ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Бази даних в медичних інформаційних системах»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 Комп'ютерної інженерії
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерна інженерія
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ *Іван ВАШКЕВИЧ*
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. КІД-41
Іван ВАШКЕВИЧ
Ім'я, ПРІЗВИЩЕ
Керівник: Ярослав ТОРОШАНКО
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ
Рецензент:
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Комп'ютерної інженерії

Ступінь вищої освіти бакалавр

Спеціальність 123 – Комп'ютерна інженерія

Освітньо-професійна програма Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії
_____ Наталія ЛАЩЕВСЬКА
_____
«_____» _____ 2025р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Вашкевич Іван Євгенович

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Бази даних в медичних інформаційних системах
керівник кваліфікаційної роботи Ярослав ТОРОШАНКО, к.т.н., с.н.с.,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025р. № 56
2. Строк подання кваліфікаційної роботи «5» червня 2025р.
3. Вихідні дані до кваліфікаційної роботи: науково-технічна література; медичні інформаційні системи; реляційні бази даних; нереляційні бази даних; моделі даних (ER-моделі, нормалізація); мови запитів SQL; система керування базами даних SQLite; захист медичних даних; хмарні технології для збереження та обробки інформації; законодавство з безпеки даних (GDPR, HIPAA); вимоги до конфіденційності в охороні здоров'я;
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Аналіз сучасних підходів до зберігання даних у медичних інформаційних системах
2. Нормативні вимоги до збереження конфіденційної медичної інформації
3. Вибір і обгрунтування типу бази даних для медичних систем
4. Реалізація бази даних, наповнення тестовими даними і приклади SQL-запитів
5. Перелік ілюстративного матеріалу: *презентація*
6. Дата видачі завдання «24» лютого 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	25.02-24.03.2025	Виконано
2	Обробка матеріалу	25.03-06.04.2025	Виконано
3	Розробка теоретичної частини	07.04-16.04.2025	Виконано
4	Розробка бази даних	17.04-22.04.2025	Виконано
5	Висновки за результатами дослідження	23.04-06.05.2025	Виконано
6	Оформлення пояснювальної записки	07.05-08.05.2025	Виконано
7	Розробка презентації	09.05-11.05.2025	Виконано
8	Передзахист	27.05-29.05.2025	Виконано
9	Здача в деканат	02.06-05.06.2025	Виконано

Здобувач вищої освіти _____ Іван ВАШКЕВИЧ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Ярослав ТОРОШАНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 43 стор., 27 рис., 30 джерел.

Мета роботи – дослідження теоретичних засад та практична реалізація медичної бази даних.

Об'єкт дослідження – дослідження інформаційних систем, призначених для обробки медичних даних.

Предмет дослідження – база даних як ключовий компонент медичної інформаційної системи.

Короткий зміст роботи: дослідження особливостей використання баз даних у медичних інформаційних системах. Проведення аналізу вимог до безпеки, продуктивності та масштабованості. Обґрунтування вибору реляційної моделі. Реалізація прикладу бази даних з основними сутностями, наповнення тестовими даними та виконання прикладів SQL-запитів у середовищі DB Browser for SQLite.

КЛЮЧОВІ СЛОВА: МЕДИЧНА ІНФОРМАЦІЙНА СИСТЕМА, МЕДИЧНІ ДАНІ, БАЗА ДАНИХ, ІНФОРМАЦІЙНА БЕЗПЕКА, SQL, NOSQL, ХМАРНА БАЗА ДАНИХ, РЕЛЯЦІЙНА МОДЕЛЬ, ПАЦІЄНТ, ЛІКАР, ПРИЙОМ, SQL-ЗАПИТ, ЕЛЕКТРОННА МЕДИЧНА КАРТКА.

ЗМІСТ

ВСТУП…… 8
1 ТЕОРЕТИЧНІ ОСНОВИ БАЗ ДАНИХ У МЕДИЦИНІ…….. 9
1.1 Поняття та класифікація медичних інформаційних систем…….. 9
1.2 Види баз даних, їх характеристика та застосування в МІС…….... 12
1.3 Вимоги до баз даних у медицині…….. 17
2 СПЕЦИФІКА ВИКОРИСТАННЯ БД У МЕДИЧНИХ СИСТЕМАХ…… 20
2.1 Приклади медичних інформаційних систем з реалізованими БД 20
2.2 Нормативно-правове регулювання захисту медичних даних…... 24
2.3 Проблеми впровадження: конфіденційність, інтеграція з обладнанням…… 28
2.4 Оцінка продуктивності, масштабованості та безпеки різних типів баз даних……...…… 30
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ БАЗИ ДАНИХ……... 38
3.1 Постановка завдання та вибір інструментів…… 38
3.2 Проєктування та створення бази даних……... 39
3.3 Наповнення бази даних тестовими даними та приклади запитів 43
3.4 Висновки до практичної частини…… 48
ВИСНОВКИ…… 50
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ…….. 51
Додаток А. Демонстраційні матеріали……..... 54

ВСТУП

У сучасних умовах цифровізації охорони здоров’я бази даних стали фундаментом для ефективного функціонування медичних інформаційних систем. Вони забезпечують структуроване зберігання, швидкий пошук і аналіз медичних даних, що є критично важливим для прийняття клінічних рішень, управління медичними ресурсами та проведення наукових досліджень. Особливу актуальність ця тема набуває в контексті розвитку електронних систем охорони здоров’я в Україні, зокрема державної системи eHealth, яка покликана реформувати галузь шляхом впровадження єдиного інформаційного простору.

Медичні інформаційні системи, що базуються на сучасних базах даних, дозволяють не лише автоматизувати рутинні процеси, але й забезпечують цілісність і конфіденційність даних пацієнтів. В Україні ці питання набувають особливої ваги через необхідність відповідності міжнародним стандартам, таким як GDPR, та національним вимогам щодо захисту персональних даних. Система eHealth, яка функціонує з 2018 року, стала першим кроком до створення єдиної цифрової інфраструктури охорони здоров’я, проте її розвиток потребує подальшого вдосконалення, зокрема в частині вибору оптимальних баз даних, які б забезпечували високу продуктивність, масштабованість і відмовостійкість.

Метою даної роботи є комплексний аналіз використання баз даних у медичних інформаційних системах із акцентом на особливості їхньої реалізації в українському контексті. У дослідженні розглядаються різні типи баз даних, їхні переваги та недоліки, а також нормативні вимоги, що регулюють їхнє використання в медицині. Практична частина роботи демонструє можливості застосування простої бази даних для організації медичної інформації, що може бути корисним для розуміння основних принципів роботи МІС.

Результати дослідження мають не лише теоретичне значення, але й практичну цінність, оскільки можуть бути використані для оптимізації вибору технологій баз даних при розробці або модернізації медичних інформаційних систем в Україні.

1 ТЕОРЕТИЧНІ ОСНОВИ БАЗ ДАНИХ У МЕДИЦИНІ

1.1 Поняття та класифікація медичних інформаційних систем

Медичні інформаційні системи (МІС) сьогодні являють собою складні організаційно-технологічні комплекси, які інтегрують методи обробки інформації, програмне забезпечення, апаратні рішення та нормативну базу для всебічного управління процесами в охороні здоров'я. Це не просто інструменти для зберігання даних, а цілісні екосистеми, що трансформують традиційні підходи до медичної практики.

Медичні інформаційні системи – це інтелектуальні платформи, що:

– агрегують структуровані та неструктуровані медичні дані (від текстових записів до зображень МРТ);

– забезпечують семантичну взаємодію між різнорідними системами

– реалізують механізми предиктивної аналітики;

– вбудовують знання (knowledge-based systems) у клінічні протоколи.

Еволюція МІС в Україні:

– *1990-2000*. Локальні автоматизовані системи обліку (наприклад, "Лікувальна справа");

– *2010-2017*. Децентралізовані електронні картки (розробки регіональних лікарень);

– *2018-нині*. Єдина державна система eHealth з API-доступом.

Ключові аспекти сучасних МІС:

– *клініко-технологічна інтеграція.* Системи починають включати IoT-рішення, наприклад інтеграцію з носимими пристроями (кардіомонторинг),

автоматичне оновлення даних з лабораторних аналізаторів, системи відеомоніторингу пацієнтів (розробка "SmartWard" в Охматдиті);

– *контекстно-залежні інтерфейси.* Вони передбачають адаптивні форми введення даних, голосові помічники для лікарів (український прототип "MedVoice"), AR-візуалізацію анатомічних структур;

– *комплаєнс-менеджмент.* Система заходів управління та контролю в компанії, спрямована на попередження та зменшення ризиків невідповідності до вимог законодавства, правил, стандартів і рекомендацій (зокрема, контролюючих органів). Він передбачає автоматичний аудит відповідності стандартам, системи попередження про лікарські взаємодії, модулі фармаконагляду (як в системі "Безпечна лікарня") тощо.

Класифікація медичних інформаційних систем:

– *клінічні МІС.* Найпоширеніший тип систем, безпосередньо пов'язаний з наданням медичної допомоги. До них належать електронні медичні картки (ЕМК) - наприклад, українська система eHealth, яка станом на 2024 рік охоплює понад 30 млн медичних записів. Система дозволяє лікарям усіх рівнів отримувати доступ до історії хвороби пацієнта, призначати лікування та виписувати електронні рецепти; системи підтримки лікарських рішень (CDSS) - як приклад можна навести розробку українських вчених "MedDecision", яка інтегрується з eHealth та надає лікарям рекомендації на основі аналізу схожих клінічних випадків; радіологічні інформаційні системи (RIS) - в Україні представлені такими розробками як "RADIO-ED" (використовується в обласних лікарнях Дніпра та Харкова), що дозволяє керувати потоком діагностичних зображень та їх інтерпретацією.

– *адміністративні МІС.* Ці системи забезпечують управління ресурсами медичних закладів, до них входять госпітальні інформаційні системи (HIS), як приклад в Україні набуває популярності система "MedManager", яка інтегрує функції управління палатами, розкладом лікарів, медичними запасами. Використовується в мережі клініник "Борис" та інших приватних медичних закладах, системи електронної черги - як приклад можна навести розробку "Doc.ua", яка поєднує функції запису до лікаря з системою рейтингів медичних закладів,

фінансово-аналітичні системи - наприклад, модуль "МедФін" в системі eHealth, що дозволяє автоматизувати процес розрахунків з НСЗУ.

– *дослідницькі МІС.* Ці системи спрямовані на аналіз великих масивів медичних даних, до них входять біоінформаційні платформи (українська розробка "BioUkraine", яка використовується в Інституті молекулярної біології НАНУ для аналізу геномних даних), системи клінічних досліджень - наприклад, "UCRegistry", створена спільно МОЗ України та асоціацією "Клінічні дослідження України" для моніторингу клінічних випробувань, а також епідеміологічні моніторингові системи - система "EpiMon", розроблена під час пандемії COVID-19 для відстеження поширення захворювання.

