "services" містить наступні стовпці:

1. id - унікальний ідентифікатор послуги.
2. title - назва послуги.
3. price - вартість послуги.
4. service\_time - тривалість послуги.
5. image - зображення, що ілюструє послугу.
6. description - детальний опис послуги.
7. created\_at - час створення запису.
8. updated\_at - час останнього оновлення запису.

Ця таблиця "services" призначена для зберігання інформації про послуги. Кожен запис таблиці відповідає окремій послузі, яку надають барбершопи.

Таблиця "orders" містить наступні стовпці:

1. id - унікальний ідентифікатор замовлення.
2. client\_id - ідентифікатор клієнта.
3. service\_id - ідентифікатор обраної послуги.
4. shop\_id - ідентифікатор барбершопу.
5. selectedTime - обраний час для запису.
6. selectedDate - обрана дата для запису.
7. status - статус замовлення (новий, виконаний, скасований).
8. created\_at - час створення замовлення.
9. updated\_at - час останнього оновлення замовлення.

Ця таблиця " orders " призначена для зберігання інформації про замовлення. Кожен запис таблиці відповідає окремому замовленню.

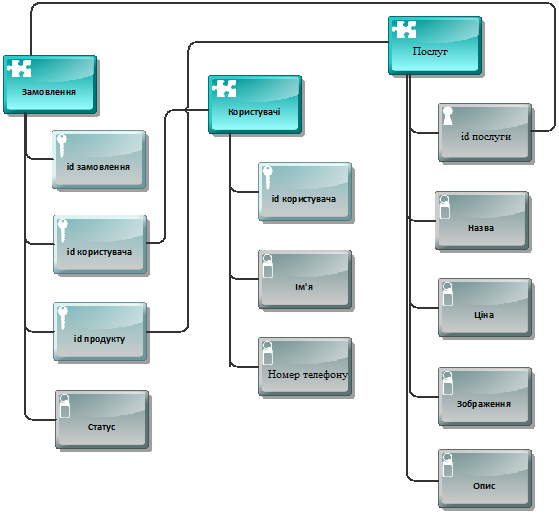


Рис. 4.1 – Концептуальна модель даних бізнес процесу «Надання послуг»

**4.2. Розробка і створення структури таблиць БД**

При проектуванні фізичної моделі бази даних для мережі барбершопів особлива увага приділяється оптимальному вибору типів даних для кожного атрибута. Правильна типізація забезпечує ефективне використання дискового простору, швидкість операцій та цілісність інформації. Розглянемо типологію атрибутів, що застосовується в системі:

Таблиця 4.1 – Типи даних

| **Тип даних** | **Розмір (байт)** | **Діапазон значень** | **Опис** |
| --- | --- | --- | --- |
| INT | 4 | -2,147,483,648 до 2,147,483,647 | Цілі числа без десяткової частини, які можуть бути позитивними, негативними або нульовими. Використовується для зберігання цілих числових значень, таких як ідентифікатори, кількість продуктів або вік. |
| FLOAT | 4 | ±3.402823466E+38 | Числа з плаваючою комою, які представляють десяткові числа. Вони можуть мати десяткову частину та експоненційну форму. Наприклад, 3.14 або 1.2E-5. Точність FLOAT може залежати від реалізації СКЛ бази даних. |
| DOUBLE | 8 | ±1.7976931348623157E+308 | Подібно до FLOAT, але з більшою точністю для зберігання значень з плаваючою комою. DOUBLE використовує більше байтів для зберігання значень, тому він може мати більшу точність, але й займати більше місця. |
| DECIMAL | Змінний | Залежить від точності, заданої користувачем | Точне числове значення з фіксованою точністю, заданою користувачем. Використовується для точних обчислень, де важлива точність до заданої кількості десяткових знаків. |
| VARCHAR | Змінний | Від 0 до 65,535 знаків (байт) | Рядки зі змінною довжиною, які можуть містити букви, цифри та спеціальні символи. Максимальна довжина VARCHAR залежить від реалізації СКЛ бази даних. |
| CHAR | Фіксовий | Від 0 до 255 знаків (байт) | Рядки фіксованої довжини, які можуть містити букви, цифри та спеціальні символи. CHAR використовує фіксований обсяг пам'яті незалежно від фактичної довжини рядка, тому він може займати більше місця, якщо рядок є коротким. |
| TEXT | Змінний | Від 0 до 65,535 знаків (байт) | Довгі текстові значення, такі як описи або коментарі. TEXT може зберігати значно більше символів, ніж VARCHAR, і часто використовується для зберігання великих текстових блоків, таких як статті, замітки або великі коментарі. |
| DATE | 3 | '1000-01-01' до '9999-12-31' | Дата без врахування часу. Використовується для зберігання дати, такої як день народження, дата створення запису тощо. |
| TIME | 3 | '-838:59:59' до '838:59:59' | Час без врахування дати. Використовується для зберігання часу, наприклад, часу запису або тривалості послуги. |
| DATETIME | 8 | '1000-01-01 00:00:00' до '9999-12-31 23:59:59' | Комбінація дати та часу. Використовується для зберігання обох значень дати та часу, наприклад, дати та часу створення запису або запланованого прийому. |
| TIMESTAMP | 4 | 1970-01-01 00:00:01 UTC до 2038-01-19 03:14:07 UTC | Використовується для зберігання мітки часу, яка автоматично оновлюється при внесенні змін у запис. Зазвичай відображає кількість секунд, що пройшли з 1 січня 1970 року (відомий як "Unix Epoch") до поточного моменту. |
| BOOLEAN | 1 | TRUE або FALSE | Логічне значення, яке може бути TRUE або FALSE. Використовується для зберігання булевих значень, таких як прапорці або умови. |

