

## **ЗМІСТ**

<b>ВСТУП</b>	<b>3</b>
<b>1. ПОСТАНОВКА ЗАДАЧІ</b>	<b>4</b>
<b>2. АРХІТЕКТУРА ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ БД</b>	<b>6</b>
2.1. Аналіз функціонування та організаційні основи підприємства	6
2.2. Проектування структури баз даних	7
2.3. Життєві цикли БД	12
<b>3. РЕАЛІЗАЦІЯ ПРОГРАМНОЇ ВЗАЄМОДІЇ З БД</b>	<b>14</b>
3.1. Керівництво користувача	14
3.2. Реалізація механізмів SQL	16
3.2.1. SQL-запити	16
3.2.2. Процедури та функції	16
3.2.3. Тригери	19
3.3. Вимоги до апаратних та програмних засобів	19
3.4. Випробування розроблених програм	19
3.5. Опис тестової бази даних	29
<b>4. ВИСНОВОК</b>	<b>40</b>
<b>5. ЛІТЕРАТУРА</b>	<b>42</b>
<b>ДОДАТКИ</b>	<b>43</b>
Додаток А: Реалізація процедур	43

## **ВСТУП**

Інформаційні системи (ІС) стали невід'ємною частиною сучасної медичної галузі, сприяючи ефективному управлінню медичними закладами та забезпеченням високої якості медичних послуг. Автоматизовані інформаційні системи (АІС) медичних організацій дозволяють не лише оптимізувати внутрішні процеси, але й забезпечують точне та своєчасне ведення медичної документації, полегшуючи роботу лікарів та обслуговуючого персоналу.

# **1. ПОСТАНОВКА ЗАДАЧІ**

Мета цієї курсової роботи - розробити та впровадити АІС для медичних організацій міста, що забезпечить комплексний підхід до управління медичними установами, включаючи лікарні та поліклініки.

Основні задачі, які повинна вирішувати розроблена АІС:

- 1. Облік структурних підрозділів медичних закладів:** Кожна лікарня складається з одного або декількох корпусів, у кожному з яких розміщаються відділення, що спеціалізуються на лікуванні певної групи хвороб. Кожне відділення має певну кількість палат і ліжок.
- 2. Облік персоналу:** Лікарні та поліклініки обслуговуються лікарським (хірурги, терапевти, невропатологи, окулісти, стоматологи, рентгенологи, гінекологи тощо) і обслуговуючим персоналом (медсестри, санітари, прибиральниці тощо). Лікарський персонал має специфічні характеристики, що враховуються в системі.
- 3. Можливість сумісництва:** Лікарі можуть працювати в одній або декількох лікарнях і поліклініках. Лікарі зі званням доцента або професора можуть консультувати в декількох медичних установах.
- 4. Облік лабораторій:** Лабораторії можуть обслуговувати різні лікарні та поліклініки за наявності договору. Кожна лабораторія має один або кілька профілів досліджень (біохімічні, фізіологічні, хімічні).
- 5. Облік пацієнтів:** Амбулаторне лікування проводиться в поліклініках, стаціонарне - в лікарнях. Система повинна вести персоніфікований облік пацієнтів, історію їхніх хвороб, призначення, операції тощо.
- 6. Запити до інформаційної системи:** Система повинна забезпечувати широкий спектр запитів щодо обліку лікарів, обслуговуючого персоналу, пацієнтів, палат, кабінетів, продуктивності лікарів та лабораторій.

Види запитів в інформаційній системі:

1. Отримати перелік і загальне число лікарів зазначеного профілю для конкретного медичного закладу, лікарні або всіх медичних установ міста.
2. Отримати перелік і загальне число обслуговуючого персоналу зазначеної спеціальності для конкретного медичного закладу, лікарні або всіх медичних установ міста.
3. Отримати перелік і загальне число лікарів зазначеного профілю, які зробили число операцій не менш заданого для конкретного медичного закладу, лікарні або всіх медичних установ міста.
4. Отримати перелік і загальне число лікарів зазначеного профілю, стаж роботи яких не менше заданого для конкретного медичного закладу, лікарні або всіх медичних установ міста.
5. Отримати перелік і загальне число лікарів зазначеного профілю зі ступенем кандидата або доктора медичних наук, зі званням доцента або професора для конкретного медичного закладу, або лікарні або всіх медичних установ міста.
6. Отримати перелік пацієнтів зазначененої лікарні, відділення, або конкретної палати зазначеного відділення, із зазначенням дати надходження, стану, температури, лікуючого лікаря.
7. Отримати перелік пацієнтів, які пройшли стаціонарне лікування у зазначеній лікарні, або у конкретного лікаря за деякий проміжок часу.
8. Отримати перелік пацієнтів, що спостерігаються в лікаря зазначеного профілю в конкретній поліклініці.
9. Отримати загальне число палат, ліжок зазначененої лікарні загалом і по кожному відділенню, а також число вільних ліжок у кожному відділенні і число повністю вільних палат.
10. Отримати загальне число кабінетів зазначененої поліклініки, число відвідувань кожного кабінету за певний період.
11. Отримати дані про продуктивність (середнє число прийнятих пацієнтів за день) за вказаний період для конкретного лікаря, або всіх лікарів поліклініки, або для всіх лікарів названого профілю.
12. Отримати дані про завантаження (число пацієнтів, у яких лікар в даний час є лікуючим лікарем) для вказаного лікаря, або всіх лікарів лікарні, або для всіх лікарів названого профілю.
13. Отримати перелік пацієнтів, які перенесли операції у зазначеній лікарні, або поліклініці, або у конкретного лікаря за деякий проміжок часу.
14. Отримати дані про продуктивність лабораторії (середнє число проведених обстежень в день) за вказаний період для даного медичного закладу або всіх медичних установ міста.

## **2. АРХІТЕКТУРА ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ БД**

### **2.1. Аналіз функціонування та організаційні основи підприємства**

Медичні організації міста, зокрема лікарні та поліклініки, функціонують як складні структури, що надають широкий спектр медичних послуг. Їхня діяльність охоплює такі основні аспекти:

- **Структурна організація:** Кожна лікарня складається з одного або декількох корпусів, що включають різні відділення (наприклад, хірургічне, терапевтичне, неврологічне тощо). Кожне відділення має певну кількість палат та ліжок для пацієнтів.
- **Персонал:** Медичний персонал поділяється на лікарський (лікарі різних спеціальностей) та обслуговуючий (медсестри, санітари, технічний персонал). Лікарі можуть мати різні спеціалізації та кваліфікації, що враховується в системі управління.
- **Лабораторії:** Лабораторії проводять різні медичні аналізи і можуть обслуговувати декілька медичних закладів на основі договорів. Вони спеціалізуються на біохімічних, фізіологічних або хімічних дослідженнях.
- **Пацієнти:** Облік пацієнтів ведеться як в амбулаторних умовах (поліклініки), так і в стаціонарних (лікарні). Кожен пацієнт має персоніфікований облік, який включає історію хвороб, призначення та результати лікування.
- **Сумісництво та консультативні послуги:** Лікарі можуть працювати в різних медичних закладах одночасно та консультувати у кількох лікарнях або поліклініках.

## 2.2. Проектування структури баз даних

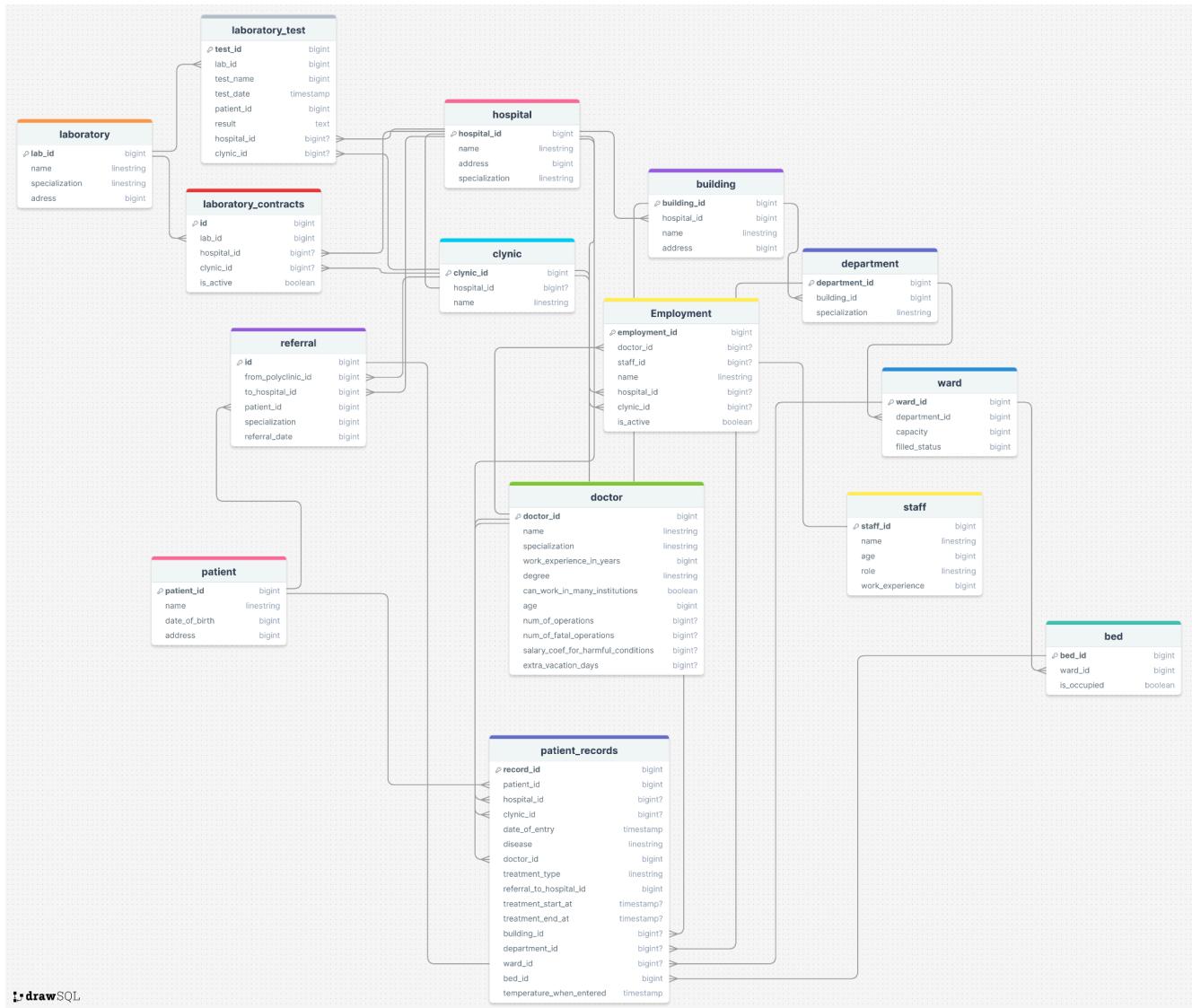


Рис. 2.1. EER-діаграма бази даних

### Таблиця hospital (Лікарні)

- hospital\_id: унікальний ідентифікатор лікарні.
- name: назва лікарні.
- address: адреса лікарні.
- specialization: спеціалізація лікарні.

## **Таблиця building (Корпуси)**

- building\_id: унікальний ідентифікатор корпусу.
- hospital\_id: ідентифікатор лікарні, до якої належить корпус (зовнішній ключ).
- name: назва корпусу.
- address: адреса корпусу.

## **Таблиця department (Відділення)**

- department\_id: унікальний ідентифікатор відділення.
- building\_id: ідентифікатор корпусу, до якого належить відділення (зовнішній ключ).
- name: назва відділення.
- specialization: спеціалізація відділення.

## **Таблиця ward (Палати)**

- ward\_id: унікальний ідентифікатор палати.
- department\_id: ідентифікатор відділення, до якого належить палата (зовнішній ключ).
- capacity: кількість ліжок у палаті.
- filled\_status: кількість зайнятих ліжок у палаті.

## **Таблиця bed (Ліжка)**

- bed\_id: унікальний ідентифікатор ліжка.
- ward\_id: ідентифікатор палати, до якої належить ліжко (зовнішній ключ).
- is\_occupied: статус занятості ліжка (TRUE - зайнято, FALSE - вільне).

## **Таблиця clinic (Поліклініки)**

- clinic\_id: унікальний ідентифікатор поліклініки.

- hospital\_id: ідентифікатор лікарні, до якої належить поліклініка (зовнішній ключ).
- name: назва поліклініки.

### **Таблиця doctor (Лікарі)**

- doctor\_id: унікальний ідентифікатор лікаря.
- name: ім'я лікаря.
- specialization: спеціалізація лікаря.
- work\_experience\_in\_years: стаж роботи лікаря.
- degree: науковий ступінь лікаря.
- can\_work\_in\_many\_institutions: можливість працювати в кількох установах (TRUE - так, FALSE - ні).
- age: вік лікаря.
- num\_of\_operations: кількість проведених операцій.
- num\_of\_fatal\_operations: кількість операцій з летальним результатом.
- salary\_coef\_for\_harmful\_conditions: коефіцієнт до зарплати за шкідливі умови праці.
- extra\_vacation\_days: додаткові дні відпустки.

### **Таблиця staff (Персонал)**

- staff\_id: унікальний ідентифікатор персоналу.
- name: ім'я персоналу.
- age: вік персоналу.
- role: роль (посада) персоналу.
- work\_experience: стаж роботи.

### **Таблиця employment (Зайнятість)**

- employment\_id: унікальний ідентифікатор занятості.
- staff\_id: ідентифікатор персоналу (зовнішній ключ).

- doctor\_id: ідентифікатор лікаря (зовнішній ключ).
- name: назва зайнятості.
- hospital\_id: ідентифікатор лікарні (зовнішній ключ).
- clinic\_id: ідентифікатор поліклініки (зовнішній ключ).
- is\_active: статус зайнятості (TRUE - активна, FALSE - неактивна).

### **Таблиця patient (Пацієнти)**

- patient\_id: унікальний ідентифікатор пацієнта.
- name: ім'я пацієнта.
- date\_of\_birth: дата народження пацієнта.
- address: адреса пацієнта.

### **Таблиця laboratory (Лабораторії)**

- lab\_id: унікальний ідентифікатор лабораторії.
- name: назва лабораторії.
- specialization: спеціалізація лабораторії.
- address: адреса лабораторії.

### **Таблиця laboratory\_test (Лабораторні тести)**

- test\_id: унікальний ідентифікатор тесту.
- lab\_id: ідентифікатор лабораторії (зовнішній ключ).
- test\_name: назва тесту.
- test\_date: дата проведення тесту.
- patient\_id: ідентифікатор пацієнта (зовнішній ключ).
- hospital\_id: ідентифікатор лікарні (зовнішній ключ).
- clinic\_id: ідентифікатор поліклініки (зовнішній ключ).

### **Таблиця laboratory\_contracts (Контракти лабораторії)**

- lab\_id: ідентифікатор лабораторії (зовнішній ключ).

- hospital\_id: ідентифікатор лікарні (зовнішній ключ).
- clinic\_id: ідентифікатор поліклініки (зовнішній ключ).
- is\_active: статус контракту (TRUE - активний, FALSE - неактивний).
- **Первинний ключ (PRIMARY KEY)**: комбінація lab\_id, hospital\_id, clinic\_id.