Ключова особливість МІС. Сучасні медичні інформаційні системи отримують свою ефективність завдяки глибокій інтеграції з спеціалізованими базами даних, що забезпечує цілий спектр критично важливих функцій. Ця інтеграція дозволяє створити єдиний інформаційний простір, де дані з різних джерел (лабораторій, діагностичних приладів, електронних медичних карток) автоматично агрегуються, стандартизуються і стають доступними для аналізу.

Наприклад, українська система eHealth демонструє, як належна організація бази даних дозволяє забезпечити безперервність медичного обслуговування. У випадку, коли пацієнт змінює місце проживання або лікаря, вся його медична історія залишається доступною через централізовану базу даних. Особливо важливо, що система підтримує стандарти HL7 та FHIR, що дозволяє інтегруватися з міжнародними медичними системами.

Інтеграція з БД також забезпечує автоматичне оновлення медичних протоколів, що особливо важливо в умовах швидкого розвитку медицини. Наприклад, коли МОЗ України оновлює рекомендації щодо лікування певного захворювання, ці зміни автоматично відображаються в системах підтримки прийняття рішень, що дозволяє лікарям по всій країні отримувати актуальну інформацію.

Крім того, така інтеграція створює основу для застосування штучного інтелекту в медицині. Український стартап "MedAI" використовує дані з eHealth

для тренування алгоритмів діагностики, що дозволяє виявляти певні захворювання на ранніх стадіях з точністю до 92%.

Таким чином, інтеграція МІС з базами даних не лише підвищує ефективність роботи медичних закладів, а й відкриває нові можливості для покращення якості медичного обслуговування, проведення масштабних досліджень та впровадження інноваційних підходів у охороні здоров'я.

1.2 Види баз даних, їх характеристика та застосування в МІС

Реляційні бази даних (SQL) представляють собою класичний підхід до організації структурованих даних, заснований на математичній теорії множин і реляційній алгебрі. Їх фундаментальною особливістю є організація інформації у вигляді таблиць (відносин), де кожен рядок є унікальним записом, а стовпці визначають атрибути цих записів (рис. 1.1). Між таблицями встановлюються логічні зв'язки через систему первинних і зовнішніх ключів, що дозволяє ефективно керувати складними взаємозв'язками в медичних даних. Наприклад, пацієнт може бути пов'язаний з кількома діагнозами, які в свою чергу пов'язані з різними методами лікування - і вся ця структура чітко відображається в реляційній схемі.

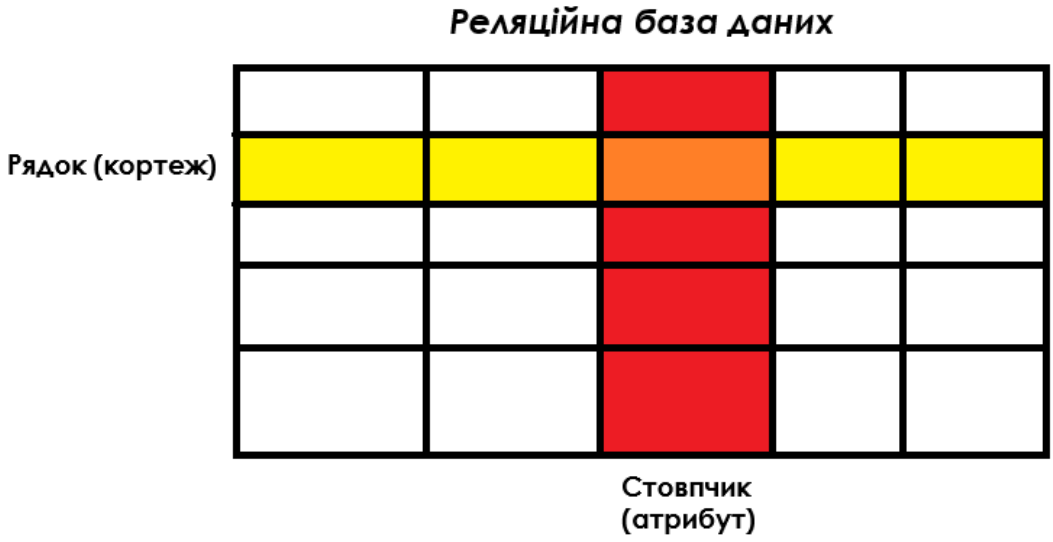


Рисунок 1.1 – Реляційна база даних

Основні характеристики реляційних баз даних:

– *ACID-транзакції (Atomicity, Consistency, Isolation, Durability)*. Фундаментальна концепція, що забезпечує надійність операцій з даними. Вона гарантує атомарність (повноту виконання операцій), узгодженість (дотримання бізнес-правил), ізольованість (паралельну роботу без конфліктів) та стійкість (збереження даних після збоїв). Наприклад, при оновленні історії хвороби система гарантує, що всі зміни будуть застосовані комплексно або скасовані повністю у разі помилки;

– *жорстка схема даних*. Вимагає чіткого визначення структури інформації до початку роботи, що сприяє стандартизації медичних записів. Кожен стовпець таблиці має конкретний тип даних (текст, число, дата), що запобігає помилкам введення та полегшує аналіз. Проте це ж обмежує гнучкість при роботі з нестандартними медичними випадками;

– *мова запитів SQL (Structured Query Language)*. Є потужним інструментом для роботи з медичними даними, дозволяючи виконувати складні запити для аналізу симптомів, діагнозів і методів лікування. Вона стандартизована і підтримується всіма сучасними СУБД, що спрощує розробку медичних додатків;

– *вертикальна масштабованість*. Є традиційним підходом для реляційних систем, що добре підходить для централізованих медичних архітектур. Однак воно має фізичні обмеження, що стимулює пошук альтернативних рішень для великих медичних установ.

Висновки для медичних застосувань реляційних БД:

– реляційні БД ідеальні для структурованих клінічних даних з чіткими зв'язками;

– ACID-гарантії критично важливі для відповідності медичним стандартам. Для українських лікарень середнього розміру оптимальним вибором є PostgreSQL, який поєднує ACID-гарантії з відносно невисокою вартістю володіння;

– SQL дозволяє реалізувати складні медичні аналітичні звіти;

– вертикальне масштабування підходить для централізованих систем, але має обмеження для великих лікарняних мереж.

NoSQL (нереляційні) бази даних, на відміну від реляційних, представляють собою сучасний підхід до організації неструктурованих та напівструктурованих даних, заснований на гнучких моделях сховища (рис. 1.2). Їх фундаментальною особливістю є відмова від жорсткої табличної структури на користь документної, ключ-значення, колоночної або графової моделей, що дозволяє ефективно керувати різнорідними медичними даними. Наприклад, результати комплексного обстеження пацієнта можуть містити структуровані лабораторні показники, неструктуровані нотатки лікаря та двійкові дані діагностичних зображень - і вся ця інформація може зберігатися в єдиному документі.

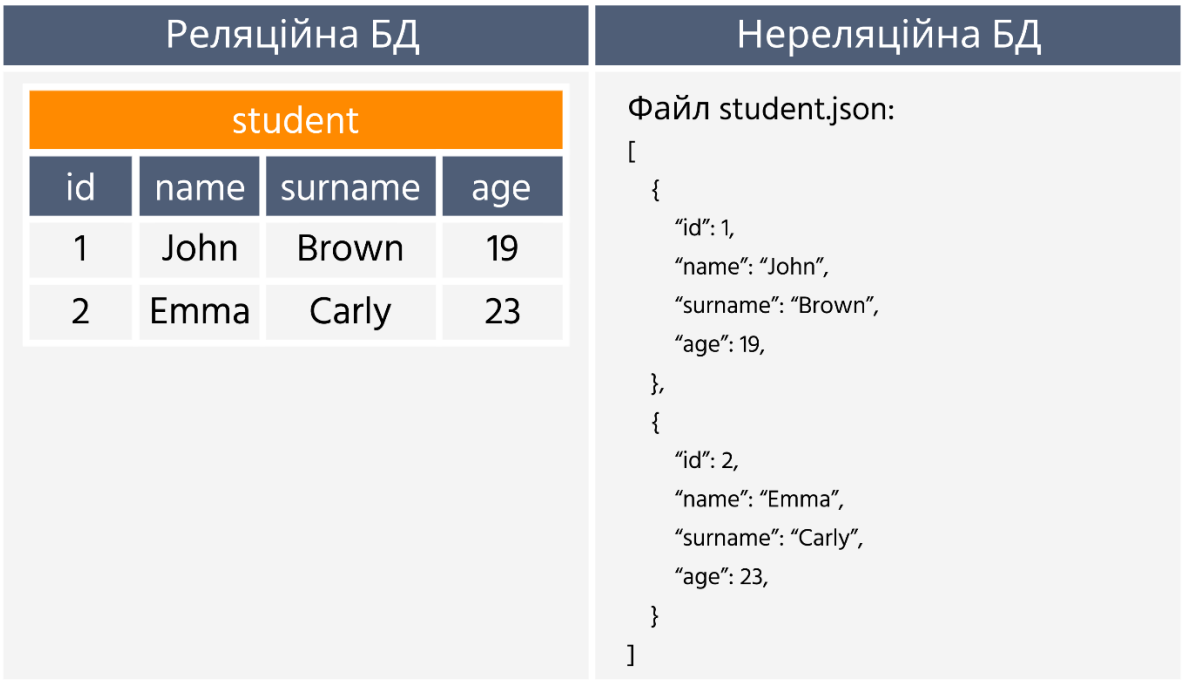


Рисунок 1.2 – Різниця між реляційними і нереляційними БД

Основні характеристики NoSQL баз даних:

– *гнучкість схеми даних (Schema-less)* дозволяє адаптувати структуру даних під конкретні медичні випадки без попереднього визначення всіх полів. Це особливо важливо для динамічних медичних досліджень, роботи з різнорідними джерелами даних і швидкого прототипування медичних додатків;

– *горизонтальна масштабованість* забезпечує можливість розподілу навантаження між кількома серверами, що критично важливо для великих медичних архітектур, систем з високим навантаженням (наприклад, телемедицина) і обробки потокових медичних даних;

– *висока продуктивність* при роботі з великими обсягами даних досягається за рахунок оптимізації під конкретні операції читання/запису, відсутності складних JOIN-операцій та ефективного кешування;

– *підтримка розподілених систем* дозволяє забезпечувати високу доступність медичних даних, працювати в умовах обмеженого інтернет-з'єднання і реалізовувати георозподілені медичні рішення.