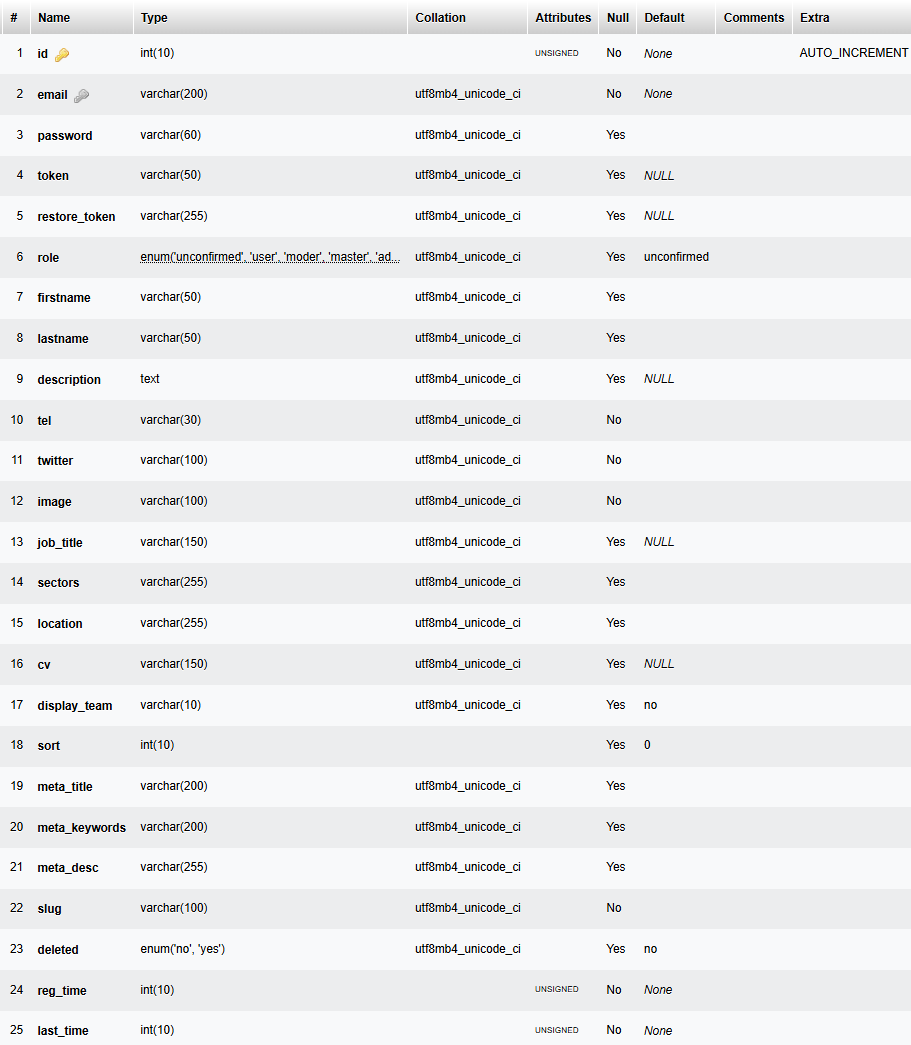


Рис. 4.2 – Структура таблиці **«users»**

Таблиця "users" виступає ядром системи ідентифікації та аутентифікації учасників бізнес-процесу з наступною технічною специфікацією:

1. "id": Первинний ключ високої кардинальності типу "bigint" з атрибутами UNSIGNED та AUTO\_INCREMENT для гарантованої унікальності та автоматичної генерації при внесенні нових суб'єктів до системи.
2. "first\_name": Строковий атрибут у форматі "varchar(255)" з підтримкою багатомовності через колацію "utf8mb4\_unicode\_ci" та обмеженням NOT NULL для забезпечення обов'язкового заповнення персональних даних.
3. "last\_name": Строковий атрибут аналогічної конфігурації з "first\_name", що в комбінації забезпечує формування повного імені для інтерфейсів системи та клієнтської комунікації.
4. "role": Енумераційний атрибут у форматі "varchar(255)" з фіксованим набором категоріальних значень, що визначає матрицю доступу користувача до функціоналу системи відповідно до бізнес-ролі.
5. "email": Унікальний строковий атрибут з індексацією для оптимізації пошукових операцій, що виконує функцію логіну та контактної інформації з відповідними валідаційними правилами.
6. "password": Строковий атрибут підвищеної конфіденційності для зберігання хешованих значень аутентифікаційних даних з застосуванням сучасних криптографічних алгоритмів.
7. "created\_at": Темпоральний атрибут типу "timestamp" для фіксації моменту створення облікового запису з можливістю прийняття NULL-значень для зворотної сумісності з історичними даними.
8. "updated\_at": Темпоральний атрибут аналогічної конфігурації для фіксації моменту модифікації запису з автоматичним оновленням при виконанні операцій оновлення даних.

Ця спеціалізована структура забезпечує ефективне управління обліковими записами, диференціацію рівнів доступу та аудит змін у критично важливій сутності системи.

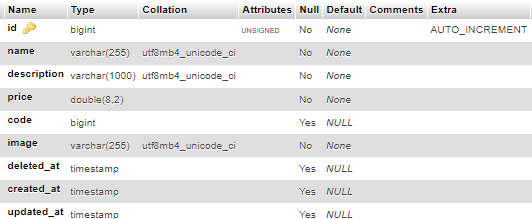


Рис. 4.3 – Структура таблиці **«products»**

Таблиця "products" має наступну структуру:

1. "id" - тип поля "bigint UNSIGNED". Це первинний ключ (Primary Key) таблиці, який автоматично збільшується (AUTO\_INCREMENT).
2. "name" - тип поля "varchar(255)". Це текстове поле, яке може містити до 255 символів.
3. "description" - тип поля "varchar(1000)". Це текстове поле, яке може містити до 1000 символів.
4. "price" - тип поля "double(8,2)". Це числове поле з плаваючою комою, яке може мати до 8 цифр загалом, з 2 знаками після коми.
5. "code" - тип поля "bigint". Це числове поле, яке може містити довгі цілі числа. Це поле також допускає значення "NULL" (порожнє).
6. "image" - тип поля "varchar(255)". Це текстове поле, яке може містити до 255 символів.
7. "deleted\_at" - тип поля "timestamp". Це поле дати та часу, яке використовується для вказання часу, коли запис був видалений. Це поле також допускає значення "NULL" (порожнє).
8. "created\_at" - тип поля "timestamp". Це поле дати та часу, яке вказує на час створення запису. Це поле також допускає значення "NULL" (порожнє).
9. "updated\_at" - тип поля "timestamp". Це поле дати та часу, яке вказує на час останнього оновлення запису. Це поле також допускає значення "NULL" (порожнє).



Рис. 4.4 – Структура таблиці **«services»**

Сутність "services" імплементована як оптимізована структура з наступними технічними характеристиками:

1. "id" - інкрементальний ідентифікатор типу "bigint UNSIGNED" з атрибутом AUTO\_INCREMENT, що забезпечує унікальність та послідовність нумерації при розширенні каталогу послуг.
2. "name" - лінгвістичний дескриптор типу "varchar(255)" для маркетингового найменування сервісної пропозиції з обмеженням оптимальної довжини для користувацьких інтерфейсів.
3. "description" - розширений текстовий атрибут типу "varchar(1000)" для деталізованого опису процедури, матеріалів та очікуваних результатів з достатнім обсягом для конверсійного маркетингового контенту.
4. "price" - фінансовий атрибут типу "double(8,2)" з прецизійною конфігурацією для відображення вартості з точністю до копійок і можливістю гнучкого цінового діапазону.
5. "duration" - операційний числовий атрибут типу "int" з дозволеними NULL-значеннями для тривалості виконання в хвилинах, критичний для оптимізації розкладу та планування робочого навантаження.
6. "image" - референсний атрибут типу "varchar(255)" для інтеграції візуального контенту через посилання або відносні шляхи до ресурсів у файловій системі.
7. "deleted\_at" - службовий темпоральний маркер типу "timestamp" з підтримкою NULL для реалізації механізму програмного видалення (soft delete), що забезпечує цілісність історії та аналітики без фізичного знищення даних.
8. "created\_at" - хронологічний атрибут типу "timestamp" для фіксації моменту внесення послуги до каталогу з підтримкою NULL для історичних записів.
9. "updated\_at" - хронологічний атрибут типу "timestamp" для аудиту модифікацій з підтримкою NULL та автоматичним оновленням при зміні характеристик послуги.