### **Таблиця referral (Направлення)**

- id: унікальний ідентифікатор направлення.
- from\_polyclinic\_id: ідентифікатор поліклініки, що направляє (зовнішній ключ).
- to\_hospital\_id: ідентифікатор лікарні, що приймає (зовнішній ключ).
- patient\_id: ідентифікатор пацієнта (зовнішній ключ).
- specialization: спеціалізація направлення.
- referral\_date: дата направлення.

### **Таблиця patient\_records (Записи пацієнтів)**

- record\_id: унікальний ідентифікатор запису.
- patient\_id: ідентифікатор пацієнта (зовнішній ключ).
- hospital\_id: ідентифікатор лікарні (зовнішній ключ).
- clinic\_id: ідентифікатор поліклініки (зовнішній ключ).
- date\_of\_entry: дата надходження.
- disease: діагноз.
- doctor\_id: ідентифікатор лікаря (зовнішній ключ).
- treatment\_type: тип лікування.
- cabinet\_id: ідентифікатор кабінету (зовнішній ключ).
- referral\_to\_hospital\_id: ідентифікатор направлення до лікарні (зовнішній ключ).
- treatment\_start\_at: дата початку лікування.
- treatment\_end\_at: дата завершення лікування.
- building\_id: ідентифікатор корпусу (зовнішній ключ).

- department\_id: ідентифікатор відділення (зовнішній ключ).
- ward\_id: ідентифікатор палати (зовнішній ключ).
- bed\_id: ідентифікатор ліжка (зовнішній ключ).
- temperature\_when\_entered: температура при надходженні.

### **Таблиця cabinets (Кабінети клініки)**

- cabinet\_id: унікальний ідентифікатор кабінету.
- clinic\_id: ідентифікатор поліклініки (зовнішній ключ).

## **2.3. Життєві цикли БД**

Життєвий цикл бази даних (БД) включає кілька етапів:

- Аналіз вимог: Визначення вимог користувачів до системи, обсягів даних, які необхідно зберігати, та типів запитів, які будуть виконуватись.
- Проектування: Створення концептуальної, логічної та фізичної моделі бази даних. Визначення основних сутностей, атрибутів та зв'язків між ними.
- Реалізація: Створення структури бази даних у СУБД (системі управління базами даних). Налаштування таблиць, індексів, зв'язків та тригерів.
- Заповнення даними: Внесення початкових даних до бази. Можливе використання інструментів міграції даних або імпорту з інших джерел.
- Експлуатація: Використання бази даних користувачами. Виконання запитів, обробка даних та генерування звітів.

- Обслуговування та підтримка: Забезпечення безперервної роботи бази даних. Включає резервне копіювання, оптимізацію продуктивності та оновлення структури даних.
- Архівування та видалення даних: Переміщення старих або неактуальних даних до архіву або їхнє видалення з бази даних для зменшення обсягу та підвищення продуктивності.
- Закриття бази даних: Проводиться у випадку припинення використання бази даних. Включає збереження резервних копій та остаточне видалення даних.

## **3. РЕАЛІЗАЦІЯ ПРОГРАМНОЇ ВЗАЄМОДІЇ З БД**

### **3.1. Керівництво користувача**

Для того, щоб користуватися даним сервером баз даних, необхідно встановити MySQL і продовжувати роботу з ним через консоль, вводячи команди з лістингу програму, наданому у додатку. Нижче наведено процес встановлення та запуску MySQL на різних операційних системах.

#### **Встановлення та запуск MySQL на Windows**

##### **1. Завантаження MySQL Installer:**

- Відвідайте офіційний вебсайт MySQL (<https://dev.mysql.com/downloads/installer/>).
- Завантажте MySQL Installer.

##### **2. Встановлення MySQL:**

- Запустіть MySQL Installer.
- Виберіть "Custom" або "Developer Default" для встановлення потрібних компонентів.
- Натисніть "Next" і слідуйте інструкціям установника.
- На етапі конфігурації задайте root пароль та створіть користувачів, якщо потрібно.

##### **3. Запуск MySQL:**

- Після завершення установки запустіть MySQL Server з панелі керування або командного рядка (cmd): net start MySQL

##### **4. Доступ до MySQL через консоль:**

- Відкрийте командний рядок (cmd) і введіть команду для запуску MySQL консолі: mysql -u root -p
- Введіть пароль користувача root.

## **Встановлення та запуск MySQL на macOS**

### **1. Завантаження MySQL:**

- Відвідайте офіційний вебсайт MySQL  
(<https://dev.mysql.com/downloads/mysql/>).
- Завантажте MySQL DMG Archive для macOS.

### **2. Встановлення MySQL:**

- Відкрийте завантажений DMG файл і запустіть MySQL Installer.
- Слідуйте інструкціям для встановлення MySQL.
- На етапі конфігурації задайте root пароль.

### **3. Запуск MySQL:**

- Відкрийте "System Preferences" і знайдіть "MySQL".
- Виберіть "Start MySQL Server".

### **4. Доступ до MySQL через консоль:**

- Відкрийте Terminal і введіть команду для запуску MySQL консолі:  
`mysql -u root -p`
- Введіть пароль користувача root.

## **Встановлення та запуск MySQL на Linux (Ubuntu)**

### **1. Оновлення пакетів та встановлення MySQL:**

- Відкрийте Terminal і виконайте команду для оновлення пакетів: `sudo apt update`
- Встановіть MySQL Server: `sudo apt install mysql-server`

### **2. Конфігурація MySQL:**

- Запустіть MySQL Secure Installation для початкової конфігурації: `sudo mysql_secure_installation`
- Введіть root пароль та налаштуйте безпеку (вимкнення анонімних користувачів, відключення віддаленого доступу root, видалення тестової бази даних, перезавантаження привілеїв).

### **3. Запуск MySQL:**

- Запустіть MySQL Server (якщо він ще не запущений): `sudo systemctl start mysql`
- Переконайтесь, що MySQL Server запускається при завантаженні системи: `sudo systemctl enable mysql`

#### 4. Доступ до MySQL через консоль:

- Відкрийте Terminal і введіть команду для запуску MySQL консолі: `sudo mysql -u root -p`
- Введіть пароль користувача root.

## 3.2. Реалізація механізмів SQL

### 3.2.1. SQL-запити

Всі SQL-запити, які описані в розділі 1 цієї курсової роботи виконані у вигляді процедур, так як вони є легкими у створенні та користуванні.

Реалізація процедур у вигляді SQL-коду надана у додатках.

### 3.2.2. Процедури та функції

Під час проектування бази даних було створено процедури, які відповідають різним завданням, визначеним у постановці задачі:

- `get_doctors_by_specialization`: Приймає 3 параметри: профіль лікаря, ідентифікатор установи, та тип установи (лікарня або поліклініка). Повертає таблицю з переліком лікарів зазначеного профілю для конкретного медичного закладу, лікарні або всіх медичних установ міста. Включає підсумковий рядок з загальною кількістю лікарів.
- `get_staff_by_role`: Приймає 3 параметри: роль обслуговуючого персоналу, ідентифікатор установи, та тип установи (лікарня або поліклініка). Повертає таблицю з переліком обслуговуючого персоналу зазначеної спеціальності для конкретного медичного

закладу, лікарні або всіх медичних установ міста. Включає підсумковий рядок з загальною кількістю персоналу.

- `get_doctors_by_operations`: Приймає 4 параметри: профіль лікаря, мінімальну кількість операцій, ідентифікатор установи, та тип установи (лікарня або поліклініка). Повертає таблицю з переліком лікарів зазначеного профілю, які зробили кількість операцій не менше заданого для конкретного медичного закладу, лікарні або всіх медичних установ міста. Включає підсумковий рядок з загальною кількістю лікарів.
- `get_doctors_by_experience`: Приймає 4 параметри: профіль лікаря, мінімальний стаж роботи, ідентифікатор установи, та тип установи (лікарня або поліклініка). Повертає таблицю з переліком лікарів зазначеного профілю, стаж роботи яких не менше заданого для конкретного медичного закладу, лікарні або всіх медичних установ міста. Включає підсумковий рядок з загальною кількістю лікарів.
- `get_doctors_by_degree`: Приймає 4 параметри: профіль лікаря, науковий ступінь, ідентифікатор установи, та тип установи (лікарня або поліклініка). Повертає таблицю з переліком лікарів зазначеного профілю зі ступенем кандидата або доктора медичних наук для конкретного медичного закладу, лікарні або всіх медичних установ міста. Включає підсумковий рядок з загальною кількістю лікарів.
- `get_patients_by_location`: Приймає 3 параметри: ідентифікатор лікарні, ідентифікатор відділення, та ідентифікатор палати. Повертає таблицю з переліком пацієнтів зазначеної лікарні, відділення, або конкретної палати із зазначенням дати надходження, стану, температури та лікуючого лікаря.

- `get_inpatients_by_hospital_or_doctor`: Приймає 4 параметри: ідентифікатор лікарні, ідентифікатор лікаря, дату початку та закінчення періоду. Повертає таблицю з переліком пацієнтів, які пройшли стаціонарне лікування у зазначеній лікарні або у конкретного лікаря за деякий проміжок часу.
- `get_patients_by_doctor_specialization_and_clinic`: Приймає 2 параметри: профіль лікаря та ідентифікатор поліклініки. Повертає таблицю з переліком пацієнтів, що спостерігаються в лікаря зазначеного профілю в конкретній поліклініці.
- `get_wards_and_beds_info`: Приймає 1 параметр: ідентифікатор лікарні. Повертає загальне число палат і ліжок зазначеної лікарні загалом і по кожному відділенню, а також число вільних ліжок у кожному відділенні і число повністю вільних палат.
- `get_clinic_cabinets_info`: Приймає 3 параметри: ідентифікатор поліклініки, дату початку та кінця періоду. Повертає загальне число кабінетів зазначеної поліклініки, число відвідувань кожного кабінету за певний період.
- `get_doctor_performance`: Приймає 5 параметрів: ідентифікатор лікаря, ідентифікатор поліклініки, профіль лікаря, дату початку та кінця періоду. Повертає дані про продуктивність (середнє число прийнятих пацієнтів за день) за вказаний період для конкретного лікаря, або всіх лікарів поліклініки, або для всіх лікарів названого профілю.
- `get_doctor_workload_by_date`: Приймає 4 параметри: цільова дата, ідентифікатор лікаря, ідентифікатор лікарні, та профіль лікаря.

Повертає дані про завантаження (число пацієнтів, у яких лікар в даний час є лікуючим лікарем) для вказаного лікаря, або всіх лікарів лікарні, або для всіх лікарів названого профілю на задану дату.

- `get_patients_who_had_surgery`: Приймає 5 параметрів: ідентифікатор установи, тип установи (лікарня або поліклініка), ідентифікатор лікаря, дату початку та кінця періоду. Повертає таблицю з переліком пацієнтів, які перенесли операції у зазначеній лікарні, або поліклініці, або у конкретного лікаря за деякий проміжок часу.
- `get_laboratory_performance`: Приймає 5 параметрів: ідентифікатор лабораторії, ідентифікатор лікарні, ідентифікатор поліклініки, дату початку та кінця періоду. Повертає дані про продуктивність лабораторії (середнє число проведених обстежень в день) за вказаний період для даного медичного закладу або всіх медичних установ міста.

### 3.2.3. Тригери

Під час створення бази даних тригери не було використано.

## 3.3. Вимоги до апаратних та програмних засобів

Для роботи з базою даних, користувач повинен мати на своєму пристройі відповідне програмне забезпечення. В нашему випадку це MySQL Server.

## 3.4. Випробування розроблених програм

- Отримати перелік і загальне число лікарів зазначеного профілю для конкретного медичного закладу, лікарні або всіх медичних установ міста.

```

mysql> CALL get_doctors_by_specialization('Surgeon', 1, 'hospital');
+-----+-----+
| name | specialization |
+-----+-----+
| Dr. John Smith | Surgeon |
| Dr. Emily Johnson | Surgeon |
| Dr. James Anderson | Surgeon |
| Dr. Christopher Walker | Surgeon |
| TOTAL DOCTORS | 4 |
+-----+
5 rows in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> CALL get_doctors_by_specialization('Surgeon', 1, 'clinic');
+-----+-----+
| name | specialization |
+-----+-----+
| Dr. John Smith | Surgeon |
| TOTAL DOCTORS | 1 |
+-----+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

[mysql> CALL get_doctors_by_specialization('Surgeon', NULL, NULL);
+-----+-----+
| name | specialization |
+-----+-----+
| Dr. John Smith | Surgeon |
| Dr. Emily Johnson | Surgeon |
| Dr. James Anderson | Surgeon |
| Dr. Christopher Walker | Surgeon |
| TOTAL DOCTORS | 4 |
+-----+
5 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

```

рис. 3.4.1

- Отримати перелік і загальне число обслуговуючого персоналу зазначеної спеціальності для конкретного медичного закладу, лікарні або всіх медичних установ міста.

```
mysql> CALL get_staff_by_role('Nurse', 1, 'hospital');
+-----+-----+
| name | role |
+-----+-----+
| John Doe | Nurse |
| Alexander White | Nurse |
| TOTAL STAFF | 2 |
+-----+-----+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_staff_by_role('Nurse', 1, 'clinic');
+-----+-----+
| name | role |
+-----+-----+
| TOTAL STAFF | 0 |
+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

[mysql> CALL get_staff_by_role('Nurse', NULL, NULL);
+-----+-----+
| name | role |
+-----+-----+
| John Doe | Nurse |
| Alexander White | Nurse |
| Andrew Adams | Nurse |
| Ludmila Allen | Nurse |
| Oleg Scott | Nurse |
| TOTAL STAFF | 5 |
+-----+-----+
6 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

рис. 3.4.2

- Отримати перелік і загальне число лікарів зазначеного профілю, які зробили число операцій не менш заданого для конкретного медичного закладу, лікарні або всіх медичних установ міста.