Висновки для медичних застосувань NoSQL БД:

– NoSQL БД ідеальні для роботи з неструктурованими та напівструктурованими медичними даними;

– горизонтальна масштабованість критично важлива для великих медичних систем;

– гнучкість схеми дозволяє швидко адаптуватися до змін у медичних протоколах;

– підходять для роботи з мультимодальними медичними даними (текст, зображення, сигнали).

Хмарні бази даних (Cloud DB) представляють собою сучасний підхід до організації та управління медичними даними, де інфраструктура повністю або частково розміщується в хмарному середовищі (рис. 1.3). Їх фундаментальною особливістю є надання доступу до даних як сервісу (DBaaS - Database as a Service), що дозволяє медичним установам зосередитись на клінічній роботі, а не на адмініструванні інфраструктури.

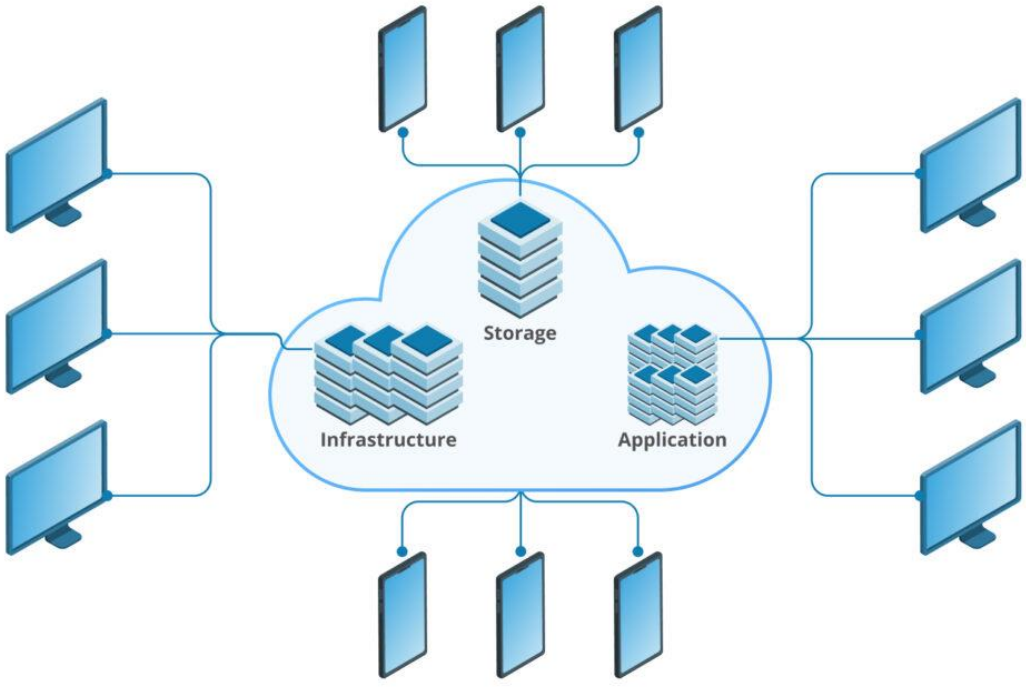


Рисунок 1.3 – Хмарні бази даних

Основні характеристики хмарних баз даних:

– *гнучкі моделі розгортання*. Публічні хмари (AWS, Azure, GCP), приватні хмарні рішення, гібридні архітектури. Для медичних даних особливе значення мають гібридні моделі, де чутливі дані зберігаються локально, а аналітика проводиться в хмарі;

– *автоматичне масштабування*. Динамічне збільшення/зменшення ресурсів, оплата за фактичне використання, георозподіленість даних. Це критично важливо для систем телемедицини, де навантаження може різко зростати;

– *вбудовані механізми безпеки*. Шифрування даних при передачі та зберіганні, резервування в кількох географічних локаціях, інтеграція із системами контролю доступу. Особливо важливо для відповідності вимогам GDPR та українського законодавства про медичну таємницю;

– *інтеграція з AI/ML сервісами*. Готові інструменти для аналітики, можливість тренування медичних моделей, підтримка обробки природної мови. Використовується для предиктивної діагностики та аналізу великих масивів медичних даних.

Висновки для медичних застосувань хмарних БД:

– ідеальний вибір для телемедицини та мобільних медичних додатків;
– ефективне рішення для об'єднання даних з різних медичних установ;
– підтримка складних аналітичних робочих навантажень;
– зменшення витрат на IT-інфраструктуру для приватних клінік.

1.3 Вимоги до баз даних у медицині

Сучасні медичні інформаційні системи висувають суворі вимоги до баз даних, що обумовлено критичною важливістю медичної інформації та її впливом на життя пацієнтів. Ці вимоги можна розглядати через призму трьох

фундаментальних аспектів: безпеки даних, продуктивності системи та здатності до масштабування.

Безпека медичних даних формує основу довіри до електронних систем охорони здоров'я. У контексті баз даних вона реалізується через багаторівневий захист, що включає: технічні засоби (шифрування AES-256 для даних у спокої та під час передачі), адміністративні процедури (регламенти доступу згідно з посадовими обов'язками) та юридичні механізми (відповідність GDPR, HIPAA та українському законодавству про медичну таємницю). Особливу увагу приділяють механізмам аудиту, які фіксують усі дії з даними (хто, коли і які зміни вносив), що критично важливо як для захисту конфіденційності, так і для лікарської відповідальності.

Безпека даних у МІС – це не лише технічна вимога, а й етичний обов’язок. Втрата або витік медичної інформації може призвести до серйозних наслідків: від порушення конфіденційності пацієнтів до шахрайства з медичними страхуваннями. Крім того, відсутність належного аудиту змін у базі даних ускладнює розслідування помилок лікування, що може вплинути на якість надання медичної допомоги.

Продуктивність системи визначає якість медичного обслуговування. Медичні бази даних мають забезпечувати субсекундний час відгуку для критичних операцій (наприклад, доступ до картки пацієнта в екстрених випадках), при цьому обробляючи десятки тисяч запитів одночасно. Це досягається через оптимізацію індексів, кешування частини даних у оперативній пам'яті та використання спеціалізованих форматів зберігання для різних типів медичної інформації (наприклад, колоночне зберігання для аналітичних запитів). Важливим аспектом є підтримка безперервної роботи системи навіть під час пікових навантажень, що особливо актуально для обласних лікарень та систем екстреної медичної допомоги.

Швидкість роботи бази даних безпосередньо впливає на ефективність лікарів. У критичних ситуаціях (наприклад, при надходженні пацієнта з інфарктом або інсультом) затримка у доступі до історії хвороби, алергій або попередніх призначень може коштувати життя. Також низька продуктивність системи знижує

загальну ефективність роботи медичного персоналу – якщо лікар витрачає зайві хвилини на очікування завантаження даних, це зменшує кількість прийнятих пацієнтів і погіршує якість обслуговування.

Масштабованість відображає здатність системи адаптуватися до зростання обсягів даних та кількості користувачів без втрати продуктивності. Для медичних установ це означає необхідність обробляти зростаючі обсяги діагностичних даних (особливо зображень МРТ/КТ), підтримувати географічно розподілені філії та інтегрувати нові джерела даних (наприклад, IoT-пристрої для моніторингу пацієнтів). Сучасні підходи до масштабування включають горизонтальне розширення (додавання нових серверів), використання мікросервісної архітектури та автоматичне балансування навантаження.

Масштабованість визначає довгострокову життєздатність медичної інформаційної системи. Якщо база даних не може ефективно масштабуватися, це призводить до:

– зростання витрат на підтримку застарілої інфраструктури;

– зниження швидкості роботи системи при збільшенні кількості пацієнтів;

– складностей у впровадженні нових технологій (наприклад, штучного інтелекту для діагностики);

– проблем із сумісністю при інтеграції з іншими медичними системами.

Наприклад, якщо обласна лікарня не має можливості масштабувати базу даних, це може призвести до неможливості підключення нових філій або додавання функцій телемедицини. У результаті пацієнти віддалених районів отримують менш якісну медичну допомогу.

Таким чином, безпека, швидкість і масштабованість баз даних – це не просто технічні параметри, а ключові фактори, що безпосередньо впливають на якість медичного обслуговування, ефективність роботи лікарів і захист прав пацієнтів. Відповідність цим вимогам дозволяє створити стабільну, надійну та ефективну систему охорони здоров’я, здатну ефективно функціонувати як у мирний час, так і в умовах надзвичайних ситуацій.

2 СПЕЦИФІКА ВИКОРИСТАННЯ БАЗ ДАНИХ У МЕДИЧНИХ СИСТЕМАХ

2.1 Приклади медичних інформаційних систем з реалізованими базами даних

Сучасний ринок медичних інформаційних систем пропонує різноманітні рішення для роботи з даними, кожне з яких має унікальні архітектурні особливості та підходи до організації баз даних. Міжнародні лідери, такі як EPIC та Cerner, демонструють високий рівень інтеграції складних БД-рішень, тоді як українські локальні розробки відзначаються адаптацією до специфіки національної системи охорони здоров'я.

Приклади міжнародних систем:

– *EPIC Systems* (США) реалізує комплексний підхід до управління медичними даними через власну реляційну базу на основі технології InterSystems Caché (рис. 2.1). Ця система відзначається глибокою інтеграцією модулів (від запису на прийом до управління стаціонаром), вбудованими інструментами клінічної аналітики та підтримкою стандартів HL7 FHIR (Fast Healthcare Interoperability Resources) для обміну медичними даними, завдяки якому медичні інформаційні системи можуть ділитися важливою інформацією про здоров’я пацієнтів і працювати разом, навіть якщо створені різними компаніями.

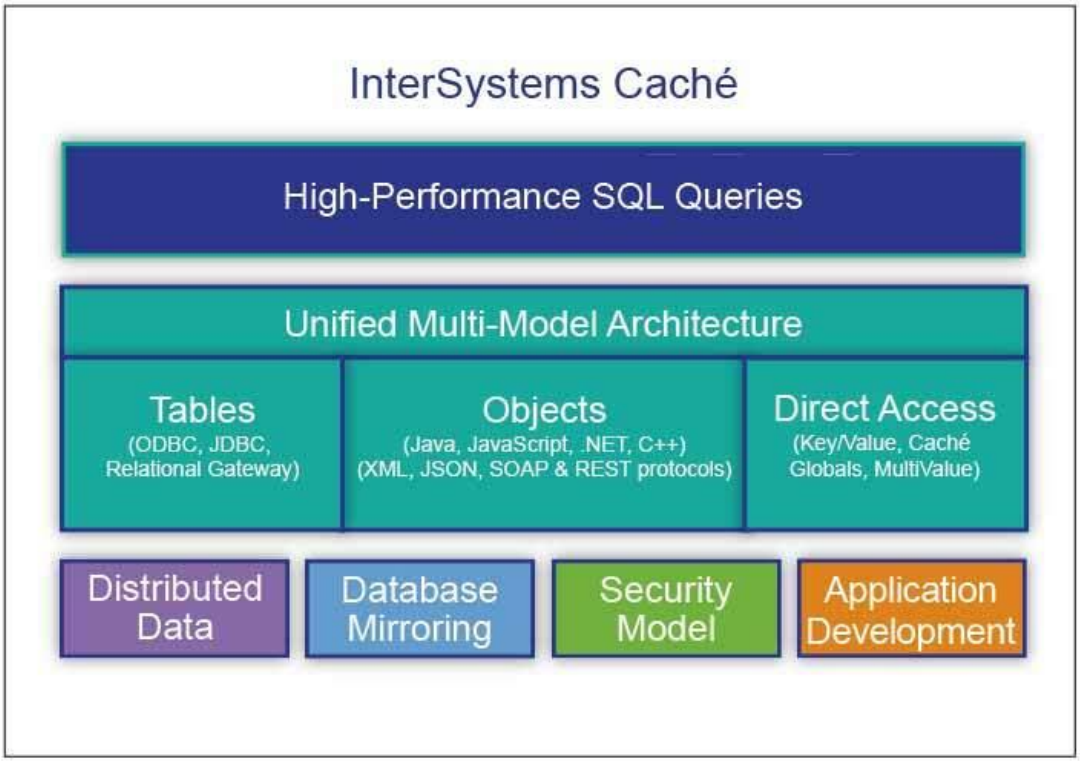


Рисунок 2.1 – Архітектура InterSystems Caché

– *Cerner Millennium* (США) використовує гібридний підхід, поєднуючи реляційну Oracle Database для структурованих клінічних даних і нереляційну MongoDB для роботи з діагностичними зображеннями (рис. 2.2), а також власні механізми індексації для прискорення пошуку.

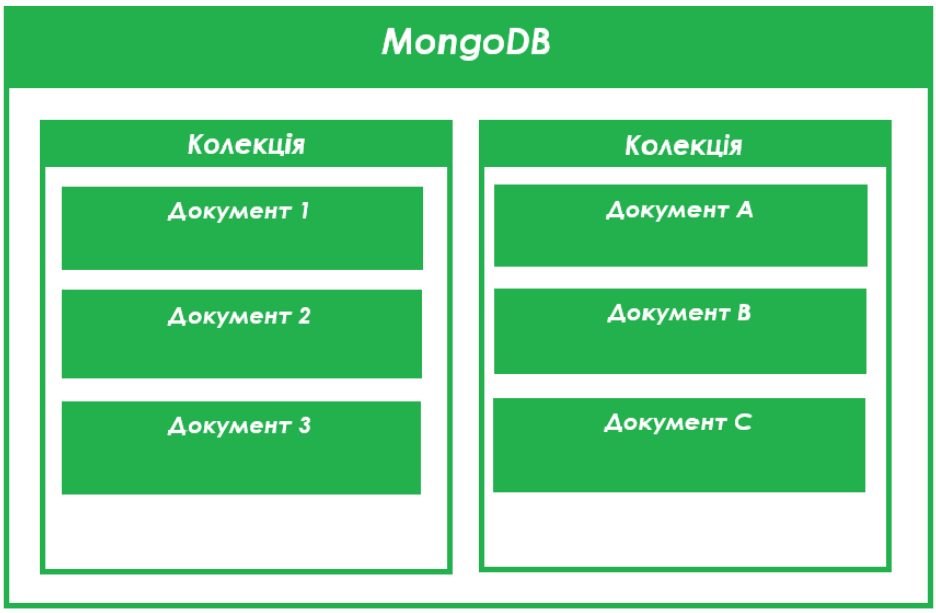


Рисунок 2.2 – Структура MongoDB

Система eHealth – це державна електронна платформа, створена в Україні для централізації та цифровізації процесів охорони здоров'я. Вона покликана забезпечити ефективне управління медичними даними, підвищити якість медичного обслуговування, знизити адміністративне навантаження на медичний персонал та забезпечити пацієнтам зручний доступ до власної медичної інформації.

Поняття та архітектура системи eHealth. eHealth є складною розподіленою інформаційною системою, що складається з центрального ядра та низки зовнішніх сервісів, які взаємодіють між собою через захищені API. Центральне ядро відповідає за зберігання та обробку основних медичних даних у реляційній базі даних, яка оптимізована для роботи з великими обсягами інформації та швидкого доступу.

Архітектура системи побудована за принципом мікросервісів, що дозволяє масштабувати кожен компонент окремо, забезпечувати відмовостійкість та гнучкість оновлень. Застосовується кластеризація баз даних для забезпечення високої доступності, а також багаторівневе кешування для пришвидшення обробки запитів.

Для забезпечення надійності використовуються механізми резервного копіювання даних, а також системи моніторингу стану серверів і баз даних, що дозволяють оперативно реагувати на можливі збої.

Функціональні модулі системи. Система eHealth складається з кількох ключових функціональних модулів, кожен з яких має власну роль:

– *реєстр пацієнтів*. Зберігає персональні дані (ПІБ, дату народження, ідентифікаційний код), медичну історію, записи про візити до лікарів, результати аналізів та обстежень. Модуль підтримує зв'язок із зовнішніми системами для імпорту та експорту даних.

– *реєстр медичних закладів і лікарів*. Містить інформацію про всі учасники системи: лікарні, поліклініки, окремих лікарів і медичний персонал. Це дозволяє централізовано керувати ролями, правами доступу та адміністративною інформацією.

– *модуль електронних рецептів і направлень*. Забезпечує створення, зберігання та передачу електронних медичних документів, таких як рецепти на ліки, направлення на обстеження чи лікування. Це значно скорочує паперовий документообіг і дозволяє відстежувати виконання призначень.

– *аналітичний модуль*. Агрегує статистичні дані про використання медичних послуг, поширені діагнози, ефективність лікування та інші показники. Ця інформація використовується для контролю якості медичного обслуговування та прийняття управлінських рішень.

Заходи безпеки у системі eHealth. Безпека медичних даних – один з головних пріоритетів системи eHealth. Для цього впроваджені багаторівневі заходи захисту:

– *шифрування даних*. Забезпечує конфіденційність інформації як при зберіганні, так і під час передачі між серверами та користувачами. Це унеможливлює перехоплення та несанкціонований доступ до чутливих даних.

– *аутентифікація та авторизація користувачів*. Реалізована через використання електронних підписів, двофакторної автентифікації та ролей користувачів. Це гарантує, що лише уповноважені особи мають доступ до конкретних даних або функцій системи.

– *ведення журналів доступу та змін*. Кожна дія користувача у системі фіксується в спеціальних логах, що дозволяє провести аудит, виявити потенційні порушення або зловживання;

– *резервне копіювання та відновлення даних*. Регулярне створення копій баз даних забезпечує можливість швидкого відновлення інформації у випадку технічних збоїв, атак або людських помилок.

Інтеграція з іншими системами. Однією з важливих переваг eHealth є її здатність до інтеграції з різноманітними зовнішніми інформаційними системами (рис. 2.3):

– *лікарняні інформаційні системи (ЛІС)*. eHealth обмінюється даними з локальними системами медичних закладів, що дозволяє уникнути дублювання інформації та забезпечує актуальність даних в режимі реального часу.

– *аптечні мережі*. Електронні рецепти автоматично передаються до аптек, що спрощує видачу ліків та запобігає підробці рецептів.

– *лабораторії та діагностичні центри*. Результати аналізів та обстежень інтегруються безпосередньо в електронні медичні картки пацієнтів.

– *інші державні реєстри*. Реєстри страхування, демографічні бази даних, що дозволяє підвищити точність і повноту медичної інформації.

Іншими словами, система eHealth чудово і гнучко інтегрована у інші інформаційні системи, пов'язані і не пов'язані з охороною здоров'я. Завдяки такому рівню інтеграції забезпечується зручність і безперервність медичного обслуговування, підвищується оперативність прийняття рішень та знижується ймовірність помилок через людський фактор.



Рисунок 2.3 – Рівень інтеграції системи eHealth

2.2 Нормативно-правове регулювання захисту медичних даних

Збирання, обробка, зберігання та передача медичних даних є не лише технічною, а й юридичною проблематикою, що потребує чіткого дотримання законодавчих вимог. Будь-яка медична інформаційна система (МІС), яка працює з

персональними даними пацієнтів, повинна відповідати правилам захисту, що гарантують конфіденційність, цілісність і контрольований доступ до інформації. Основними орієнтирами у цій сфері є міжнародні стандарти, такі як Загальний регламент захисту даних (GDPR) у Європейському Союзі, Закон HIPAA у США, а також національне законодавство України.

GDPR (General Data Protection Regulation). Європейська модель захисту персональних даних. Загальний регламент захисту даних (GDPR), що набув чинності у травні 2018 року, встановлює обов'язкові вимоги до обробки персональних даних у межах ЄС. Медичні дані в ньому класифікуються як чутливі, тобто такі, що потребують особливого захисту. Ключові принципи регламенту – законність, мінімізація обробки, обмеження зберігання, прозорість та цілісність (рис. 2.4). У практиці МІС це означає: бази даних повинні надавати лише необхідний доступ (принцип мінімальних привілеїв), реалізовувати контроль доступу, шифрування, журналювання (логування) дій, а також забезпечувати права пацієнтів – зокрема, право на виправлення чи видалення даних.



Рисунок 2.4 – Ключові принципи GDPR

HIPAA (Health Insurance Portability and Accountability Act). Стандарти безпеки в США. Американський Закон HIPAA (Health Insurance Portability and Accountability Act), ухвалений у 1996 році, став основним регулятором захисту електронної медичної інформації (ePHI). HIPAA визначає три групи вимог: адміністративні, фізичні та технічні заходи безпеки. Для розробників MIC це означає обов'язкову реалізацію процедур аутентифікації, контролю доступу, шифрування даних при передачі і на носіях, регулярне резервне копіювання, аудит дій користувачів (рис. 2.5). Закон також передбачає високі штрафи за порушення – до мільйонів доларів у випадку витоку або втрати даних.