```
mysql> -- Тестування процедури
Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_doctors_by_operations('Surgeon', 100, 1, 'hospital');
+-----+-----+-----+
| name | specialization | num_of_operations |
+-----+-----+-----+
| Dr. John Smith | Surgeon | 300 |
| Dr. Emily Johnson | Surgeon | 250 |
| Dr. James Anderson | Surgeon | 500 |
| Dr. Christopher Walker | Surgeon | 100 |
| TOTAL DOCTORS | 4 | NULL |
+-----+-----+-----+
5 rows in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> CALL get_doctors_by_operations('Surgeon', 100, 1, 'clinic');
+-----+-----+-----+
| name | specialization | num_of_operations |
+-----+-----+-----+
| Dr. John Smith | Surgeon | 300 |
| TOTAL DOCTORS | 1 | NULL |
+-----+-----+-----+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_doctors_by_operations('Surgeon', 100, NULL, NULL);
+-----+-----+-----+
| name | specialization | num_of_operations |
+-----+-----+-----+
| Dr. John Smith | Surgeon | 300 |
| Dr. Emily Johnson | Surgeon | 250 |
| Dr. James Anderson | Surgeon | 500 |
| Dr. Christopher Walker | Surgeon | 100 |
| TOTAL DOCTORS | 4 | NULL |
+-----+-----+-----+
5 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

рис. 3.4.3

- Отримати перелік і загальне число лікарів зазначеного профілю, стаж роботи яких не менше заданого для конкретного медичного закладу, лікарні або всіх медичних установ міста.

```

mysql> -- Тестування процедури
Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_doctors_by_experience('Surgeon', 10, 1, 'hospital');
+-----+-----+-----+
| name | specialization | work_experience_in_years |
+-----+-----+-----+
| Dr. John Smith | Surgeon | 20 |
| Dr. Emily Johnson | Surgeon | 15 |
| Dr. James Anderson | Surgeon | 30 |
| Dr. Christopher Walker | Surgeon | 10 |
| TOTAL DOCTORS | 4 | NULL |
+-----+-----+-----+
5 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_doctors_by_experience('Surgeon', 10, 1, 'clinic');
+-----+-----+-----+
| name | specialization | work_experience_in_years |
+-----+-----+-----+
| Dr. John Smith | Surgeon | 20 |
| TOTAL DOCTORS | 1 | NULL |
+-----+-----+-----+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

[mysql]> CALL get_doctors_by_experience('Surgeon', 10, NULL, NULL);
+-----+-----+-----+
| name | specialization | work_experience_in_years |
+-----+-----+-----+
| Dr. John Smith | Surgeon | 20 |
| Dr. Emily Johnson | Surgeon | 15 |
| Dr. James Anderson | Surgeon | 30 |
| Dr. Christopher Walker | Surgeon | 10 |
| TOTAL DOCTORS | 4 | NULL |
+-----+-----+-----+
5 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

```

рис. 3.4.4

- Отримати перелік і загальне число лікарів зазначеного профілю зі ступенем кандидата або доктора медичних наук, зі званням доцента або професора для конкретного медичного закладу, або лікарні або всіх медичних установ міста.

```

mysql> -- Тестування процедури
Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_doctors_by_degree('Surgeon', 'PhD', 1, 'hospital');
+-----+-----+-----+
| name           | specialization | degree |
+-----+-----+-----+
| Dr. John Smith | Surgeon       | PhD     |
| Dr. James Anderson | Surgeon       | PhD     |
| TOTAL DOCTORS   | 2             | NULL    |
+-----+-----+-----+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_doctors_by_degree('Surgeon', 'MD', 1, 'clinic');
+-----+-----+-----+
| name           | specialization | degree |
+-----+-----+-----+
| TOTAL DOCTORS | 0             | NULL    |
+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

[mysql> CALL get_doctors_by_degree('Surgeon', 'PhD', NULL, NULL);
+-----+-----+-----+
| name           | specialization | degree |
+-----+-----+-----+
| Dr. John Smith | Surgeon       | PhD     |
| Dr. James Anderson | Surgeon       | PhD     |
| TOTAL DOCTORS   | 2             | NULL    |
+-----+-----+-----+
3 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

```

рис. 3.4.5

- Отримати перелік пацієнтів зазначеної лікарні, відділення, або конкретної палати зазначеного відділення, із зазначенням дати надходження, стану, температури, лікуючого лікаря.

```

mysql> -- Тестування процедури
Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_patients_by_location(1, NULL, NULL); -- Отримати всіх пацієнтів лікарні з ID 1
+-----+-----+-----+-----+-----+
| patient_name | date_of_entry | patient_condition | temperature_when_entered | doctor_name |
+-----+-----+-----+-----+
| John Doe     | 2024-03-01   | Heart Disease      | 37.5          | Dr. John Smith |
| Robert Johnson | 2024-03-10   | Surgical Procedure | 37.8          | Dr. Michael Brown |
| Michael Davis | 2024-03-20   | Pediatric Treatment | 37.2          | Dr. William Wilson |
| James Martinez | 2024-03-30   | Gynecological Treatment | 37.7          | Dr. James Anderson |
| David Wilson | 2024-04-05   | Gastroenterological Treatment | 37.4          | Dr. Robert Thomas |
+-----+-----+-----+-----+
5 rows in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> CALL get_patients_by_location(1, 3, NULL); -- Отримати всіх пацієнтів відділення з ID 2 у лікарні з ID 1
+-----+-----+-----+-----+
| patient_name | date_of_entry | patient_condition | temperature_when_entered | doctor_name |
+-----+-----+-----+-----+
| Robert Johnson | 2024-03-10   | Surgical Procedure | 37.8          | Dr. Michael Brown |
| James Martinez | 2024-03-30   | Gynecological Treatment | 37.7          | Dr. James Anderson |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_patients_by_location(2, 4, 7); -- Отримати всіх пацієнтів палати з ID 3 у відділенні з ID 2 у лікарні з ID 1
+-----+-----+-----+-----+
| patient_name | date_of_entry | patient_condition | temperature_when_entered | doctor_name |
+-----+-----+-----+-----+
| Linda Brown | 2024-03-15   | Orthopedic Surgery | 36.9          | Dr. Sarah Davis |
| Sarah Rodriguez | 2024-04-01   | Dermatological Treatment | 36.8          | Dr. Barbara Taylor |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

```

рис. 3.4.6

- Отримати перелік пацієнтів, які пройшли стаціонарне лікування у зазначеній лікарні, або у конкретного лікаря за деякий проміжок часу.

```

mysql> -- Тестування процедури
Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_inpatients_by_hospital_or_doctor(1, NULL, '2024-01-01', '2024-12-31'); -- Отримати всіх пацієнтів, які пройшли стаціонарне лікування у лікарні з ID 1 за 2024 рік
+-----+-----+-----+-----+
| patient_name | date_of_entry | treatment_start_at | treatment_end_at | doctor_name |
+-----+-----+-----+-----+
| John Doe     | 2024-03-01   | 2024-03-01   | 2024-03-10   | Dr. John Smith |
| Robert Johnson | 2024-03-10   | 2024-03-10   | 2024-03-28   | Dr. Michael Brown |
| Michael Davis | 2024-03-20   | 2024-03-20   | 2024-03-30   | Dr. William Wilson |
| James Martinez | 2024-03-30   | 2024-03-30   | 2024-04-10   | Dr. James Anderson |
| David Wilson | 2024-04-05   | 2024-04-05   | 2024-04-20   | Dr. Robert Thomas |
+-----+-----+-----+-----+
5 rows in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> CALL get_inpatients_by_hospital_or_doctor(NULL, 1, '2024-01-01', '2024-12-31'); -- Отримати всіх пацієнтів, які пройшли стаціонарне лікування у лікарні з ID 1 за 2024 рік
+-----+-----+-----+-----+
| patient_name | date_of_entry | treatment_start_at | treatment_end_at | doctor_name |
+-----+-----+-----+-----+
| John Doe     | 2024-03-01   | 2024-03-01   | 2024-03-10   | Dr. John Smith |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_inpatients_by_hospital_or_doctor(1, 1, '2024-01-01', '2024-12-31'); -- Отримати всіх пацієнтів, які пройшли стаціонарне лікування у лікарні з ID 1 у лікарні з ID 1 за 2024 рік
+-----+-----+-----+-----+
| patient_name | date_of_entry | treatment_start_at | treatment_end_at | doctor_name |
+-----+-----+-----+-----+
| John Doe     | 2024-03-01   | 2024-03-01   | 2024-03-10   | Dr. John Smith |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

```

рис. 3.4.7

- Отримати перелік пацієнтів, що спостерігаються в лікарні зазначеного профілю в конкретній поліклініці.

```

mysql> -- Тестування процедури
Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_patients_by_doctor_specialization_and_clinic('Surgeon', 2); -- Отримати всіх пацієнтів, що спостерігаються у хірурга у поліклініці з ID 1
+-----+-----+-----+-----+
| patient_name | doctor_name | specialization | date_of_entry | treatment_type |
+-----+-----+-----+-----+
| Jane Smith   | Dr. Christopher Walker | Surgeon       | 2024-03-05    | Medication     |
| James Martinez | Dr. Christopher Walker | Surgeon       | 2024-03-30    | Medication     |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_patients_by_doctor_specialization_and_clinic('Neurologist', 2); -- Отримати всіх пацієнтів, що спостерігаються у невролог у поліклініці з ID 2
Empty set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> 

```

рис. 3.4.8

- Отримати загальне число палат, ліжок зазначеної лікарні загалом і по кожному відділенню, а також число вільних ліжок у кожному відділенні і число повністю вільних палат.

```

mysql> CALL get_wards_and_beds_info(1); -- Отримати інформацію про палати та ліжка у лікарні з ID 1
+-----+-----+
| total_wards | total_beds |
+-----+-----+
|       6      |      120     |
+-----+-----+
1 row in set (0.01 sec)

+-----+-----+-----+
| department_name | total_wards | total_beds |
+-----+-----+-----+
| General Medicine |        2 |      60 |
| Cardiology       |        2 |      40 |
| Surgery          |        2 |      20 |
+-----+-----+-----+
3 rows in set (0.01 sec)

+-----+-----+
| department_name | free_beds |
+-----+-----+
| General Medicine |      25 |
| Cardiology       |      15 |
| Surgery          |       6 |
+-----+-----+
3 rows in set (0.01 sec)

Empty set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

[mysql> CALL get_wards_and_beds_info(2); -- Отримати інформацію про палати та ліжка у лікарні з ID 2
+-----+-----+
| total_wards | total_beds |
+-----+-----+
|       4      |      80      |
+-----+-----+
1 row in set (0.00 sec)

+-----+-----+-----+
| department_name | total_wards | total_beds |
+-----+-----+-----+
| Emergency      |        2 |      50 |
| Trauma          |        2 |      30 |
+-----+-----+-----+
2 rows in set (0.00 sec)

+-----+-----+
| department_name | free_beds |
+-----+-----+
| Emergency      |      20 |
| Trauma          |       8 |
+-----+-----+
2 rows in set (0.00 sec)

Empty set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

```

рис. 3.4.9

- Отримати загальне число кабінетів зазначеної поліклініки, число відвідувань кожного кабінету за певний період.

```
mysql> CALL get_clinic_cabinets_info(1, '2024-01-01', '2024-12-31'); -- Отримати інформацію про кабінети та відвідування у поліклініці з ID 1 за
+-----+
| total_cabinets |
+-----+
|          4 |
+-----+
1 row in set (0.00 sec)

+-----+-----+
| cabinet_id | visit_count |
+-----+-----+
|      1 |        1 |
|      2 |        1 |
|      3 |        1 |
|      4 |        1 |
+-----+-----+
4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_clinic_cabinets_info(2, '2024-01-01', '2024-12-31'); -- Отримати інформацію про кабінети та відвідування у поліклініці з ID 2 за
+-----+
| total_cabinets |
+-----+
|          4 |
+-----+
1 row in set (0.00 sec)

+-----+-----+
| cabinet_id | visit_count |
+-----+-----+
|      5 |        1 |
|      6 |        1 |
|      7 |        1 |
|      8 |        1 |
+-----+-----+
4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

рис. 3.4.10

- Отримати дані про продуктивність (середнє число прийнятих пацієнтів за день) за вказаний період для конкретного лікаря, або всіх лікарів поліклініки, або для всіх лікарів названого профілю.

```
mysql> -- Тестування процедури з періодом 2024-03-01 до 2024-03-30
Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_doctor_performance(1, NULL, NULL, '2024-03-01', '2024-03-30'); -- Продуктивність для лікаря з ID 1 за березень 2024 року
+-----+
| doctor_name | average_patients_per_day |
+-----+
| Dr. John Smith |          0.0345 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_doctor_performance(NULL, 1, NULL, '2024-03-01', '2024-03-30'); -- Продуктивність для всіх лікарів у поліклініці з ID 1 за березень 2
+-----+
| doctor_name | average_patients_per_day |
+-----+
| Dr. Charles Harris |          0.0690 |
| Dr. Karen Lee |          0.0345 |
+-----+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_doctor_performance(NULL, NULL, 'Surgeon', '2024-03-01', '2024-03-30'); -- Продуктивність для всіх хірургів за березень 2024 року
+-----+
| doctor_name | average_patients_per_day |
+-----+
| Dr. John Smith |          0.0345 |
| Dr. Emily Johnson |          0.0345 |
| Dr. James Anderson |          0.0345 |
| Dr. Christopher Walker |          0.0690 |
+-----+
4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

рис. 3.4.11

- Отримати дані про завантаження (число пацієнтів, у яких лікар в даний час є лікуючим лікарем) для вказаного лікаря, або всіх лікарів лікарні, або для всіх лікарів названого профілю.

```

mysql> -- Тестування процедури
Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_doctor_workload_by_date('2024-03-15', 1, NULL, NULL); -- Завантаження для лікаря з ID 1 на 15 березня 2024 року
Empty set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_doctor_workload_by_date('2024-03-15', NULL, 1, NULL); -- Завантаження для всіх лікарів у лікарні з ID 1 на 15 березня 2024 року
+-----+-----+
| doctor_name | patient_count |
+-----+-----+
| Dr. Michael Brown | 1 |
| Dr. William Wilson | 1 |
| Dr. James Anderson | 1 |
| Dr. Robert Thomas | 1 |
+-----+-----+
4 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_doctor_workload_by_date('2024-03-15', NULL, NULL, 'Surgeon'); -- Завантаження для всіх хірургів на 15 березня 2024 року
+-----+-----+
| doctor_name | patient_count |
+-----+-----+
| Dr. James Anderson | 1 |
| Dr. Christopher Walker | 1 |
+-----+-----+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

```

рис. 3.4.12

- Отримати перелік пацієнтів, які перенесли операції у зазначеній лікарні, або поліклініці, або у конкретного лікаря за деякий проміжок часу.

```

mysql> -- Тестування процедури
Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_patients_who_had_surgery(1, 'hospital', NULL, '2024-01-01', '2024-03-30'); -- Перелік пацієнтів, які перенесли операції у лікарні з ID 1 за період 01.01.2024 – 30.03.2024
+-----+-----+-----+-----+
| patient_name | date_of_entry | treatment_type | doctor_name |
+-----+-----+-----+-----+
| John Doe | 2024-03-01 | Surgery | Dr. John Smith |
| Robert Johnson | 2024-03-10 | Surgery | Dr. Michael Brown |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_patients_who_had_surgery(1, 'clinic', NULL, '2024-01-01', '2024-03-30'); -- Перелік пацієнтів, які перенесли операції у поліклініці з ID 1 за період 01.01.2024 – 30.03.2024
Empty set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_patients_who_had_surgery(NULL, NULL, 1, '2024-01-01', '2024-03-30'); -- Перелік пацієнтів, які перенесли операції у лікаря з ID 1 за період 01.01.2024 – 30.03.2024
+-----+-----+-----+-----+
| patient_name | date_of_entry | treatment_type | doctor_name |
+-----+-----+-----+-----+
| John Doe | 2024-03-01 | Surgery | Dr. John Smith |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