Рисунок 2.5 – Заходи безпеки HIPAA

Національне законодавство України у сфері захисту медичних даних. В Україні ключовим нормативним актом є Закон «Про захист персональних даних», який запроваджує базові принципи обробки даних: згода суб'єкта, обмеження доступу, захист від несанкціонованої обробки. Медичні дані визнані чутливими, а отже потребують особливого режиму. Закон вимагає технічних і організаційних заходів безпеки, зокрема обмеження прав доступу, фізичний захист серверів, шифрування тощо. Крім того, діють галузеві нормативи – зокрема, наказ МОЗ

України № 411, який регламентує порядок ведення електронної медичної картки та накладає вимоги до безпеки медичних записів.

eHealth та поступове наближення до міжнародних стандартів. У межах цифровізації охорони здоров'я в Україні поступово впроваджується система eHealth, яка включає елементи нормативної адаптації до стандартів GDPR та HIPAA. Зокрема, запроваджено обов'язкове логування доступу, формування уніфікованих інтерфейсів обміну, обмеження прямого доступу до персональних даних лікарями лише через рольову модель. На рівні архітектури системи формуються технічні вимоги до структур зберігання, резервного копіювання, автентифікації та ведення журналів змін – усе це безпосередньо стосується і побудови БД у таких системах.

Порушення нормативних вимог і юридичні наслідки. Недотримання зазначених стандартів тягне за собою не лише технічні ризики, а й юридичну відповідальність. GDPR передбачає штрафи до 20 мільйонів євро або 4% річного обігу компанії. У США HIPAA дозволяє накладати багаторівневі санкції – як за умисні, так і за ненавмисні порушення. В Україні відповідальність передбачена у вигляді адміністративних стягнень, однак у контексті розвитку eHealth розглядається запровадження більш жорстких правил аудиту безпеки.

Таким чином, нормативно-правова база є невіддільним елементом проєктування та впровадження баз даних у медичних інформаційних системах. Вона формує фундамент безпеки, надійності та відповідальності, а також створює умови для міжнародної сумісності, що набуває дедалі більшого значення в умовах глобалізації цифрової медицини.

2.3 Проблеми впровадження: конфіденційність, інтеграція з обладнанням

Незважаючи на численні переваги використання баз даних у медичних інформаційних системах, їх впровадження супроводжується низкою серйозних викликів. Найбільш критичними з них є забезпечення конфіденційності медичних даних та технічна інтеграція інформаційної системи з медичним обладнанням. Ці аспекти вимагають не лише технологічних рішень, а й стратегічного бачення щодо політик, стандартів, безпеки та етичної відповідальності.

Проблеми конфіденційності медичних даних. Конфіденційність медичної інформації є основою довіри між пацієнтом і медичним закладом. Усі персональні медичні записи – включаючи історії хвороб, результати аналізів, рецепти, дані про алергії, процедури – є надзвичайно чутливими. Будь-яке їхнє несанкціоноване розкриття може призвести до репутаційних, фінансових і навіть кримінальних наслідків для медичного закладу.

Сучасні МІС зазвичай функціонують у середовищі, що передбачає інтеграцію з хмарними технологіями та мобільними додатками. Це, з одного боку, забезпечує високу доступність даних, з іншого – створює вразливості до кібератак, витоків даних та несанкціонованого доступу. Уразливими залишаються канали передачі інформації між локальними базами даних, віддаленими серверами та інтерфейсами для лікарів і пацієнтів.

Крім технічних вразливостей, важливими є й організаційні ризики – недостатня підготовка персоналу, неуніфіковані політики доступу до даних, відсутність багатофакторної аутентифікації. У багатьох закладах практикується надмірно широкий доступ до баз даних без належного розмежування прав, що суттєво підвищує ризик витоку конфіденційної інформації.

Особливого значення набуває дотримання міжнародних і національних нормативів, оскільки невідповідність їм може блокувати участь установи в міжнародних проєктах або залучення страхових компаній (рис. 2.6). Проблемою

залишається і відсутність єдиного протоколу реагування на інциденти кібербезпеки в закладах охорони здоров'я.

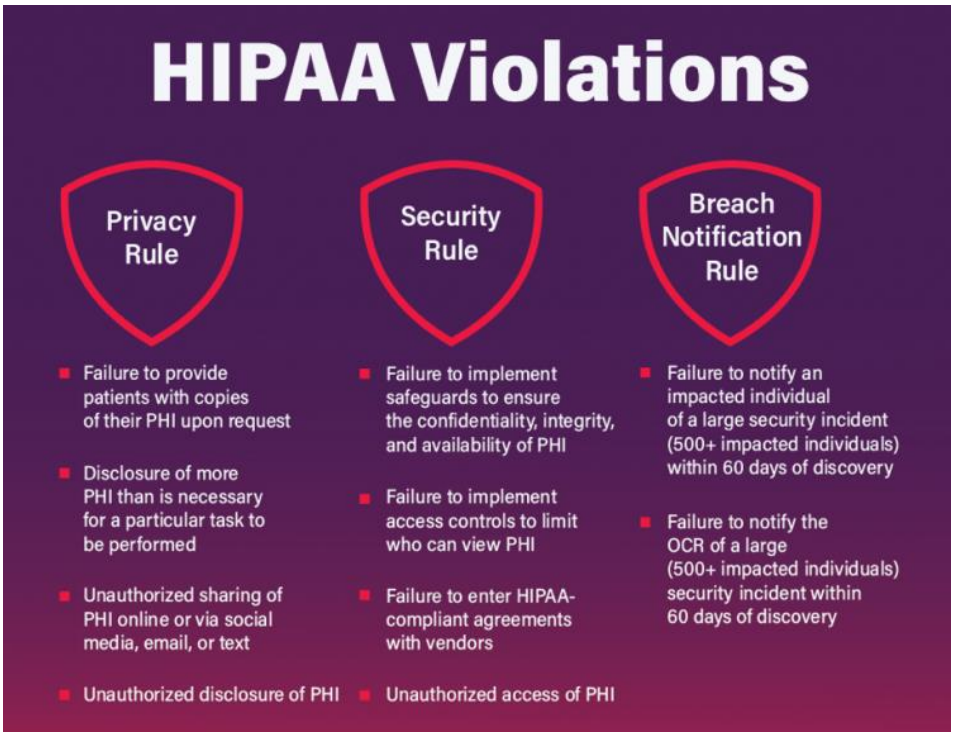


Рисунок 2.6 – Порушення стандартів HIPAA

Інтеграція з медичним обладнанням. Інтеграція інформаційних систем із медичними приладами є критичною умовою повноцінного функціонування МІС. Йдеться про автоматичне зчитування та передачу результатів діагностичних обстежень (електрокардіограми, МРТ, КТ, аналізи крові) до електронної медичної карти. Проте така інтеграція ускладнена через кілька чинників:

– Існує величезна різноманітність медичного обладнання, виготовленого різними виробниками, з власними протоколами обміну даними. Більшість з них не підтримує відкриті стандарти (наприклад, HL7 чи FHIR), або ж реалізують їх з відхиленнями. У таких випадках інтеграція вимагає розробки специфічних драйверів, шлюзів або програмних адаптерів;

– Частина обладнання, особливо в державних закладах, є застарілим. Воно або не має цифрових інтерфейсів взагалі, або підтримує лише протоколи

обміну даними, які несумісні з сучасними базами даних і системами керування ними;

– Відсутність стандартизованого підходу до інтерпретації отриманих показників ускладнює автоматичний аналіз інформації. Наприклад, результати ультразвукового дослідження виводяться у вигляді зображень, які важко піддаються автоматизованій обробці без спеціального ПЗ.

Ще однією проблемою є безперервна робота пристроїв. Усі етапи інтеграції потрібно проводити без зупинки основного медичного процесу – обстеження, діагностики чи лікування. Це створює додаткове навантаження на ІТ-відділи, які змушені працювати з обладнанням «на ходу», уникаючи простоїв.

Зрештою, важливим викликом стає питання кіберзахисту при інтеграції. Уразливості в підключеному обладнанні (особливо коли воно працює на старих прошивках або з відкритими портами) можуть стати воротами для атак на всю медичну ІС. Звичайне зараження вірусом через інтерфейс USB може дати змогу зловмиснику отримати доступ до локальної мережі, а отже – і до бази даних.

Таким чином, попри значні технологічні досягнення в сфері медичних інформаційних систем, реальні сценарії впровадження демонструють численні практичні бар'єри. Забезпечення конфіденційності даних потребує не лише впровадження технічних засобів захисту, але й глибокої нормативної та організаційної підготовки. У той час як ефективна інтеграція з обладнанням залежить від сумісності, стандартизації та безпеки інфраструктури. Подолання цих проблем є ключовою умовою для повноцінного переходу до цифрової медицини.

2.4 Оцінка продуктивності, масштабованості та безпеки різних типів баз даних

У процесі проєктування та впровадження медичних інформаційних систем особливу роль відіграє правильний вибір типу бази даних. Медичні заклади

працюють з чутливою інформацією про пацієнтів, обсяг якої постійно зростає, а запити на швидкість доступу та безпечне зберігання – стають дедалі вищими. У таких умовах критичними стають три характеристики баз даних: продуктивність, масштабованість і безпека. Кожен тип БД – реляційна, документно-орієнтована чи хмарна – має свої сильні та слабкі сторони у цих аспектах. У цьому підрозділі буде здійснено порівняльний аналіз баз даних відповідно до зазначених критеріїв, що дозволить визначити доцільність використання конкретного типу БД залежно від потреб медичної установи.

Продуктивність. Продуктивність бази даних у межах медичної інформаційної системи визначає її здатність обробляти великий обсяг запитів від користувачів із мінімальною затримкою. У медичному контексті це означає швидкий доступ до електронних медичних карток, результатів обстежень, історії призначень, та іншої критично важливої інформації. Високі вимоги до продуктивності обумовлені не лише кількістю одночасних користувачів, а й потребою у стабільній роботі під час пікових навантажень, наприклад, у періоди епідеміологічної напруги чи у великих лікарнях.

– Реляційні бази даних (наприклад, MySQL, PostgreSQL) традиційно використовуються у більшості МІС завдяки своїй зрілості, структурованості та підтримці транзакцій. Вони показують високу продуктивність при роботі з чітко визначеними схемами даних, де структура інформації є стабільною та ієрархічною (наприклад, персональні дані пацієнтів, результати лабораторних аналізів). Проте, при зростанні обсягів даних або кількості паралельних запитів продуктивність таких систем може знижуватись через обмеження вертикального масштабування та складність оптимізації SQL-запитів у складних схемах;

– NoSQL-системи, зокрема документно-орієнтовані (наприклад, MongoDB), забезпечують вищу продуктивність у сценаріях, де дані мають менш чітку структуру або часто змінюються. Вони особливо ефективні для зберігання великих обсягів неструктурованої або напівструктурованої інформації, як-от дані зі смартпристроїв, журналів спостережень чи записів моніторингу пацієнтів у режимі реального часу. Завдяки можливості горизонтального масштабування

(розподіл навантаження на кілька серверів) такі бази демонструють стабільну швидкодію навіть при значному розширенні системи;

– Хмарні бази даних (наприклад, Amazon Aurora, Google Cloud Firestore) поєднують у собі переваги як SQL-, так і NoSQL-підходів. Їхня продуктивність забезпечується за рахунок автоматичного масштабування ресурсів, балансування навантаження та розподілу збереження даних між кількома датацентрами. У хмарних МІС система може адаптуватися до навантаження в реальному часі без втрати швидкості, що є вирішальним для телемедичних сервісів та віддаленого моніторингу пацієнтів. Крім того, такі рішення часто пропонують вбудовану оптимізацію запитів та індексацію, що знижує час очікування на результати запиту.