```

рис. 3.4.13

- Отримати дані про продуктивність лабораторії (середнє число проведених обстежень в день) за вказаний період для даного медичного закладу або всіх медичних установ міста.

```

mysql> -- Тестування процедури
Query OK, 0 rows affected (0.01 sec)

mysql> CALL get_laboratory_performance(1, NULL, NULL, '2024-03-01', '2024-03-30'); -- Продуктивність для лабораторії з ID 1 за період 01.03.2024 – 30.03.2024
+-----+-----+
| laboratory_name | average_tests_per_day |
+-----+-----+
| BioLab          |      0.0345 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_laboratory_performance(NULL, 1, NULL, '2024-03-01', '2024-03-30'); -- Продуктивність для всіх лабораторій у лікарні з ID 1 за період 01.03.2024 – 30.03.2024
+-----+-----+
| laboratory_name | average_tests_per_day |
+-----+-----+
| Physiolab       |      0.0345 |
| Generallab      |      0.0345 |
+-----+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_laboratory_performance(NULL, NULL, 1, '2024-03-01', '2024-03-30'); -- Продуктивність для всіх лабораторій у поліклініці з ID 1 за період 01.03.2024 – 30.03.2024
+-----+-----+
| laboratory_name | average_tests_per_day |
+-----+-----+
| ChemLab         |      0.0345 |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL get_laboratory_performance(NULL, NULL, NULL, '2024-03-01', '2024-03-30'); -- Продуктивність для всіх лабораторій у місті за період 01.03.2024 – 30.03.2024
+-----+-----+
| laboratory_name | average_tests_per_day |
+-----+-----+
| Physiolab       |      0.0345 |
| ChemLab         |      0.0345 |
| AdvancedLab    |      0.0345 |
| Generallab      |      0.0345 |
| BioLab          |      0.0345 |
+-----+
5 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>

```

рис. 3.4.14

### 3.5. Опис тестової бази даних

Під час виконання курсової роботи було створено 16 таблиць, розглянемо інформацію, що міститься в них:

- hospital

```

mysql> select * from hospital;
+-----+-----+-----+
| hospital_id | name           | address        | specialization |
+-----+-----+-----+
| 1            | Central City Hospital | 123 Main St   | General        |
| 2            | Regional Hospital    | 456 Regional Rd | Trauma and Emergency |
+-----+-----+-----+
2 rows in set (0.00 sec)

```

рис. 3.5.1

- building

```
[mysql]> select * from building;
+-----+-----+-----+-----+
| building_id | hospital_id | name | address |
+-----+-----+-----+-----+
| 1 | 1 | Main Building | 123 Main St |
| 2 | 1 | Surgery Building | 125 Main St |
| 3 | 2 | Emergency Building | 456 Regional Rd |
| 4 | 2 | Trauma Building | 458 Regional Rd |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

рис. 3.5.2

- department

```
[mysql]> select * from department;
+-----+-----+-----+-----+
| department_id | building_id | name | specialization |
+-----+-----+-----+-----+
| 1 | 1 | General Medicine | General Treatment |
| 2 | 1 | Cardiology | Heart Diseases |
| 3 | 2 | Surgery | Surgeries |
| 4 | 3 | Emergency | Emergency Treatment |
| 5 | 4 | Trauma | Trauma Treatment |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

рис 3.5.3

- ward

```
[mysql]> select * from ward;
+-----+-----+-----+-----+
| ward_id | department_id | capacity | filled_status |
+-----+-----+-----+-----+
| 1 | 1 | 30 | 20 |
| 2 | 1 | 30 | 15 |
| 3 | 2 | 20 | 15 |
| 4 | 2 | 20 | 10 |
| 5 | 3 | 10 | 8 |
| 6 | 3 | 10 | 6 |
| 7 | 4 | 25 | 18 |
| 8 | 4 | 25 | 12 |
| 9 | 5 | 15 | 12 |
| 10 | 5 | 15 | 10 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

рис. 3.5.4

- bed

```
[mysql] > select * from bed;
```

bed_id	ward_id	is_occupied
1	1	1
2	1	0
3	1	1
4	1	0
5	1	1
6	1	0
7	2	1
8	2	0
9	2	1
10	2	0
11	2	1
12	2	0
13	3	1
14	3	0
15	3	1
16	3	0
17	3	1
18	3	0
19	4	1
20	4	0
21	4	1
22	4	0
23	4	1
24	4	0
25	5	1
26	5	0
27	5	1
28	5	0
29	5	1
30	5	0
31	6	1
32	6	0
33	6	1

33	6	1
34	6	0
35	6	1
36	6	0
37	7	1
38	7	0
39	7	1
40	7	0
41	7	1
42	7	0
43	8	1
44	8	0
45	8	1
46	8	0
47	8	1
48	8	0
49	9	1
50	9	0
51	9	1
52	9	0
53	9	1
54	9	0
55	10	1
56	10	0
57	10	1
58	10	0
59	10	1
60	10	0

60 rows in set (0.00 sec)

рис 3.5.5

- clinic

```
[mysql] > select * from clinic;
+-----+-----+
| clinic_id | hospital_id | name
+-----+-----+
| 1 | 1 | Downtown Polyclinic |
| 2 | 2 | Eastside Polyclinic |
| 3 | NULL | Independent Polyclinic |
+-----+
3 rows in set (0.01 sec)
```

рис. 3.5.6

- cabinets

```
[mysql] > select * from cabinets;
+-----+-----+
| cabinet_id | clinic_id |
+-----+-----+
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |
| 5 | 2 |
| 6 | 2 |
| 7 | 2 |
| 8 | 2 |
| 9 | 3 |
| 10 | 3 |
| 11 | 3 |
| 12 | 3 |
+-----+
12 rows in set (0.00 sec)
```

рис. 3.5.7

- doctor

```

mysql> select * from doctor;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| doctor_id | name | specialization | work_experience_in_years | degree | can_work_in_many_institutions | age | num_of_operations | num_of_fatal_operations | salary_coeff_for_harmful_cons |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Dr. John Smith | Surgeon | 20 | PhD | 1 | 50 | 300 | 5 | | | |
| NULL | NULL | Surgeon | 15 | MD | 0 | 40 | 250 | 2 |
| NULL | 2 | Dr. Emily Johnson | NULL | Gynecologist | 25 | PhD | 1 | 55 | 400 | 1 |
| NULL | 3 | Dr. Michael Brown | NULL | Dentist | 18 | PhD | 1 | 45 | 200 | 0 |
| NULL | 4 | Dr. Sarah Davis | NULL | NULL | 22 | MD | 0 | 48 | NULL | NULL |
| 1.2 | 5 | Dr. William Wilson | 30 | Dr. Linda Martinez | Neurologist | 17 | MD | 0 | 43 | NULL | NULL |
| NULL | 6 | Dr. James Anderson | 30 | Surgeon | 30 | PhD | 1 | 60 | 500 | 10 |
| NULL | 7 | Dr. Barbara Taylor | NULL | Gynecologist | 12 | MD | 0 | 38 | 150 | 0 |
| NULL | 8 | Dr. Robert Thomas | NULL | Dentist | 14 | MD | 0 | 41 | 180 | 0 |
| 1.2 | 9 | Dr. Patricia Jackson | NULL | Radiologist | 19 | MD | 0 | 44 | NULL | NULL |
| 1.3 | 10 | Dr. Charles Harris | 30 | Neurologist | 16 | MD | 0 | 42 | NULL | NULL |
| NULL | 11 | Dr. Christopher Walker | 30 | Surgeon | 10 | MD | 0 | 35 | 100 | 0 |
| NULL | 12 | Dr. Karen Lee | NULL | Gynecologist | 20 | PhD | 1 | 46 | 300 | 2 |
| NULL | 13 | Dr. Daniel Hall | NULL | Dentist | 11 | MD | 0 | 36 | 120 | 0 |
| 1.2 | 14 | Dr. Jennifer Young | NULL | Radiologist | 23 | MD | 0 | 50 | NULL | NULL |
| 1.3 | 15 | Dr. Vladimir Turner | 30 | NULL | NULL | NULL | NULL | NULL | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
15 rows in set (0.00 sec)

```

рис. 3.5.8

- staff

```

mysql> select * from staff;
+-----+-----+-----+-----+-----+
| staff_id | name | age | role | work_experience |
+-----+-----+-----+-----+-----+
| 1 | John Doe | 45 | Nurse | 20 |
| 2 | Mary Smith | 38 | Orderly | 15 |
| 3 | Helen Brown | 30 | Cleaner | 8 |
| 4 | Alexander White | 50 | Nurse | 25 |
| 5 | Natalie Green | 28 | Orderly | 6 |
| 6 | Victoria Black | 42 | Cleaner | 12 |
| 7 | Andrew Adams | 35 | Nurse | 10 |
| 8 | Svetlana Clark | 47 | Orderly | 22 |
| 9 | Dmitry Hall | 33 | Cleaner | 7 |
| 10 | Ludmila Allen | 41 | Nurse | 18 |
| 11 | Vladimir Young | 29 | Orderly | 5 |
| 12 | Anna King | 36 | Cleaner | 9 |
| 13 | Oleg Scott | 39 | Nurse | 14 |
| 14 | Tatiana Turner | 48 | Orderly | 23 |
| 15 | Michael Carter | 34 | Cleaner | 11 |
+-----+-----+-----+-----+-----+
15 rows in set (0.00 sec)

```

рис. 3.5.9

- employment

mysql> select * from employment;					hospital_id	clinic_id	is_active
employment_id	staff_id	doctor_id	name				
1	1	NULL	John Doe		1	NULL	1
2	2	NULL	Mary Smith		1	NULL	1
3	3	NULL	Helen Brown		1	NULL	1
4	4	NULL	Alexander White		1	NULL	1
5	5	NULL	Natalie Green		1	NULL	1
6	6	NULL	Victoria Black		2	NULL	1
7	7	NULL	Andrew Adams		2	NULL	1
8	8	NULL	Svetlana Clark		2	NULL	1
9	9	NULL	Dmitry Hall		2	NULL	1
10	10	NULL	Ludmila Allen		2	NULL	1
11	11	NULL	Vladimir Young		NULL	1	1
12	12	NULL	Anna King		NULL	1	1
13	13	NULL	Oleg Scott		NULL	2	1
14	14	NULL	Tatiana Turner		NULL	2	1
15	15	NULL	Michael Carter		NULL	3	1
16	NULL	1	Dr. John Smith		1	NULL	1
17	NULL	1	Dr. John Smith		2	NULL	1
18	NULL	1	Dr. John Smith		1	1	1
19	NULL	2	Dr. Emily Johnson		1	NULL	1
20	NULL	3	Dr. Michael Brown		1	NULL	1
21	NULL	3	Dr. Michael Brown		2	NULL	1
22	NULL	3	Dr. Michael Brown		2	2	1
23	NULL	4	Dr. Sarah Davis		1	NULL	1
24	NULL	5	Dr. William Wilson		NULL	1	1
25	NULL	5	Dr. William Wilson		NULL	2	1
26	NULL	5	Dr. William Wilson		1	2	1
27	NULL	6	Dr. Linda Martinez		NULL	2	1
28	NULL	7	Dr. James Anderson		1	NULL	1
29	NULL	7	Dr. James Anderson		2	NULL	1
30	NULL	7	Dr. James Anderson		1	3	1
31	NULL	8	Dr. Barbara Taylor		2	NULL	1
32	NULL	9	Dr. Robert Thomas		NULL	1	1
33	NULL	10	Dr. Patricia Jackson		NULL	2	1
34	NULL	11	Dr. Charles Harris		NULL	2	1
35	NULL	12	Dr. Christopher Walker		1	NULL	1
36	NULL	13	Dr. Karen Lee		1	NULL	1
37	NULL	13	Dr. Karen Lee		2	NULL	1
38	NULL	13	Dr. Karen Lee		1	3	1
39	NULL	14	Dr. Daniel Hall		NULL	1	1
40	NULL	15	Dr. Jennifer Young		NULL	3	1

40 rows in set (0.00 sec)

рис. 3.5.10

- patient

```
[mysql] > select * from patient;
+-----+-----+-----+-----+
| patient_id | name | date_of_birth | address |
+-----+-----+-----+-----+
| 1 | John Doe | 1985-01-15 | 123 Main St |
| 2 | Jane Smith | 1990-02-20 | 456 Elm St |
| 3 | Robert Johnson | 1982-03-25 | 789 Maple St |
| 4 | Linda Brown | 1975-04-30 | 101 Pine St |
| 5 | Michael Davis | 1988-05-05 | 202 Oak St |
| 6 | Emily Garcia | 1992-06-10 | 303 Cedar St |
| 7 | James Martinez | 1979-07-15 | 404 Birch St |
| 8 | Sarah Rodriguez | 1983-08-20 | 505 Spruce St |
| 9 | David Wilson | 1991-09-25 | 606 Walnut St |
| 10 | Laura Moore | 1987-10-30 | 707 Cherry St |
| 11 | Richard Taylor | 1984-11-05 | 808 Chestnut St |
| 12 | Susan Anderson | 1978-12-10 | 909 Hickory St |
| 13 | Charles Thomas | 1981-01-15 | 111 Ash St |
| 14 | Jessica Jackson | 1993-02-20 | 222 Willow St |
| 15 | Christopher White | 1989-03-25 | 333 Poplar St |
| 16 | Karen Harris | 1976-04-30 | 444 Fir St |
| 17 | Matthew Martin | 1990-05-05 | 555 Redwood St |
| 18 | Patricia Thompson | 1985-06-10 | 666 Cypress St |
| 19 | Joshua Lee | 1980-07-15 | 777 Alder St |
| 20 | Elizabeth King | 1994-08-20 | 888 Hemlock St |
| 21 | Andrew Scott | 1986-09-25 | 999 Magnolia St |
| 22 | Barbara Green | 1983-10-30 | 121 Maple St |
| 23 | Daniel Adams | 1991-11-05 | 234 Oak St |
| 24 | Nancy Baker | 1979-12-10 | 345 Cedar St |
| 25 | Paul Gonzalez | 1987-01-15 | 456 Birch St |
| 26 | Sandra Nelson | 1992-02-20 | 567 Spruce St |
| 27 | Steven Carter | 1984-03-25 | 678 Walnut St |
| 28 | Betty Mitchell | 1988-04-30 | 789 Cherry St |
| 29 | George Perez | 1990-05-05 | 890 Chestnut St |
| 30 | Donna Roberts | 1985-06-10 | 901 Hickory St |
+-----+-----+-----+-----+
30 rows in set (0.00 sec)
```

рис. 3.5.11

- referral

```
[mysql] > select * from referral;
+----+-----+-----+-----+-----+-----+
| id | from_polyclinic_id | to_hospital_id | patient_id | specialization | referral_date |
+----+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 1 | Cardiology | 2024-01-10 |
| 2 | 2 | 2 | 2 | Neurology | 2024-01-15 |
| 3 | 3 | 1 | 3 | Surgery | 2024-01-20 |
| 4 | 1 | 2 | 4 | Orthopedics | 2024-01-25 |
| 5 | 2 | 1 | 5 | Pediatrics | 2024-02-01 |
| 6 | 3 | 2 | 6 | Radiology | 2024-02-05 |
| 7 | 1 | 1 | 7 | Gynecology | 2024-02-10 |
| 8 | 2 | 2 | 8 | Dermatology | 2024-02-15 |
| 9 | 3 | 1 | 9 | Gastroenterology | 2024-02-20 |
| 10 | 1 | 2 | 10 | Urology | 2024-02-25 |
+----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

рис. 3.5.12

- patient\_records

рис. 3.5.13

- laboratory

```
[mysql]> select * from laboratory;
+-----+-----+-----+-----+
| lab_id | name      | specialization           | address      |
+-----+-----+-----+-----+
|     1  | BioLab    | Biochemical             | 123 Bio St   |
|     2  | PhysioLab | Physiological          | 456 Physio Rd |
|     3  | ChemLab   | Chemical                | 789 Chem Ave  |
|     4  | AdvancedLab | Biochemical, Physiological | 101 Advanced Blvd |
|     5  | GeneralLab | Biochemical, Chemical   | 202 General Ln |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

рис. 3.5.14

- laboratory\_contracts