Таким чином, у контексті продуктивності:

– реляційні БД є ефективними для структурованих медичних записів, але менш гнучкими до масштабування;

– NoSQL-рішення забезпечують високу швидкість при роботі з гнучкими або об’ємними даними;

– хмарні БД забезпечують найвищий рівень продуктивності у сценаріях з великою кількістю запитів і змінним навантаженням.

Масштабованість бази даних – це її здатність ефективно адаптуватися до зростання обсягів даних та кількості користувачів без суттєвого зниження продуктивності чи зміни архітектури системи (рис. 2.7). Для медичних інформаційних систем ця характеристика є критично важливою: медична установа може розширюватися, впроваджувати нові модулі, інтегрувати нові джерела даних, а також підключати нових користувачів – лікарів, пацієнтів, лабораторії, мобільні пристрої тощо. Усі ці зміни призводять до експоненційного зростання обсягів даних, які потрібно обробляти у реальному часі.

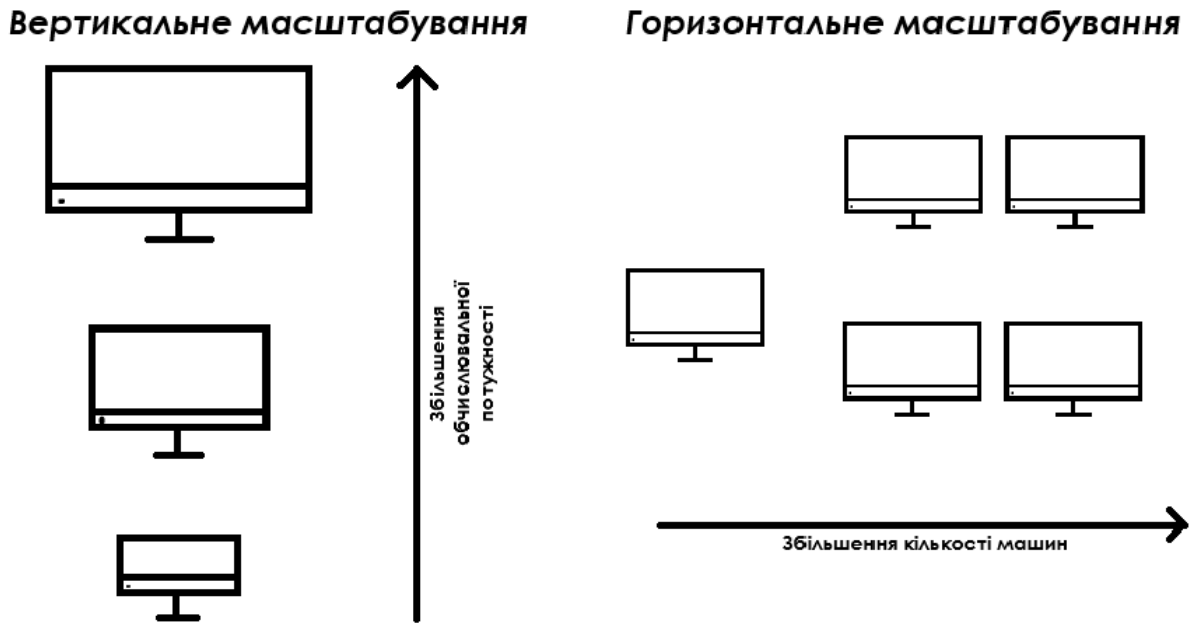


Рисунок 2.7 – Види масштабування

Реляційні бази даних (SQL) історично мають обмежену масштабованість. Вони розраховані на вертикальне масштабування, тобто підвищення потужності одного сервера (наприклад, додавання оперативної пам’яті або CPU). Проте такий підхід має свої межі: апаратні ресурси мають фізичні обмеження, а перенесення на потужніші сервери – високі витрати. Крім того, масштабування реляційної системи вимагає обережної роботи з транзакціями та збереження цілісності даних, що ускладнює адаптацію до розширення.

NoSQL-системи, навпаки, спроєктовані з урахуванням горизонтального масштабування. Це означає, що система може легко розподіляти дані між багатьма вузлами (серверами), автоматично або з мінімальною участю адміністратора. Для медичних інформаційних систем це дозволяє масштабуватися «вшир» при зростанні числа підключених пристроїв або віддалених користувачів, не змінюючи архітектуру БД. Наприклад, дані з численних медичних кабінетів, мобільних застосунків або IoT-пристроїв можуть безперешкодно зберігатися у документно-орієнтованій базі даних, яка розподіляється між датацентрами.

Хмарні бази даних (Cloud SQL, Firebase, Amazon DynamoDB тощо) представляють найвищий рівень масштабованості. Їх інфраструктура динамічно

масштабовується без втручання користувача – додаткові обчислювальні ресурси або сховища додаються автоматично у відповідь на зростання навантаження. Такі рішення особливо ефективні в умовах, коли медична система повинна працювати з піковими навантаженнями, наприклад, у періоди пандемій або під час масових вакцинацій. Хмарні платформи також підтримують реплікацію даних у різних регіонах, що дозволяє організувати стабільну роботу для розподілених медичних установ.

У контексті медичних МІС можна зробити такі висновки:

– Реляційні БД мають обмежену масштабованість, що потребує додаткових ресурсів і зусиль при розширенні;

– NoSQL-системи краще підходять для динамічних і зростаючих обсягів медичних даних завдяки природному горизонтальному масштабуванню;

– Хмарні БД забезпечують автоматичне масштабування, що у контексті масштабованості робить їх оптимальним вибором для МІС, орієнтованих на гнучкість, стабільність і безперервну доступність.

Безпека баз даних у медичних інформаційних системах (МІС) є не лише технічною вимогою, а й правовим зобов'язанням, оскільки робота з персональними медичними даними напряму пов'язана з приватністю, етичними принципами та дотриманням міжнародних стандартів захисту інформації. Помилки у цій сфері можуть призвести до витоку конфіденційної інформації, фінансових штрафів, юридичної відповідальності та втрати довіри пацієнтів.

Реляційні SQL-системи зазвичай мають добре продуману модель безпеки, засновану на:

– ролевому керуванні доступом (RBAC) – це дозволяє налаштувати рівні прав для кожного типу користувача (лікар, адміністратор, лаборант) (рис. 2.8);

– транзакційній цілісності – що унеможливлює неконсистентне оновлення критичних медичних записів;

– механізмах журналювання – що дозволяє відслідковувати дії користувачів і створювати звіти для аудиту.

Однак захист у SQL-системах сильно залежить від локального адміністрування, і неправильна конфігурація або людський фактор можуть нівелювати ці переваги. Крім того, централізоване зберігання даних може стати вразливістю у разі фізичного доступу або внутрішнього інсайду.

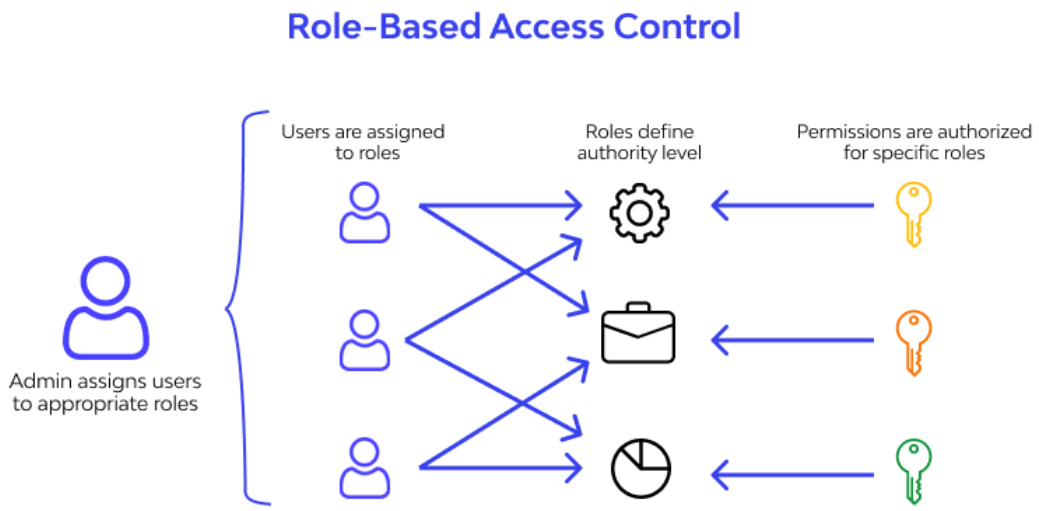


Рисунок 2.8 – Схема керування доступом на основі ролей (RBAC)

NoSQL-рішення, особливо у своїх перших реалізаціях, приділяли безпеці менше уваги, віддаючи пріоритет масштабованості й швидкості. Проте з розвитком систем ситуація покращилася. Сучасні NoSQL-сервери (MongoDB, Cassandra тощо) підтримують:

– шифрування даних як під час передавання (TLS), так і при зберіганні (AES);

– автентифікацію та контроль доступу;

– аудит змін і інтеграцію з зовнішніми системами керування доступом (LDAP, Active Directory).

Втім, розподілений характер NoSQL-систем може ускладнювати централізовану безпеку: наприклад, необхідно контролювати доступ на кожному вузлі, що створює додаткові точки ризику.

Хмарні сервіси, зокрема Amazon RDS, Google Cloud Firestore чи Azure Cosmos DB, є лідерами в контексті інформаційної безпеки. Вони надають:

– автоматичне шифрування даних (на рівні диска, з'єднань і API);

– георезервування та багаторівневу автентифікацію;

– інтеграцію з протоколами безпеки HIPAA, GDPR, ISO/IEC 27001;

– автоматичне оновлення систем безпеки – що особливо актуально у випадках виявлення нових вразливостей.

Однак важливо пам’ятати, що навіть у хмарі кінцева відповідальність за конфігурацію захисту лежить на власнику медичного застосунку або установи. Помилки у налаштуванні дозволів, API-доступів або авторизації можуть бути критичними, попри високий рівень базової безпеки.