```
[mysql] > select * from laboratory_contracts;
+-----+-----+-----+-----+-----+
| contract_id | lab_id | hospital_id | clinic_id | is_active |
+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | NULL | 1 |
| 2 | 1 | 2 | NULL | 1 |
| 3 | 2 | 1 | NULL | 1 |
| 4 | 3 | NULL | 1 | 1 |
| 5 | 3 | NULL | 2 | 1 |
| 6 | 4 | 2 | NULL | 1 |
| 7 | 4 | NULL | 3 | 1 |
| 8 | 5 | 1 | NULL | 1 |
| 9 | 5 | NULL | 3 | 1 |
+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

рис. 3.5.15

- laboratory\_test

```
[mysql] > select * from laboratory_test;
+-----+-----+-----+-----+-----+-----+-----+
| test_id | lab_id | test_name | test_date | patient_id | hospital_id | clinic_id |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 1 | Blood Test | 2024-01-10 | 1 | 1 | NULL |
| 2 | 1 | Urine Test | 2024-01-15 | 2 | 2 | NULL |
| 3 | 2 | Stress Test | 2024-01-20 | 3 | 1 | NULL |
| 4 | 3 | Chemical Analysis | 2024-01-25 | 4 | NULL | 1 |
| 5 | 3 | Toxicology | 2024-02-01 | 5 | NULL | 2 |
| 6 | 4 | Comprehensive Test | 2024-02-05 | 6 | 2 | NULL |
| 7 | 4 | Hormone Test | 2024-02-10 | 7 | NULL | 3 |
| 8 | 5 | Basic Metabolic Panel | 2024-02-15 | 8 | 1 | NULL |
| 9 | 5 | Complete Blood Count | 2024-02-20 | 9 | NULL | 3 |
| 10 | 1 | Blood Test | 2024-02-25 | 10 | 1 | NULL |
| 11 | 2 | Cardiac Stress Test | 2024-03-01 | 11 | 1 | NULL |
| 12 | 3 | Chemical Analysis | 2024-03-05 | 12 | NULL | 1 |
| 13 | 4 | Hormone Test | 2024-03-10 | 13 | NULL | 3 |
| 14 | 5 | Basic Metabolic Panel | 2024-03-15 | 14 | 1 | NULL |
| 15 | 1 | Urine Test | 2024-03-20 | 15 | 2 | NULL |
+-----+-----+-----+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

рис. 3.5.16

## 4. ВИСНОВОК

У даній курсовій роботі було розглянуто процес проєктування та впровадження автоматизованої інформаційної системи (AIC) для медичних організацій міста. Робота охоплює різні аспекти структури медичних закладів, включаючи лікарні та поліклініки, а також управління даними про лікарський та обслуговуючий персонал, пацієнтів та лабораторії.

Основні результати роботи включають:

- 1. Аналіз функціонування медичних організацій:** Було визначено основні організаційні засади медичних закладів, включаючи структуру лікарень, поліклінік, відділень, палат та ліжок, а також персоналу, який працює у цих закладах.
- 2. Проєктування структури бази даних:** Розроблено структуру бази даних, яка включає таблиці для зберігання даних про лікарні, поліклініки, відділення, палати, ліжка, лікарів, персонал, пацієнтів та лабораторії. Визначено зв'язки між таблицями для забезпечення цілісності та узгодженості даних.
- 3. Створення процедур та функцій:** Було розроблено 13 процедур, які відповідають різним завданням медичних організацій. Ці процедури забезпечують ефективний доступ до даних про лікарів, персонал, пацієнтів, палати та лабораторії, дозволяючи отримувати необхідну інформацію для управління медичними закладами.
- 4. Встановлення та запуск MySQL:** Описано процес встановлення та налаштування MySQL на різних операційних системах (Windows, macOS, Linux), що дозволяє користувачам легко розгорнути базу даних та працювати з нею через консоль.

Розроблена інформаційна система забезпечує централізоване управління даними про медичні заклади, їхні структурні підрозділи, персонал та пацієнтів, що сприяє

підвищенню ефективності роботи медичних організацій та якості наданих медичних послуг.

Завдяки впровадженню даної АІС медичні заклади можуть більш ефективно вести облік пацієнтів, планувати роботу персоналу, контролювати стан палат та ліжок, а також аналізувати продуктивність лабораторій та лікарів. Це забезпечує своєчасний доступ до необхідної інформації, знижує адміністративне навантаження та покращує загальне управління медичними організаціями.

## **5. ЛІТЕРАТУРА**

- 1) **MySQL 8.0 Reference Manual.** Available at:  
<https://dev.mysql.com/doc/refman/8.0/en/>
- 2) **Beaulieu, A. (2009).** *Learning SQL*, 2nd Edition. O'Reilly Media.
- 3) **Connolly, T., & Begg, C. (2015).** Database Systems: A Practical Approach to Design, Implementation, and Management. Pearson.
- 4) **Date, C. J. (2003).** *An Introduction to Database Systems*, 8th Edition. Addison-Wesley.
- 5) **Silberschatz, A., Korth, H. F., Sudarshan, S. (2019).** *Database System Concepts*, 7th Edition. McGraw-Hill Education.
- 6) **MySQL Documentation: Stored Procedures and Functions.** Available at:  
<https://dev.mysql.com/doc/refman/8.0/en/stored-routines.html>
- 7) **Stack Overflow.** *Various Q&A on MySQL Optimization and Performance.* Available at: <https://stackoverflow.com/>

# ДОДАТКИ

## Додаток А: Лістинг програми

```
DROP SCHEMA IF EXISTS `iasa`;
CREATE SCHEMA IF NOT EXISTS `iasa` DEFAULT CHARACTER SET utf8mb3 ;
USE `iasa` ;

—CREATING TABLES
-- Лікарні
DROP TABLE IF EXISTS hospital;
CREATE TABLE IF NOT EXISTS hospital (
    hospital_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    address VARCHAR(255),
    specialization VARCHAR(255)
);
-- Корпуси
DROP TABLE IF EXISTS building;
CREATE TABLE IF NOT EXISTS building (
    building_id INT AUTO_INCREMENT PRIMARY KEY,
    hospital_id INT,
    name VARCHAR(255),
    address VARCHAR(255),
    FOREIGN KEY (hospital_id) REFERENCES hospital(hospital_id)
);
-- Відділення
DROP TABLE IF EXISTS department;
CREATE TABLE IF NOT EXISTS department (
    department_id INT AUTO_INCREMENT PRIMARY KEY,
    building_id INT,
    name VARCHAR(255),
    specialization VARCHAR(255),
    FOREIGN KEY (building_id) REFERENCES building(building_id)
);
-- Палати
DROP TABLE IF EXISTS ward;
CREATE TABLE IF NOT EXISTS ward (
    ward_id INT AUTO_INCREMENT PRIMARY KEY,
    department_id INT,
    capacity INT,
    filled_status INT,
    FOREIGN KEY (department_id) REFERENCES department(department_id)
);
-- Ліжка
DROP TABLE IF EXISTS bed;
CREATE TABLE IF NOT EXISTS bed (
    bed_id INT AUTO_INCREMENT PRIMARY KEY,
    ward_id INT,
    is_occupied BOOLEAN,
    FOREIGN KEY (ward_id) REFERENCES ward(ward_id)
);
-- Поліклініки
DROP TABLE IF EXISTS clinic;
CREATE TABLE IF NOT EXISTS clinic (
    clinic_id INT AUTO_INCREMENT PRIMARY KEY,
    hospital_id INT,
    name VARCHAR(255),
    FOREIGN KEY (hospital_id) REFERENCES hospital(hospital_id)
```

```

);
-- Лікарі
DROP TABLE IF EXISTS doctor;
CREATE TABLE IF NOT EXISTS doctor (
    doctor_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    specialization VARCHAR(255),
    work_experience_in_years INT,
    degree VARCHAR(255),
    can_work_in_many_institutions BOOLEAN,
    age INT,
    num_of_operations INT,
    num_of_fatal_operations INT,
    salary_coeff_for_harmful_conditions FLOAT,
    extra_vacation_days INT
);

-- Персонал
DROP TABLE IF EXISTS staff;
CREATE TABLE IF NOT EXISTS staff (
    staff_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    age INT,
    role VARCHAR(255),
    work_experience INT
);

-- Занятість
DROP TABLE IF EXISTS employment;
CREATE TABLE IF NOT EXISTS employment (
    employment_id INT AUTO_INCREMENT PRIMARY KEY,
    staff_id INT,
    doctor_id INT,
    name VARCHAR(255),
    hospital_id INT,
    clinic_id INT,
    is_active BOOLEAN,
    FOREIGN KEY (staff_id) REFERENCES staff(staff_id),
    FOREIGN KEY (doctor_id) REFERENCES doctor(doctor_id),
    FOREIGN KEY (hospital_id) REFERENCES hospital(hospital_id),
    FOREIGN KEY (clinic_id) REFERENCES clinic(clinic_id)
);

-- Пациєнти
DROP TABLE IF EXISTS patient;
CREATE TABLE IF NOT EXISTS patient (
    patient_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    date_of_birth DATE,
    address VARCHAR(255)
);

-- Лабораторії
DROP TABLE IF EXISTS laboratory;
CREATE TABLE IF NOT EXISTS laboratory (
    lab_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    specialization VARCHAR(255),
    address VARCHAR(255)
);

-- Лабораторні тести
DROP TABLE IF EXISTS laboratory_test;
CREATE TABLE IF NOT EXISTS laboratory_test (
    test_id INT AUTO_INCREMENT PRIMARY KEY,

```

```

lab_id INT,
test_name VARCHAR(255),
test_date DATE,
patient_id INT,
hospital_id INT,
clinic_id INT,
FOREIGN KEY (lab_id) REFERENCES laboratory(lab_id),
FOREIGN KEY (patient_id) REFERENCES patient(patient_id),
FOREIGN KEY (hospital_id) REFERENCES hospital(hospital_id),
FOREIGN KEY (clinic_id) REFERENCES clinic(clinic_id)
);

-- Контракти лабораторії
DROP TABLE IF EXISTS laboratory_contracts;
CREATE TABLE IF NOT EXISTS laboratory_contracts (
    contract_id INT AUTO_INCREMENT PRIMARY KEY,
    lab_id INT,
    hospital_id INT NULL,
    clinic_id INT NULL,
    is_active BOOLEAN,
    FOREIGN KEY (lab_id) REFERENCES laboratory(lab_id),
    FOREIGN KEY (hospital_id) REFERENCES hospital(hospital_id),
    FOREIGN KEY (clinic_id) REFERENCES clinic(clinic_id)
);

-- Направлення
DROP TABLE IF NOT EXISTS referral;
CREATE TABLE IF NOT EXISTS referral (
    id INT AUTO_INCREMENT PRIMARY KEY,
    from_polyclinic_id INT,
    to_hospital_id INT,
    patient_id INT,
    specialization VARCHAR(255),
    referral_date DATE,
    FOREIGN KEY (from_polyclinic_id) REFERENCES clinic(clinic_id),
    FOREIGN KEY (to_hospital_id) REFERENCES hospital(hospital_id),
    FOREIGN KEY (patient_id) REFERENCES patient(patient_id)
);

-- Записи пацієнтів
DROP TABLE IF EXISTS patient_records;
CREATE TABLE IF NOT EXISTS patient_records (
    record_id INT AUTO_INCREMENT PRIMARY KEY,
    patient_id INT NOT NULL,
    hospital_id INT,
    clinic_id INT,
    date_of_entry DATE NOT NULL,
    disease VARCHAR(255) NOT NULL,
    doctor_id INT NOT NULL,
    treatment_type VARCHAR(255) NOT NULL,
    cabinet_id INT,
    referral_to_hospital_id INT,
    treatment_start_at DATE,
    treatment_end_at DATE,
    building_id INT,
    department_id INT,
    ward_id INT,
    bed_id INT,
    temperature_when_entered FLOAT,
    FOREIGN KEY (patient_id) REFERENCES patient(patient_id),
    FOREIGN KEY (hospital_id) REFERENCES hospital(hospital_id),
    FOREIGN KEY (clinic_id) REFERENCES clinic(clinic_id),
);

```

```

FOREIGN KEY (doctor_id) REFERENCES doctor(doctor_id),
FOREIGN KEY (referral_to_hospital_id) REFERENCES referral(id),
FOREIGN KEY (building_id) REFERENCES building(building_id),
FOREIGN KEY (department_id) REFERENCES department(department_id),
FOREIGN KEY (ward_id) REFERENCES ward(ward_id),
FOREIGN KEY (bed_id) REFERENCES bed(bed_id)
);

-- Кабінети клініки
DROP TABLE IF EXISTS cabinets;
CREATE TABLE IF NOT EXISTS cabinets (
    cabinet_id INT AUTO_INCREMENT PRIMARY KEY,
    clinic_id INT,
    FOREIGN KEY (clinic_id) REFERENCES clinic(clinic_id)
);
—INSERTING DATA INTO TABLES
-- Заповнення таблиці hospital
INSERT INTO hospital (name, address, specialization) VALUES
('Central City Hospital', '123 Main St', 'General'),
('Regional Hospital', '456 Regional Rd', 'Trauma and Emergency');

-- Заповнення таблиці building
INSERT INTO building (hospital_id, name, address) VALUES
(1, 'Main Building', '123 Main St'),
(1, 'Surgery Building', '125 Main St'),
(2, 'Emergency Building', '456 Regional Rd'),
(2, 'Trauma Building', '458 Regional Rd');

-- Заповнення таблиці department
INSERT INTO department (building_id, name, specialization) VALUES
(1, 'General Medicine', 'General Treatment'),
(1, 'Cardiology', 'Heart Diseases'),
(2, 'Surgery', 'Surgeries'),
(3, 'Emergency', 'Emergency Treatment'),
(4, 'Trauma', 'Trauma Treatment');

-- Заповнення таблиці ward
INSERT INTO ward (department_id, capacity, filled_status) VALUES
(1, 30, 20), (1, 30, 15), -- Збільшено кількість палат
(2, 20, 15), (2, 20, 10), -- Збільшено кількість палат
(3, 10, 8), (3, 10, 6), -- Збільшено кількість палат
(4, 25, 18), (4, 25, 12), -- Збільшено кількість палат
(5, 15, 12), (5, 15, 10); -- Збільшено кількість палат

-- Заповнення таблиці bed
INSERT INTO bed (ward_id, is_occupied) VALUES
(1, TRUE), (1, FALSE), (1, TRUE), (1, FALSE), (1, TRUE), (1, FALSE),
(2, TRUE), (2, FALSE), (2, TRUE), (2, FALSE), (2, TRUE), (2, FALSE),
(3, TRUE), (3, FALSE), (3, TRUE), (3, FALSE), (3, TRUE), (3, FALSE),
(4, TRUE), (4, FALSE), (4, TRUE), (4, FALSE), (4, TRUE), (4, FALSE),
(5, TRUE), (5, FALSE), (5, TRUE), (5, FALSE), (5, TRUE), (5, FALSE),
(6, TRUE), (6, FALSE), (6, TRUE), (6, FALSE), (6, TRUE), (6, FALSE),
(7, TRUE), (7, FALSE), (7, TRUE), (7, FALSE), (7, TRUE), (7, FALSE),
(8, TRUE), (8, FALSE), (8, TRUE), (8, FALSE), (8, TRUE), (8, FALSE),
(9, TRUE), (9, FALSE), (9, TRUE), (9, FALSE), (9, TRUE), (9, FALSE),
(10, TRUE), (10, FALSE), (10, TRUE), (10, FALSE), (10, TRUE), (10, FALSE);

-- Заповнення таблиці clinic
INSERT INTO clinic (hospital_id, name) VALUES
(1, 'Downtown Polyclinic'),
(2, 'Eastside Polyclinic'),
(NULL, 'Independent Polyclinic');

-- Заповнення таблиці cabinets
INSERT INTO cabinets (clinic_id) VALUES
(1), (1), (1), (1), -- Збільшено кількість кабінетів
(2), (2), (2), (2), -- Збільшено кількість кабінетів

```

(3), (3), (3); -- Збільшено кількість кабінетів

-- Fill the staff table

```
INSERT INTO staff (name, age, role, work_experience) VALUES
('John Doe', 45, 'Nurse', 20),
('Mary Smith', 38, 'Orderly', 15),
('Helen Brown', 30, 'Cleaner', 8),
('Alexander White', 50, 'Nurse', 25),
('Natalie Green', 28, 'Orderly', 6),
('Victoria Black', 42, 'Cleaner', 12),
('Andrew Adams', 35, 'Nurse', 10),
('Svetlana Clark', 47, 'Orderly', 22),
('Dmitry Hall', 33, 'Cleaner', 7),
('Ludmila Allen', 41, 'Nurse', 18),
('Vladimir Young', 29, 'Orderly', 5),
('Anna King', 36, 'Cleaner', 9),
('Oleg Scott', 39, 'Nurse', 14),
('Tatiana Turner', 48, 'Orderly', 23),
('Michael Carter', 34, 'Cleaner', 11);
```

-- Fill the doctor table

```
INSERT INTO doctor (name, specialization, work_experience_in_years, degree, can_work_in_many_institutions, age,
num_of_operations, num_of_fatal_operations, salary_coef_for_harmful_conditions, extra_vacation_days) VALUES
('Dr. John Smith', 'Surgeon', 20, 'PhD', TRUE, 50, 300, 5, NULL, NULL),
('Dr. Emily Johnson', 'Surgeon', 15, 'MD', FALSE, 40, 250, 2, NULL, NULL),
('Dr. Michael Brown', 'Gynecologist', 25, 'PhD', TRUE, 55, 400, 1, NULL, NULL),
('Dr. Sarah Davis', 'Dentist', 18, 'PhD', TRUE, 45, 200, 0, 1.2, NULL),
('Dr. William Wilson', 'Radiologist', 22, 'MD', FALSE, 48, NULL, NULL, 1.3, 30),
('Dr. Linda Martinez', 'Neurologist', 17, 'MD', FALSE, 43, NULL, NULL, NULL, 30),
('Dr. James Anderson', 'Surgeon', 30, 'PhD', TRUE, 60, 500, 10, NULL, NULL),
('Dr. Barbara Taylor', 'Gynecologist', 12, 'MD', FALSE, 38, 150, 0, NULL, NULL),
('Dr. Robert Thomas', 'Dentist', 14, 'MD', FALSE, 41, 180, 0, 1.2, NULL),
('Dr. Patricia Jackson', 'Radiologist', 19, 'MD', FALSE, 44, NULL, NULL, 1.3, 30),
('Dr. Charles Harris', 'Neurologist', 16, 'MD', FALSE, 42, NULL, NULL, NULL, 30),
('Dr. Christopher Walker', 'Surgeon', 10, 'MD', FALSE, 35, 100, 0, NULL, NULL),
('Dr. Karen Lee', 'Gynecologist', 20, 'PhD', TRUE, 46, 300, 2, NULL, NULL),
('Dr. Daniel Hall', 'Dentist', 11, 'MD', FALSE, 36, 120, 0, 1.2, NULL),
('Dr. Jennifer Young', 'Radiologist', 23, 'MD', FALSE, 50, NULL, NULL, 1.3, 30);
```

-- Fill the employment table

```
INSERT INTO employment (staff_id, doctor_id, name, hospital_id, clinic_id, is_active) VALUES
```

-- Staff employment

```
(1, NULL, 'John Doe', 1, NULL, TRUE),
(2, NULL, 'Mary Smith', 1, NULL, TRUE),
(3, NULL, 'Helen Brown', 1, NULL, TRUE),
(4, NULL, 'Alexander White', 1, NULL, TRUE),
(5, NULL, 'Natalie Green', 1, NULL, TRUE),
(6, NULL, 'Victoria Black', 2, NULL, TRUE),
(7, NULL, 'Andrew Adams', 2, NULL, TRUE),
(8, NULL, 'Svetlana Clark', 2, NULL, TRUE),
(9, NULL, 'Dmitry Hall', 2, NULL, TRUE),
(10, NULL, 'Ludmila Allen', 2, NULL, TRUE),
(11, NULL, 'Vladimir Young', NULL, 1, TRUE),
(12, NULL, 'Anna King', NULL, 1, TRUE),
(13, NULL, 'Oleg Scott', NULL, 2, TRUE),
(14, NULL, 'Tatiana Turner', NULL, 2, TRUE),
(15, NULL, 'Michael Carter', NULL, 3, TRUE),
```

-- Doctor employment

```
(NULL, 1, 'Dr. John Smith', 1, NULL, TRUE),
(NULL, 1, 'Dr. John Smith', 2, NULL, TRUE),
(NULL, 1, 'Dr. John Smith', 1, 1, TRUE),
(NULL, 2, 'Dr. Emily Johnson', 1, NULL, TRUE),
(NULL, 3, 'Dr. Michael Brown', 1, NULL, TRUE),
```

```

(NULL, 3, 'Dr. Michael Brown', 2, NULL, TRUE),
(NULL, 3, 'Dr. Michael Brown', 2, 2, TRUE),
(NULL, 4, 'Dr. Sarah Davis', 1, NULL, TRUE),
(NULL, 5, 'Dr. William Wilson', NULL, 1, TRUE),
(NULL, 5, 'Dr. William Wilson', NULL, 2, TRUE),
(NULL, 5, 'Dr. William Wilson', 1, 2, TRUE),
(NULL, 6, 'Dr. Linda Martinez', NULL, 2, TRUE),
(NULL, 7, 'Dr. James Anderson', 1, NULL, TRUE),
(NULL, 7, 'Dr. James Anderson', 2, NULL, TRUE),
(NULL, 7, 'Dr. James Anderson', 1, 3, TRUE),
(NULL, 8, 'Dr. Barbara Taylor', 2, NULL, TRUE),
(NULL, 9, 'Dr. Robert Thomas', NULL, 1, TRUE),
(NULL, 10, 'Dr. Patricia Jackson', NULL, 2, TRUE),
(NULL, 11, 'Dr. Charles Harris', NULL, 2, TRUE),
(NULL, 12, 'Dr. Christopher Walker', 1, NULL, TRUE),
(NULL, 13, 'Dr. Karen Lee', 1, NULL, TRUE),
(NULL, 13, 'Dr. Karen Lee', 2, NULL, TRUE),
(NULL, 13, 'Dr. Karen Lee', 1, 3, TRUE),
(NULL, 14, 'Dr. Daniel Hall', NULL, 1, TRUE),
(NULL, 15, 'Dr. Jennifer Young', NULL, 3, TRUE);

```

-- Fill the patients table

```

INSERT INTO patient (name, date_of_birth, address) VALUES
('John Doe', '1985-01-15', '123 Main St'),
('Jane Smith', '1990-02-20', '456 Elm St'),
('Robert Johnson', '1982-03-25', '789 Maple St'),
('Linda Brown', '1975-04-30', '101 Pine St'),
('Michael Davis', '1988-05-05', '202 Oak St'),
('Emily Garcia', '1992-06-10', '303 Cedar St'),
('James Martinez', '1979-07-15', '404 Birch St'),
('Sarah Rodriguez', '1983-08-20', '505 Spruce St'),
('David Wilson', '1991-09-25', '606 Walnut St'),
('Laura Moore', '1987-10-30', '707 Cherry St'),
('Richard Taylor', '1984-11-05', '808 Chestnut St'),
('Susan Anderson', '1978-12-10', '909 Hickory St'),
('Charles Thomas', '1981-01-15', '111 Ash St'),
('Jessica Jackson', '1993-02-20', '222 Willow St'),
('Christopher White', '1989-03-25', '333 Poplar St'),
('Karen Harris', '1976-04-30', '444 Fir St'),
('Matthew Martin', '1990-05-05', '555 Redwood St'),
('Patricia Thompson', '1985-06-10', '666 Cypress St'),
('Joshua Lee', '1980-07-15', '777 Alder St'),
('Elizabeth King', '1994-08-20', '888 Hemlock St'),
('Andrew Scott', '1986-09-25', '999 Magnolia St'),
('Barbara Green', '1983-10-30', '121 Maple St'),
('Daniel Adams', '1991-11-05', '234 Oak St'),
('Nancy Baker', '1979-12-10', '345 Cedar St'),
('Paul Gonzalez', '1987-01-15', '456 Birch St'),
('Sandra Nelson', '1992-02-20', '567 Spruce St'),
('Steven Carter', '1984-03-25', '678 Walnut St'),
('Betty Mitchell', '1988-04-30', '789 Cherry St'),
('George Perez', '1990-05-05', '890 Chestnut St'),
('Donna Roberts', '1985-06-10', '901 Hickory St');

```

-- Fill the referral table

```

INSERT INTO referral (from_polyclinic_id, to_hospital_id, patient_id, specialization, referral_date) VALUES
(1, 1, 1, 'Cardiology', '2024-01-10'),
(2, 2, 2, 'Neurology', '2024-01-15'),
(3, 1, 3, 'Surgery', '2024-01-20'),
(1, 2, 4, 'Orthopedics', '2024-01-25'),
(2, 1, 5, 'Pediatrics', '2024-02-01'),
(3, 2, 6, 'Radiology', '2024-02-05'),

```

```

(1, 1, 7, 'Gynecology', '2024-02-10'),
(2, 2, 8, 'Dermatology', '2024-02-15'),
(3, 1, 9, 'Gastroenterology', '2024-02-20'),
(1, 2, 10, 'Urology', '2024-02-25');

-- Records for patients treated in hospitals
INSERT INTO patient_records (patient_id, hospital_id, clinic_id, date_of_entry, disease, doctor_id, treatment_type, cabinet_id, referral_to_hospital_id, treatment_start_at, treatment_end_at, building_id, department_id, ward_id, bed_id, temperature_when_entered)
VALUES
(1, 1, NULL, '2024-03-01', 'Heart Disease', 1, 'Surgery', NULL, 1, '2024-03-01', '2024-03-10', 1, 1, 1, 1, 37.5),
(2, 2, NULL, '2024-03-05', 'Neurological Disorder', 2, 'Therapy', NULL, 2, '2024-03-05', '2024-03-12', 3, 2, 3, 5, 38.0),
(3, 1, NULL, '2024-03-10', 'Surgical Procedure', 3, 'Surgery', NULL, 3, '2024-03-10', '2024-03-20', 2, 3, 5, 9, 37.8),
(4, 2, NULL, '2024-03-15', 'Orthopedic Surgery', 4, 'Surgery', NULL, 4, '2024-03-15', '2024-03-25', 4, 4, 7, 13, 36.9),
(5, 1, NULL, '2024-03-20', 'Pediatric Treatment', 5, 'Medication', NULL, 5, '2024-03-20', '2024-03-30', 1, 1, 1, 1, 37.2),
(6, 2, NULL, '2024-03-25', 'Radiological Exam', 6, 'Therapy', NULL, 6, '2024-03-25', '2024-04-05', 3, 2, 3, 5, 38.2),
(7, 1, NULL, '2024-03-30', 'Gynecological Treatment', 7, 'Medication', NULL, 7, '2024-03-30', '2024-04-10', 2, 3, 5, 9, 37.7),
(8, 2, NULL, '2024-04-01', 'Dermatological Treatment', 8, 'Therapy', NULL, 8, '2024-04-01', '2024-04-15', 4, 4, 7, 13, 36.8),
(9, 1, NULL, '2024-04-05', 'Gastroenterological Treatment', 9, 'Medication', NULL, 9, '2024-04-05', '2024-04-20', 1, 1, 1, 1, 37.4),
(10, 2, NULL, '2024-04-10', 'Urological Treatment', 10, 'Surgery', NULL, 10, '2024-04-10', '2024-04-25', 3, 2, 3, 5, 37.9);

-- Records for patients treated in clinics
INSERT INTO patient_records (patient_id, hospital_id, clinic_id, date_of_entry, disease, doctor_id, treatment_type, cabinet_id, referral_to_hospital_id, treatment_start_at, treatment_end_at, building_id, department_id, ward_id, bed_id, temperature_when_entered)
VALUES
(1, NULL, 1, '2024-03-01', 'Flu', 11, 'Consultation', 1, NULL, '2024-03-01', '2024-03-03', NULL, NULL, NULL, NULL, NULL, NULL),
(2, NULL, 2, '2024-03-05', 'Cold', 12, 'Medication', 2, NULL, '2024-03-05', '2024-03-06', NULL, NULL, NULL, NULL, NULL, NULL),
(3, NULL, 1, '2024-03-10', 'Toothache', 13, 'Therapy', 3, NULL, '2024-03-10', '2024-03-11', NULL, NULL, NULL, NULL, NULL, NULL),
(4, NULL, 2, '2024-03-15', 'Skin Rash', 14, 'Consultation', 4, NULL, '2024-03-15', '2024-03-16', NULL, NULL, NULL, NULL, NULL, NULL),
(5, NULL, 3, '2024-03-20', 'Migraine', 15, 'Therapy', 5, NULL, '2024-03-20', '2024-03-21', NULL, NULL, NULL, NULL, NULL, NULL),
(6, NULL, 1, '2024-03-25', 'Back Pain', 11, 'Consultation', 6, NULL, '2024-03-25', '2024-03-26', NULL, NULL, NULL, NULL, NULL, NULL),
(7, NULL, 2, '2024-03-30', 'Allergy', 12, 'Medication', 7, NULL, '2024-03-30', '2024-03-31', NULL, NULL, NULL, NULL, NULL, NULL),
(8, NULL, 3, '2024-04-01', 'Stomachache', 13, 'Therapy', 8, NULL, '2024-04-01', '2024-04-02', NULL, NULL, NULL, NULL, NULL, NULL),
(9, NULL, 1, '2024-04-05', 'Ear Infection', 14, 'Medication', 9, NULL, '2024-04-05', '2024-04-06', NULL, NULL, NULL, NULL, NULL, NULL),
(10, NULL, 2, '2024-04-10', 'Sore Throat', 15, 'Consultation', 10, NULL, '2024-04-10', '2024-04-11', NULL, NULL, NULL, NULL, NULL, NULL);

-- Fill the laboratory table
INSERT INTO laboratory (name, specialization, address) VALUES
('BioLab', 'Biochemical', '123 Bio St'),
('PhysioLab', 'Physiological', '456 Physio Rd'),
('ChemLab', 'Chemical', '789 Chem Ave'),
('AdvancedLab', 'Biochemical, Physiological', '101 Advanced Blvd'),
('GeneralLab', 'Biochemical, Chemical', '202 General Ln');