Порівняння різних типів баз даних за критеріями продуктивності, масштабованості та безпеки дозволяє зробити низку важливих висновків, релевантних для проєктування медичних інформаційних систем (табл. 2.1).

– Реляційні бази даних (SQL) демонструють високу стабільність при роботі з чітко структурованими даними, які характерні для медичних систем: пацієнтські записи, історії хвороб, призначення, звіти тощо. Завдяки підтримці транзакцій, обмежень цілісності та складних запитів, вони ідеально підходять для забезпечення точності та консистентності, що є критично важливим у медичному контексті;

– Документно-орієнтовані бази (NoSQL) справляються краще при обробці великих обсягів неструктурованих або напівструктурованих даних, однак поступаються SQL-системам у забезпеченні складної логіки зв’язків між сутностями, яка часто необхідна в клінічних системах;

– Хмарні бази даних, у свою чергу, пропонують гнучкість і масштабованість, проте залежать від зовнішнього постачальника, що може стати обмеженням у разі суворих вимог до безпеки та локального зберігання медичних даних.

З огляду на поставлену мету дослідження та формат практичного завдання, доцільно обрати саме реляційну базу даних, яка дозволяє легко моделювати типову медичну структуру даних, не вимагає складного розгортання та має добре розвинені засоби для реалізації базових вимог безпеки. Такий підхід не лише

відповідає вимогам до простоти реалізації, але й дозволяє зберегти академічну валідність демонстрації.

Таблиця 2.1 – Порівняння типів баз даних

Критерій	Реляційні БД (SQL)	Нереляційні БД (NoSQL)	Хмарні БД
Продуктивність	Висока для структурованих даних і складних запитів; оптимізовані для транзакцій	Висока при роботі з великими обсягами неструктурованих даних	Залежить від провайдера; зазвичай висока завдяки автоматичному масштабуванню
Масштабованість	Вертикальна; масштабування обмежене апаратними ресурсами	Горизонтальна; добре масштабуються за рахунок розподіленої архітектури	Автоматична горизонтальна масштабованість, гнучка конфігурація
Безпека	Розвинені механізми контролю доступу, транзакційність, шифрування	Можуть мати обмежену підтримку складної безпеки, залежить від реалізації	Зазвичай включають вбудовані засоби шифрування, аудит, але залежать від постачальника

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ БАЗИ ДАНИХ

3.1 Постановка завдання та вибір інструментів

У межах практичної частини дослідження передбачається створення простої реляційної бази даних, що моделює базовий функціонал МІС. Такий підхід дозволяє на практиці продемонструвати, як може бути реалізовано зберігання та обробка медичних даних за допомогою сучасних СУБД, а також проілюструвати ключові переваги реляційного підходу, зокрема в контексті структурування інформації, забезпечення зв'язків між сутностями та простоти запитів.

Мета реалізації – створити базу даних для умовної МІС, яка міститиме чотири основні сутності: «Пацієнти», «Лікарі», «Прийоми» та «Медичні записи». Ця структура дозволяє імітувати типові процеси зберігання інформації про користувачів системи, медичний персонал і результати їхніх взаємодій. У результаті буде продемонстровано процес створення таблиць, встановлення зв'язків між ними, наповнення даними та виконання запитів до бази.

Пріоритетом при виборі інструментів стали такі критерії:

– максимальна простота встановлення і використання;
– мінімальні системні вимоги;
– наявність графічного інтерфейсу для візуалізації структури БД;
– відсутність необхідності в налаштуванні серверної частини;
– використання лише локальних ресурсів без підключення до мережі.

Такі параметри дозволять ефективно і просто створити базу даних, що повністю відповідає вимогам практичної частини і має змогу в повній мірі продемонструвати свої основні функції.

SQLite – легка, вбудована реляційна система управління базами даних з відкритим вихідним кодом, що повною мірою задовольняє усі зазначені вимоги. Вона не потребує окремого серверного оточення чи встановлення додаткових

служб. Зберігається у вигляді одного локального файлу, що дозволяє швидко організувати обробку даних без складних конфігурацій. Попри свою простоту, SQLite підтримує переважну більшість функцій SQL, включаючи транзакції, зовнішні ключі, індекси та запити з агрегацією, що робить її ідеальним кандидатом для демонстраційної реалізації.

Таким чином, SQLite дозволяє швидко й ефективно реалізувати практичну частину роботи, зосередившись саме на логіці побудови бази даних і основних процесах роботи з нею.

3.2 Проєктування та створення бази даних

Після аналізу вимог до зберігання та обробки даних у медичних інформаційних системах, було розроблено структуру бази даних, яка відображає ключові сутності системи: Пацієнти, Лікарі, Прийоми та Медичні записи. У цьому підрозділі наведено покроковий опис створення такої бази з використанням SQLite Studio – зручного інструменту для локального тестування структури та логіки БД. На комп’ютер було завантажено та встановлено DB Browser for SQLite (рис. 3.1).

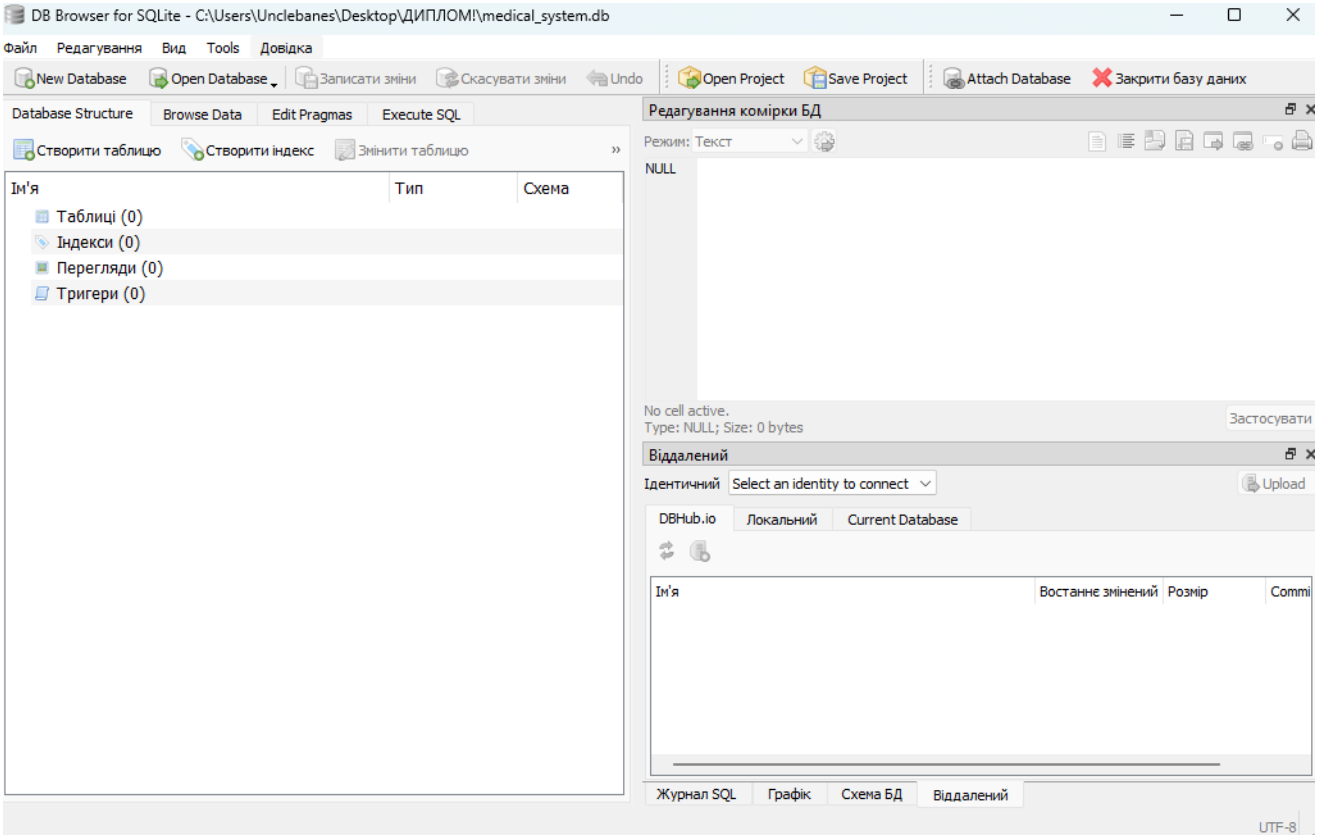


Рисунок 3.1 – Інтерфейс SQLite

Створення нової бази даних. У головному меню DB Browser for SQLite обрано пункт "New Database". Вказано місце збереження, назву файлу “medical_system.db”. База даних зберігається локально у вигляді одного файлу.

Створення таблиць. Усі таблиці ми будемо створювати вручну через SQL-редактор для точного контролю над структурою:

1) Таблиця «patients», має атрибути: «id» – первинний ключ, автоматично зростає, забезпечує унікальність кожного запису, «birth_date» – тип «DATE», що в даному випадку використовується для зберігання дати народження, «gender» – текстове поле, що дозволяє зберігати стать (чоловік, жінка, інше), «phone» – поле для номера телефону, передбачає можливість швидкого зв’язку (рис. 3.2).

```sql
CREATE TABLE patients (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    full_name TEXT NOT NULL,
    birth_date DATE,
    gender TEXT,
    phone TEXT
);
```

Рисунок 3.2 – SQL-код для створення таблиці «patients»

2) Таблиця «doctors», структура схожа на «patients», однак додається поле «specialty» для вказання спеціалізації лікаря (терапевт, кардіолог тощо), що дозволяє надалі фільтрувати або класифікувати лікарів у запитах (рис. 3.3).

```sql
CREATE TABLE doctors (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    full_name TEXT NOT NULL,
    specialty TEXT,
    phone TEXT
);
```

Рисунок 3.3 – SQL-код для створення таблиці «doctors»

3) Таблиця «appointments», реалізує зв’язок «багато до багатьох» між пацієнтами та лікарями через подію – прийом. «patient_id» та «doctor_id» – зовнішні ключі, що вказують на відповідні таблиці «patients» та «doctors». Використання зовнішніх ключів гарантує, що прийом не може бути створений для неіснуючого пацієнта чи лікаря. «appointment_date» – дозволяє вказати точну дату прийому, а «reason» – короткий опис причини звернення (рис. 3.4).

```
CREATE TABLE appointments (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    patient_id INTEGER,
    doctor_id INTEGER,
    appointment_date DATE,
    reason TEXT,
    FOREIGN KEY (patient_id) REFERENCES patients(id),
    FOREIGN KEY (doctor_id) REFERENCES doctors(id)
);
```

Рисунок 3.4 – SQL-код для створення таблиці «appointments»

4) Таблиця «medical_records», містить історію хвороб пацієнтів. «record_date» дозволяє вказати, коли було зроблено запис, «diagnosis» і «treatment» – текстові поля для фіксації результатів обстеження та призначеного лікування, також вказується лише «patient_id», бо не завжди є потреба пов’язувати запис з конкретним лікарем (особливо при імпорті архівів або зовнішніх записів).

```
CREATE TABLE medical_records (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    patient_id INTEGER,
    record_date DATE,
    diagnosis TEXT,
    treatment TEXT,
    FOREIGN KEY (patient_id) REFERENCES patients(id)
);
```

Рисунок 3.5 – SQL-код для створення таблиці «medical_records»

В результаті ми отримали готову просту медичну базу даних зі своїми зв’язками. У вкладці «Database Structure» можна переглянути дерево всіх таблиць і переконатися, що вони налаштовані правильно (рис. 3.6), а для більшої зручності за допомогою вебресурсу dbdiagram.io було побудовано візуальну діаграму зв’язків між сутностями бази (рис. 3.7).