-- Fill the laboratory_contracts table
INSERT INTO laboratory_contracts (lab_id, hospital_id, clinic_id, is_active) VALUES
(1, 1, NULL), -- BioLab has a contract with Central City Hospital
(1, 2, NULL), -- BioLab has a contract with Regional Hospital
(2, 1, NULL), -- PhysioLab has a contract with Central City Hospital
(3, NULL, 1, NULL), -- ChemLab has a contract with Downtown Polyclinic
(3, NULL, 2, NULL), -- ChemLab has a contract with Eastside Polyclinic
(4, 2, NULL, NULL), -- AdvancedLab has a contract with Regional Hospital
(4, NULL, 3, NULL), -- AdvancedLab has a contract with Independent Polyclinic
(5, 1, NULL, NULL), -- GeneralLab has a contract with Central City Hospital
(5, NULL, 3, NULL); -- GeneralLab has a contract with Independent Polyclinic

-- Fill the laboratory_test table
INSERT INTO laboratory_test (lab_id, test_name, test_date, patient_id, hospital_id, clinic_id) VALUES
(1, 'Blood Test', '2024-01-10', 1, 1, NULL),
(1, 'Urine Test', '2024-01-15', 2, 2, NULL),
(2, 'Stress Test', '2024-01-20', 3, 1, NULL),
(3, 'Chemical Analysis', '2024-01-25', 4, NULL, 1),
(3, 'Toxicology', '2024-02-01', 5, NULL, 2),
(4, 'Comprehensive Test', '2024-02-05', 6, 2, NULL),

```

```

(4, 'Hormone Test', '2024-02-10', 7, NULL, 3),
(5, 'Basic Metabolic Panel', '2024-02-15', 8, 1, NULL),
(5, 'Complete Blood Count', '2024-02-20', 9, NULL, 3),
(1, 'Blood Test', '2024-02-25', 10, 1, NULL),
(2, 'Cardiac Stress Test', '2024-03-01', 11, 1, NULL),
(3, 'Chemical Analysis', '2024-03-05', 12, NULL, 1),
(4, 'Hormone Test', '2024-03-10', 13, NULL, 3),
(5, 'Basic Metabolic Panel', '2024-03-15', 14, 1, NULL),
(1, 'Urine Test', '2024-03-20', 15, 2, NULL);

PROCEDURES
DROP PROCEDURE IF EXISTS get_doctors_by_specialization;
DELIMITER $$

CREATE PROCEDURE IF NOT EXISTS get_doctors_by_specialization(IN prof VARCHAR(255), IN institution_id INT, IN institution_type VARCHAR(255))
BEGIN
    IF institution_id IS NULL THEN
        SELECT DISTINCT d.name, d.specialization
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof
    UNION
    SELECT 'TOTAL DOCTORS' AS name, COUNT(*) AS specialization
    FROM (
        SELECT DISTINCT d.name, d.specialization
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof
    ) AS subquery;
    ELSEIF institution_type = 'hospital' THEN
        SELECT DISTINCT d.name, d.specialization
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND e.hospital_id = institution_id
    UNION
    SELECT 'TOTAL DOCTORS' AS name, COUNT(*) AS specialization
    FROM (
        SELECT DISTINCT d.name, d.specialization
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND e.hospital_id = institution_id
    ) AS subquery;
    ELSEIF institution_type = 'clinic' THEN
        SELECT DISTINCT d.name, d.specialization
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND e.clinic_id = institution_id
    UNION
    SELECT 'TOTAL DOCTORS' AS name, COUNT(*) AS specialization
    FROM (
        SELECT DISTINCT d.name, d.specialization
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND e.clinic_id = institution_id
    ) AS subquery;
END

```

```

END IF;
END$$
DELIMITER ;
-- Тестування процедури
CALL get_doctors_by_specialization('Surgeon', 1, 'hospital');
CALL get_doctors_by_specialization('Surgeon', 1, 'clinic');
CALL get_doctors_by_specialization('Surgeon', NULL, NULL);
DROP PROCEDURE IF EXISTS get_staff_by_role;
DELIMITER $$

CREATE PROCEDURE IF NOT EXISTS get_staff_by_role(IN role VARCHAR(255), IN institution_id INT, IN institution_type
VARCHAR(255))
BEGIN
IF institution_id IS NULL THEN
    SELECT DISTINCT s.name, s.role
    FROM staff s
    JOIN employment e ON s.staff_id = e.staff_id
    WHERE s.role = role
UNION

SELECT 'TOTAL STAFF' AS name, COUNT(*) AS role
FROM (
    SELECT DISTINCT s.name, s.role
    FROM staff s
    JOIN employment e ON s.staff_id = e.staff_id
    WHERE s.role = role
) AS subquery;
ELSEIF institution_type = 'hospital' THEN
    SELECT DISTINCT s.name, s.role
    FROM staff s
    JOIN employment e ON s.staff_id = e.staff_id
    WHERE s.role = role AND e.hospital_id = institution_id
UNION

SELECT 'TOTAL STAFF' AS name, COUNT(*) AS role
FROM (
    SELECT DISTINCT s.name, s.role
    FROM staff s
    JOIN employment e ON s.staff_id = e.staff_id
    WHERE s.role = role AND e.hospital_id = institution_id
) AS subquery;
ELSEIF institution_type = 'clinic' THEN
    SELECT DISTINCT s.name, s.role
    FROM staff s
    JOIN employment e ON s.staff_id = e.staff_id
    WHERE s.role = role AND e.clinic_id = institution_id
UNION

SELECT 'TOTAL STAFF' AS name, COUNT(*) AS role
FROM (
    SELECT DISTINCT s.name, s.role
    FROM staff s
    JOIN employment e ON s.staff_id = e.staff_id
    WHERE s.role = role AND e.clinic_id = institution_id
) AS subquery;
END IF;
END$$
DELIMITER ;

```

```

-- Тестування процедури
CALL get_staff_by_role('Nurse', 1, 'hospital');
CALL get_staff_by_role('Nurse', 1, 'clinic');
CALL get_staff_by_role('Nurse', NULL, NULL);
DROP PROCEDURE IF EXISTS get_doctors_by_operations;
DELIMITER $$

CREATE PROCEDURE IF NOT EXISTS get_doctors_by_operations(IN prof VARCHAR(255), IN min_operations INT, IN institution_id INT, IN institution_type VARCHAR(255))
BEGIN
    IF institution_id IS NULL THEN
        SELECT DISTINCT d.name, d.specialization, d.num_of_operations
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.num_of_operations >= min_operations
    UNION
        SELECT 'TOTAL DOCTORS' AS name, COUNT(*) AS num_of_operations, NULL AS specialization
        FROM (
            SELECT DISTINCT d.name, d.specialization, d.num_of_operations
            FROM doctor d
            JOIN employment e ON d.doctor_id = e.doctor_id
            WHERE d.specialization = prof AND d.num_of_operations >= min_operations
        ) AS subquery;
    ELSEIF institution_type = 'hospital' THEN
        SELECT DISTINCT d.name, d.specialization, d.num_of_operations
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.num_of_operations >= min_operations AND e.hospital_id = institution_id
    UNION
        SELECT 'TOTAL DOCTORS' AS name, COUNT(*) AS num_of_operations, NULL AS specialization
        FROM (
            SELECT DISTINCT d.name, d.specialization, d.num_of_operations
            FROM doctor d
            JOIN employment e ON d.doctor_id = e.doctor_id
            WHERE d.specialization = prof AND d.num_of_operations >= min_operations AND e.hospital_id = institution_id
        ) AS subquery;
    ELSEIF institution_type = 'clinic' THEN
        SELECT DISTINCT d.name, d.specialization, d.num_of_operations
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.num_of_operations >= min_operations AND e.clinic_id = institution_id
    UNION
        SELECT 'TOTAL DOCTORS' AS name, COUNT(*) AS num_of_operations, NULL AS specialization
        FROM (
            SELECT DISTINCT d.name, d.specialization, d.num_of_operations
            FROM doctor d
            JOIN employment e ON d.doctor_id = e.doctor_id
            WHERE d.specialization = prof AND d.num_of_operations >= min_operations AND e.clinic_id = institution_id
        ) AS subquery;
    END IF;
END$$
DELIMITER ;
-- Тестування процедури
CALL get_doctors_by_operations('Surgeon', 100, 1, 'hospital');
CALL get_doctors_by_operations('Surgeon', 100, 1, 'clinic');

```

```

CALL get_doctors_by_operations('Surgeon', 100, NULL, NULL);
DROP PROCEDURE IF EXISTS get_doctors_by_experience;
DELIMITER $$

CREATE PROCEDURE IF NOT EXISTS get_doctors_by_experience(IN prof VARCHAR(255), IN min_experience INT, IN institution_id INT, IN institution_type VARCHAR(255))
BEGIN
    IF institution_id IS NULL THEN
        SELECT DISTINCT d.name, d.specialization, d.work_experience_in_years
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.work_experience_in_years >= min_experience
    UNION
    SELECT 'TOTAL DOCTORS' AS name, COUNT(*) AS work_experience_in_years, NULL AS specialization
    FROM (
        SELECT DISTINCT d.name, d.specialization, d.work_experience_in_years
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.work_experience_in_years >= min_experience
    ) AS subquery;
    ELSEIF institution_type = 'hospital' THEN
        SELECT DISTINCT d.name, d.specialization, d.work_experience_in_years
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.work_experience_in_years >= min_experience AND e.hospital_id = institution_id
    UNION
    SELECT 'TOTAL DOCTORS' AS name, COUNT(*) AS work_experience_in_years, NULL AS specialization
    FROM (
        SELECT DISTINCT d.name, d.specialization, d.work_experience_in_years
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.work_experience_in_years >= min_experience AND e.hospital_id = institution_id
    ) AS subquery;
    ELSEIF institution_type = 'clinic' THEN
        SELECT DISTINCT d.name, d.specialization, d.work_experience_in_years
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.work_experience_in_years >= min_experience AND e.clinic_id = institution_id
    UNION
    SELECT 'TOTAL DOCTORS' AS name, COUNT(*) AS work_experience_in_years, NULL AS specialization
    FROM (
        SELECT DISTINCT d.name, d.specialization, d.work_experience_in_years
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.work_experience_in_years >= min_experience AND e.clinic_id = institution_id
    ) AS subquery;
    END IF;
END$$
DELIMITER ;
-- Тестування процедури
CALL get_doctors_by_experience('Surgeon', 10, 1, 'hospital');
CALL get_doctors_by_experience('Surgeon', 10, 1, 'clinic');
CALL get_doctors_by_experience('Surgeon', 10, NULL, NULL);
DROP PROCEDURE IF EXISTS get_doctors_by_degree;
DELIMITER $$
```

```

CREATE PROCEDURE IF NOT EXISTS get_doctors_by_degree(IN prof VARCHAR(255), IN degree VARCHAR(255), IN
institution_id INT, IN institution_type VARCHAR(255))
BEGIN
    IF institution_id IS NULL THEN
        SELECT DISTINCT d.name, d.specialization, d.degree
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.degree = degree
    UNION
    SELECT 'TOTAL DOCTORS' AS name, COUNT(*) AS degree, NULL AS specialization
    FROM (
        SELECT DISTINCT d.name, d.specialization, d.degree
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.degree = degree
    ) AS subquery;
    ELSEIF institution_type = 'hospital' THEN
        SELECT DISTINCT d.name, d.specialization, d.degree
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.degree = degree AND e.hospital_id = institution_id
    UNION
    SELECT 'TOTAL DOCTORS' AS name, COUNT(*) AS degree, NULL AS specialization
    FROM (
        SELECT DISTINCT d.name, d.specialization, d.degree
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.degree = degree AND e.hospital_id = institution_id
    ) AS subquery;
    ELSEIF institution_type = 'clinic' THEN
        SELECT DISTINCT d.name, d.specialization, d.degree
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.degree = degree AND e.clinic_id = institution_id
    UNION
    SELECT 'TOTAL DOCTORS' AS name, COUNT(*) AS degree, NULL AS specialization
    FROM (
        SELECT DISTINCT d.name, d.specialization, d.degree
        FROM doctor d
        JOIN employment e ON d.doctor_id = e.doctor_id
        WHERE d.specialization = prof AND d.degree = degree AND e.clinic_id = institution_id
    ) AS subquery;
    END IF;
END$$
DELIMITER ;
-- Тестування процедури
CALL get_doctors_by_degree('Surgeon', 'PhD', 1, 'hospital');
CALL get_doctors_by_degree('Surgeon', 'MD', 1, 'clinic');
CALL get_doctors_by_degree('Surgeon', 'PhD', NULL, NULL);
DROP PROCEDURE IF EXISTS get_patients_by_location;
DELIMITER $$
CREATE PROCEDURE IF NOT EXISTS get_patients_by_location(IN hospital_id INT, IN department_id INT, IN ward_id INT)
BEGIN

```

```

SELECT p.name AS patient_name, pr.date_of_entry, pr.disease AS patient_condition, pr.temperature_when_entered, d.name AS
doctor_name
FROM patient p
JOIN patient_records pr ON p.patient_id = pr.patient_id
JOIN doctor d ON pr.doctor_id = d.doctor_id
WHERE (pr.hospital_id = hospital_id OR hospital_id IS NULL)
AND (pr.department_id = department_id OR department_id IS NULL)
AND (pr.ward_id = ward_id OR ward_id IS NULL);
END$$
DELIMITER ;
-- Тестування процедури
CALL get_patients_by_location(1, NULL, NULL); -- Отримати всіх пацієнтів лікарні з ID 1
CALL get_patients_by_location(1, 3, NULL); -- Отримати всіх пацієнтів відділення з ID 2 у лікарні з ID 1
CALL get_patients_by_location(2, 4, 7); -- Отримати всіх пацієнтів палати з ID 3 у відділенні з ID 2 у лікарні з ID 1
DROP PROCEDURE IF EXISTS get_inpatients_by_hospital_or_doctor;
DELIMITER $$

CREATE PROCEDURE IF NOT EXISTS get_inpatients_by_hospital_or_doctor(IN hospital_id INT, IN doctor_id INT, IN start_date
DATE, IN end_date DATE)
BEGIN
    SELECT p.name AS patient_name, pr.date_of_entry, pr.treatment_start_at, pr.treatment_end_at, d.name AS doctor_name
    FROM patient p
    JOIN patient_records pr ON p.patient_id = pr.patient_id
    JOIN doctor d ON pr.doctor_id = d.doctor_id
    WHERE (pr.hospital_id = hospital_id OR hospital_id IS NULL)
    AND (pr.doctor_id = doctor_id OR doctor_id IS NULL)
    AND pr.treatment_start_at >= start_date
    AND pr.treatment_end_at <= end_date;
END$$
DELIMITER ;
-- Тестування процедури
CALL get_inpatients_by_hospital_or_doctor(1, NULL, '2024-01-01', '2024-12-31'); -- Отримати всіх пацієнтів, які пройшли
стационарне лікування у лікарні з ID 1 за 2024 рік
CALL get_inpatients_by_hospital_or_doctor(NULL, 1, '2024-01-01', '2024-12-31'); -- Отримати всіх пацієнтів, які пройшли
стационарне лікування у лікаря з ID 1 за 2024 рік
CALL get_inpatients_by_hospital_or_doctor(1, 1, '2024-01-01', '2024-12-31'); -- Отримати всіх пацієнтів, які пройшли стационарне
лікування у лікарні з ID 1 у лікаря з ID 1 за 2024 рік
DROP PROCEDURE IF EXISTS get_patients_by_doctor_specialization_and_clinic;
DELIMITER $$

CREATE PROCEDURE IF NOT EXISTS get_patients_by_doctor_specialization_and_clinic(IN specialization VARCHAR(255), IN
clinic_id INT)
BEGIN
    SELECT p.name AS patient_name, d.name AS doctor_name, d.specialization, pr.date_of_entry, pr.treatment_type
    FROM patient p
    JOIN patient_records pr ON p.patient_id = pr.patient_id
    JOIN doctor d ON pr.doctor_id = d.doctor_id
    WHERE d.specialization = specialization
    AND pr.clinic_id = clinic_id;
END$$
DELIMITER ;
-- Тестування процедури
CALL get_patients_by_doctor_specialization_and_clinic('Surgeon', 2); -- Отримати всіх пацієнтів, що спостерігаються у хірурга у
поліклініці з ID 2
CALL get_patients_by_doctor_specialization_and_clinic('Neurologist', 2); -- Отримати всіх пацієнтів, що спостерігаються у невролога
у поліклініці з ID 1
DROP PROCEDURE IF EXISTS getwards_and_beds_info;
DELIMITER $$

CREATE PROCEDURE IF NOT EXISTS getwards_and_beds_info(IN hospital_id INT)
BEGIN
    -- Загальна кількість палат і ліжок у лікарні
    SELECT COUNT(w.ward_id) AS total_wards, SUM(w.capacity) AS total_beds

```

```

FROM ward w
JOIN department d ON w.department_id = d.department_id
JOIN building b ON d.building_id = b.building_id
WHERE b.hospital_id = hospital_id;
-- Кількість палат і ліжок по кожному відділенню
SELECT d.name AS department_name, COUNT(w.ward_id) AS total_wards, SUM(w.capacity) AS total_beds
FROM ward w
JOIN department d ON w.department_id = d.department_id
JOIN building b ON d.building_id = b.building_id
WHERE b.hospital_id = hospital_id
GROUP BY d.name;
-- Кількість вільних ліжок у кожному відділенні
SELECT d.name AS department_name, SUM(w.capacity - w.filled_status) AS free_beds
FROM ward w
JOIN department d ON w.department_id = d.department_id
JOIN building b ON d.building_id = b.building_id
WHERE b.hospital_id = hospital_id
GROUP BY d.name;
-- Кількість повністю вільних палат у кожному відділенні
SELECT d.name AS department_name, COUNT(w.ward_id) AS free_wards
FROM ward w
JOIN department d ON w.department_id = d.department_id
JOIN building b ON d.building_id = b.building_id
WHERE b.hospital_id = hospital_id AND w.filled_status = 0
GROUP BY d.name;
END$$
DELIMITER ;
-- Тестування процедури
CALL get_wards_and_beds_info(1); -- Отримати інформацію про палати та ліжка у лікарні з ID 1
CALL get_wards_and_beds_info(2); -- Отримати інформацію про палати та ліжка у лікарні з ID 2
DROP PROCEDURE IF EXISTS get_clinic_cabinets_info;
DELIMITER $$

CREATE PROCEDURE IF NOT EXISTS get_clinic_cabinets_info(IN clinic_id INT, IN start_date DATE, IN end_date DATE)
BEGIN
    -- Загальна кількість кабінетів у поліклініці
    SELECT COUNT(c.cabinet_id) AS total_cabinets
    FROM cabinets c
    WHERE c.clinic_id = clinic_id;
    -- Число відвідувань кожного кабінету за певний період
    SELECT c.cabinet_id, COUNT(pr.record_id) AS visit_count
    FROM cabinets c
    LEFT JOIN patient_records pr ON c.cabinet_id = pr.cabinet_id
    WHERE c.clinic_id = clinic_id
        AND pr.date_of_entry BETWEEN start_date AND end_date
    GROUP BY c.cabinet_id;
END$$
DELIMITER ;
-- Тестування процедури
CALL get_clinic_cabinets_info(1, '2024-01-01', '2024-12-31'); -- Отримати інформацію про кабінети та відвідування у поліклініці з ID 1 за 2024 рік
CALL get_clinic_cabinets_info(2, '2024-01-01', '2024-12-31'); -- Отримати інформацію про кабінети та відвідування у поліклініці з ID 2 за 2024 рік
DROP PROCEDURE IF EXISTS get_doctor_performance;
DELIMITER $$

CREATE PROCEDURE IF NOT EXISTS get_doctor_performance(IN doctor_id INT, IN clinic_id INT, IN specialization VARCHAR(255), IN start_date DATE, IN end_date DATE)
BEGIN
    IF doctor_id IS NOT NULL THEN
        -- Продуктивність для конкретного лікаря
        SELECT d.name AS doctor_name, COUNT(pr.patient_id) / DATEDIFF(end_date, start_date) AS average_patients_per_day

```

```

FROM doctor d
JOIN patient_records pr ON d.doctor_id = pr.doctor_id
WHERE d.doctor_id = doctor_id
AND pr.date_of_entry BETWEEN start_date AND end_date
GROUP BY d.name;
ELSEIF clinic_id IS NOT NULL THEN
-- Продуктивність для всіх лікарів у поліклініці
SELECT d.name AS doctor_name, COUNT(pr.patient_id) / DATEDIFF(end_date, start_date) AS average_patients_per_day
FROM doctor d
JOIN patient_records pr ON d.doctor_id = pr.doctor_id
WHERE pr.clinic_id = clinic_id
AND pr.date_of_entry BETWEEN start_date AND end_date
GROUP BY d.name;
ELSEIF specialization IS NOT NULL THEN
-- Продуктивність для всіх лікарів заданого профілю
SELECT d.name AS doctor_name, COUNT(pr.patient_id) / DATEDIFF(end_date, start_date) AS average_patients_per_day
FROM doctor d
JOIN patient_records pr ON d.doctor_id = pr.doctor_id
WHERE d.specialization = specialization
AND pr.date_of_entry BETWEEN start_date AND end_date
GROUP BY d.name;
ELSE
-- Якщо жоден параметр не заданий, повертаємо повідомлення
SELECT 'Please provide either doctor_id, clinic_id, or specialization' AS message;
END IF;
END$$
DELIMITER ;
-- Тестування процедури з періодом 2024-03-01 до 2024-03-30
CALL get_doctor_performance(1, NULL, NULL, '2024-03-01', '2024-03-30'); -- Продуктивність для лікаря з ID 1 за березень 2024 року
CALL get_doctor_performance(NULL, 1, NULL, '2024-03-01', '2024-03-30'); -- Продуктивність для всіх лікарів у поліклініці з ID 1 за березень 2024 року
CALL get_doctor_performance(NULL, NULL, 'Surgeon', '2024-03-01', '2024-03-30'); -- Продуктивність для всіх хірургів за березень 2024 року
DROP PROCEDURE IF EXISTS get_doctor_workload_by_date;
DELIMITER $$

CREATE PROCEDURE IF NOT EXISTS get_doctor_workload_by_date(IN target_date DATE, IN doctor_id INT, IN hospital_id INT, IN specialization VARCHAR(255))
BEGIN
IF doctor_id IS NOT NULL THEN
-- Завантаження для конкретного лікаря на задану дату
SELECT d.name AS doctor_name, COUNT(pr.patient_id) AS patient_count
FROM doctor d
JOIN patient_records pr ON d.doctor_id = pr.doctor_id
WHERE d.doctor_id = doctor_id
AND pr.treatment_end_at > target_date
GROUP BY d.name;
ELSEIF hospital_id IS NOT NULL THEN
-- Завантаження для всіх лікарів у лікарні на задану дату
SELECT d.name AS doctor_name, COUNT(pr.patient_id) AS patient_count
FROM doctor d
JOIN patient_records pr ON d.doctor_id = pr.doctor_id
WHERE pr.hospital_id = hospital_id
AND pr.treatment_end_at > target_date
GROUP BY d.name;
ELSEIF specialization IS NOT NULL THEN
-- Завантаження для всіх лікарів заданого профілю на задану дату
SELECT d.name AS doctor_name, COUNT(pr.patient_id) AS patient_count
FROM doctor d
JOIN patient_records pr ON d.doctor_id = pr.doctor_id

```

```

WHERE d.specialization = specialization
    AND pr.treatment_end_at > target_date
    GROUP BY d.name;
ELSE
    -- Якщо жоден параметр не заданий, повертаємо повідомлення
    SELECT 'Please provide either doctor_id, hospital_id, or specialization' AS message;
END IF;
END$$
DELIMITER ;
-- Тестування процедури
CALL get_doctor_workload_by_date('2024-03-15', 1, NULL, NULL); -- Завантаження для лікаря з ID 1 на 15 березня 2024 року
CALL get_doctor_workload_by_date('2024-03-15', NULL, 1, NULL); -- Завантаження для всіх лікарів у лікарні з ID 1 на 15 березня 2024 року
CALL get_doctor_workload_by_date('2024-03-15', NULL, NULL, 'Surgeon'); -- Завантаження для всіх хірургів на 15 березня 2024 року
DROP PROCEDURE IF EXISTS get_patients_who_had_surgery;
DELIMITER $$
CREATE PROCEDURE IF NOT EXISTS get_patients_who_had_surgery(IN institution_id INT, IN institution_type VARCHAR(255),
IN doctor_id INT, IN start_date DATE, IN end_date DATE)
BEGIN
    IF institution_type = 'hospital' THEN
        -- Перелік пацієнтів, які перенесли операції у зазначеній лікарні
        SELECT p.name AS patient_name, pr.date_of_entry, pr.treatment_type, d.name AS doctor_name
        FROM patient p
        JOIN patient_records pr ON p.patient_id = pr.patient_id
        JOIN doctor d ON pr.doctor_id = d.doctor_id
        WHERE pr.hospital_id = institution_id
            AND pr.treatment_type = 'Surgery'
            AND pr.date_of_entry BETWEEN start_date AND end_date;
    ELSEIF institution_type = 'clinic' THEN
        -- Перелік пацієнтів, які перенесли операції у зазначеній поліклініці
        SELECT p.name AS patient_name, pr.date_of_entry, pr.treatment_type, d.name AS doctor_name
        FROM patient p
        JOIN patient_records pr ON p.patient_id = pr.patient_id
        JOIN doctor d ON pr.doctor_id = d.doctor_id
        WHERE pr.clinic_id = institution_id
            AND pr.treatment_type = 'Surgery'
            AND pr.date_of_entry BETWEEN start_date AND end_date;
    ELSEIF doctor_id IS NOT NULL THEN
        -- Перелік пацієнтів, які перенесли операції у конкретного лікаря
        SELECT p.name AS patient_name, pr.date_of_entry, pr.treatment_type, d.name AS doctor_name
        FROM patient p
        JOIN patient_records pr ON p.patient_id = pr.patient_id
        JOIN doctor d ON pr.doctor_id = d.doctor_id
        WHERE pr.doctor_id = doctor_id
            AND pr.treatment_type = 'Surgery'
            AND pr.date_of_entry BETWEEN start_date AND end_date;
    ELSE
        -- Якщо жоден параметр не заданий, повертаємо повідомлення
        SELECT 'Please provide either institution_id and institution_type, or doctor_id' AS message;
    END IF;
END$$
DELIMITER ;
-- Тестування процедури
CALL get_patients_who_had_surgery(1, 'hospital', NULL, '2024-01-01', '2024-03-30'); -- Перелік пацієнтів, які перенесли операції у лікарні з ID 1 за період 01.01.2024 - 30.03.2024
CALL get_patients_who_had_surgery(1, 'clinic', NULL, '2024-01-01', '2024-03-30'); -- Перелік пацієнтів, які перенесли операції у поліклініці з ID 1 за період 01.01.2024 - 30.03.2024
CALL get_patients_who_had_surgery(NULL, NULL, 1, '2024-01-01', '2024-03-30'); -- Перелік пацієнтів, які перенесли операції у лікаря з ID 1 за період 01.01.2024 - 30.03.2024

```

```

DROP PROCEDURE IF EXISTS get_laboratory_performance;
DELIMITER $$

CREATE PROCEDURE IF NOT EXISTS get_laboratory_performance(IN lab_id INT, IN hospital_id INT, IN clinic_id INT, IN start_date DATE, IN end_date DATE)
BEGIN
    IF lab_id IS NOT NULL THEN
        -- Продуктивність для конкретної лабораторії
        SELECT l.name AS laboratory_name, COUNT(lt.test_id) / DATEDIFF(end_date, start_date) AS average_tests_per_day
        FROM laboratory l
        JOIN laboratory_test lt ON l.lab_id = lt.lab_id
        WHERE l.lab_id = lab_id
        AND lt.test_date BETWEEN start_date AND end_date
        GROUP BY l.name;
    ELSEIF hospital_id IS NOT NULL THEN
        -- Продуктивність для всіх лабораторій у лікарні
        SELECT l.name AS laboratory_name, COUNT(lt.test_id) / DATEDIFF(end_date, start_date) AS average_tests_per_day
        FROM laboratory l
        JOIN laboratory_test lt ON l.lab_id = lt.lab_id
        WHERE lt.hospital_id = hospital_id
        AND lt.test_date BETWEEN start_date AND end_date
        GROUP BY l.name;
    ELSEIF clinic_id IS NOT NULL THEN
        -- Продуктивність для всіх лабораторій у поліклініці
        SELECT l.name AS laboratory_name, COUNT(lt.test_id) / DATEDIFF(end_date, start_date) AS average_tests_per_day
        FROM laboratory l
        JOIN laboratory_test lt ON l.lab_id = lt.lab_id
        WHERE lt.clinic_id = clinic_id
        AND lt.test_date BETWEEN start_date AND end_date
        GROUP BY l.name;
    ELSE
        -- Продуктивність для всіх лабораторій у місті
        SELECT l.name AS laboratory_name, COUNT(lt.test_id) / DATEDIFF(end_date, start_date) AS average_tests_per_day
        FROM laboratory l
        JOIN laboratory_test lt ON l.lab_id = lt.lab_id
        WHERE lt.test_date BETWEEN start_date AND end_date
        GROUP BY l.name;
    END IF;
END$$
DELIMITER ;

-- Тестування процедури
CALL get_laboratory_performance(1, NULL, NULL, '2024-03-01', '2024-03-30'); -- Продуктивність для лабораторії з ID 1 за період 01.03.2024 - 30.03.2024
CALL get_laboratory_performance(NULL, 1, NULL, '2024-03-01', '2024-03-30'); -- Продуктивність для всіх лабораторій у лікарні з ID 1 за період 01.03.2024 - 30.03.2024
CALL get_laboratory_performance(NULL, NULL, 1, '2024-03-01', '2024-03-30'); -- Продуктивність для всіх лабораторій у поліклініці з ID 1 за період 01.03.2024 - 30.03.2024
CALL get_laboratory_performance(NULL, NULL, NULL, '2024-03-01', '2024-03-30'); -- Продуктивність для всіх лабораторій у місті за період 01.03.2024 - 30.03.2024

```