Ім'я	Тип	Схема
Таблиці (5)		
appointments		CREATE TABLE appointments (id INTEGE
id	INTEGER	"id" INTEGER
patient_id	INTEGER	"patient_id" INTEGER
doctor_id	INTEGER	"doctor_id" INTEGER
appointment_date	DATE	"appointment_date" DATE
reason	TEXT	"reason" TEXT
doctors		CREATE TABLE doctors (id INTEGER PRI
id	INTEGER	"id" INTEGER
full_name	TEXT	"full_name" TEXT NOT NULL
specialty	TEXT	"specialty" TEXT
phone	TEXT	"phone" TEXT
medical_records		CREATE TABLE medical_records (id INT
id	INTEGER	"id" INTEGER
patient_id	INTEGER	"patient_id" INTEGER
record_date	DATE	"record_date" DATE
diagnosis	TEXT	"diagnosis" TEXT
treatment	TEXT	"treatment" TEXT
patients		CREATE TABLE patients (id INTEGER PRI
id	INTEGER	"id" INTEGER
full_name	TEXT	"full_name" TEXT NOT NULL
birth_date	DATE	"birth_date" DATE
gender	TEXT	"gender" TEXT
phone	TEXT	"phone" TEXT

Рисунок 3.6 – Дерево таблиць

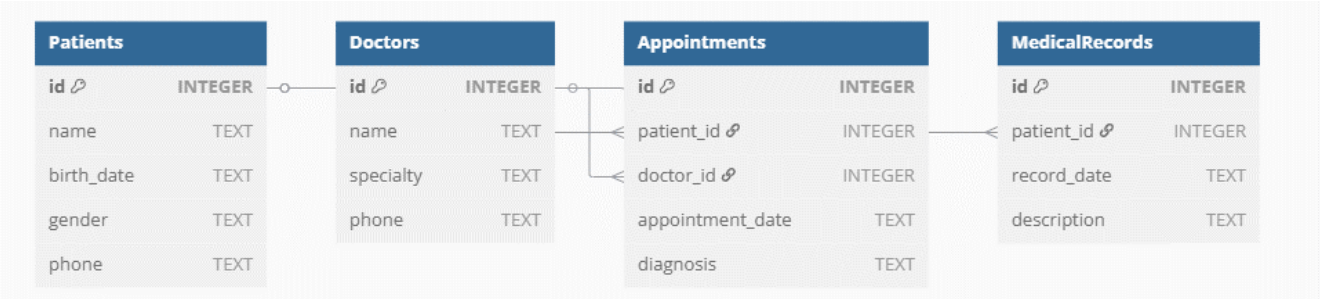


Рисунок 3.7 – Візуальна діаграма зв’язків між сутностями бази

3.3 Наповнення бази даних тестовими даними та приклади запитів

Після створення структури бази даних у попередньому підрозділі, наступним етапом є її наповнення тестовими даними та перевірка працездатності шляхом виконання типових SQL-запитів.

Вставка тестових даних. Перш за все, заповнимо таблиці Patients, Doctors, Appointments та MedicalRecords демонстраційними записами. Для цього скористаємося SQL-запитами типу INSERT INTO.

Заповнення таблиці «patients» (рис. 3.8). Вибрано три пацієнти з типово поширеними іменами, щоб відобразити реальний розподіл статі та віку:

– Іван Петренко – чоловік 1990 року народження, що відповідає середньому дорослому віку;

– Олена Коваль – жінка середнього віку (1985 рік), що дозволяє мати приклад пацієнта з хронічними захворюваннями;

– Марія Шевченко – жінка старшого віку (1978 рік), додає різноманіття у віковому спектрі.

```
INSERT INTO Patients (full_name, birth_date, gender, phone) VALUES
('Іван Петренко', '1990-05-14', 'Ч', '+380501112233'),
('Олена Коваль', '1985-11-02', 'Ж', '+380671234567'),
('Марія Шевченко', '1978-03-21', 'Ж', '+380931234567');
```

Рисунок 3.8 – Заповнення таблиці «patients»

Заповнення таблиці «doctors» (рис. 3.9). Обрано трьох лікарів з різними спеціальностями, які найчастіше зустрічаються у медичних установах: терапевт, кардіолог і педіатр. Це дає можливість показати зв’язки з різними типами прийомів. Номери телефонів також вигадані, для демонстрації коректного формату збереження.

```
INSERT INTO Doctors (full_name, specialty, phone) VALUES
('Андрій Іванов', 'Терапевт', '+380501112244'),
('Світлана Білик', 'Кардіолог', '+380672223344'),
('Олег Степаненко', 'Педіатр', '+380931234888');
```

Рисунок 3.9 – Заповнення таблиці «doctors»

Заповнення таблиці «appointments» (рис. 3.10). Для трьох пацієнтів назначено по одному прийому до різних лікарів з типовими діагнозами:

– ГРВІ (гостре респіраторне вірусне захворювання) – поширене захворювання для терапевта;

– Артеріальна гіпертензія – хронічне серцево-судинне захворювання для кардіолога;

– Плановий огляд – типова процедура для педіатра.

Ці дані демонструють типову роботу медичного закладу.

```
INSERT INTO Appointments (patient_id, doctor_id, appointment_date, reason) VALUES
(1, 1, '2024-05-01', 'ГРВІ'),
(2, 2, '2024-05-02', 'Артеріальна гіпертензія'),
(3, 3, '2024-05-03', 'Плановий огляд');
```

Рисунок 3.10 – Заповнення таблиці «appointments»

Заповнення таблиці «medical_records» (рис. 3.11). Для кожного прийому створено медичний запис із коротким описом лікування чи результатів огляду. Це показує, що база зберігає не лише загальні дані, а й медичні коментарі, які важливі для подальшого аналізу.

```
INSERT INTO medical_records (patient_id, record_date, diagnosis, treatment) VALUES
(1, '2025-05-10', 'Гостре респіраторне вірусне захворювання (ГРВІ)', 'Призначено прот
(2, '2025-05-12', 'Артеріальна гіпертензія', 'Корекція харчування, призначення антиг
(3, '2025-05-15', 'Плановий огляд здоров'я', 'Рекомендовано вітаміни, загальний огляд
```

Рисунок 3.11 – Заповнення таблиці «medical_records»

Таким чином ми наповнили готову базу даних тестовою інформацією. На рисунку 3.12 можна побачити, що у візуальному представленні таблиць усі дані, які ми вносили за допомогою SQL-коду підтягнулися вірно і на свої місця.

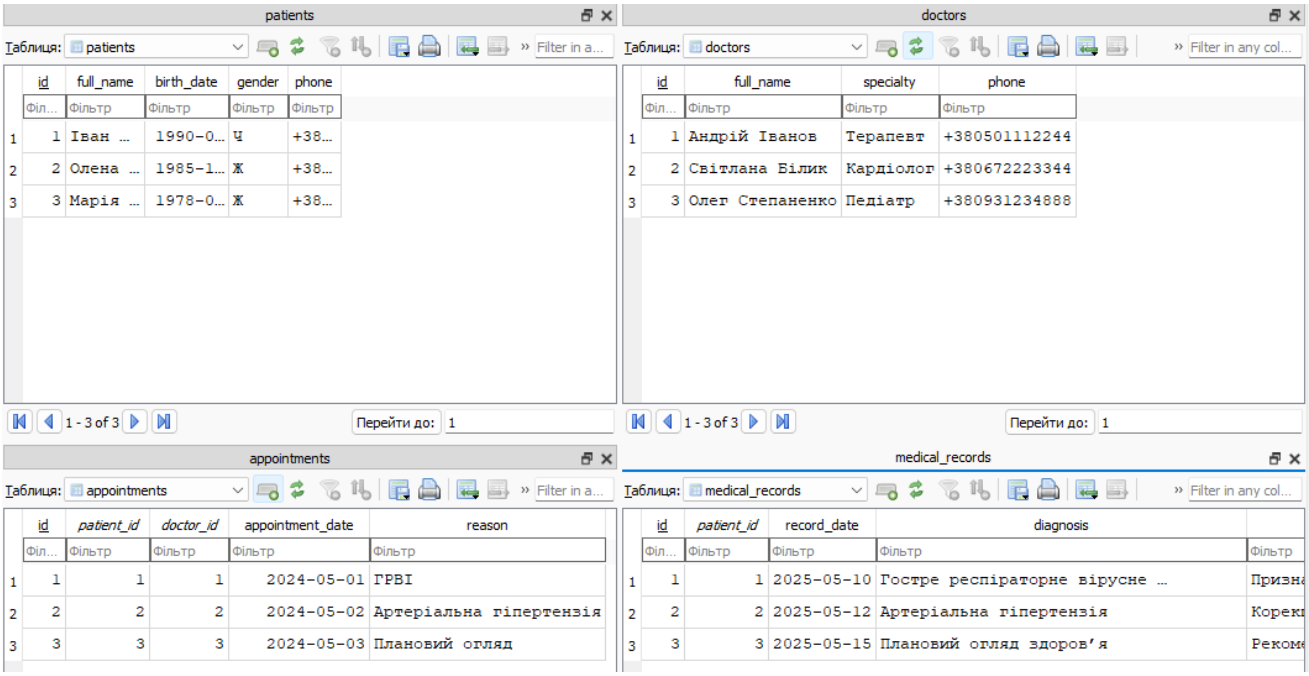


Рисунок 3.12 – Візуалізація заповнених таблиць готової бази даних.

Витягування даних з бази за допомогою SELECT-запитів. Для роботи з даними знову використовуватимемо вкладку Execute SQL. Нижче наведені приклади запитів і пояснення до них.

Перегляд усіх пацієнтів (рис. 3.13). Цей запит повертає повну інформацію про пацієнтів, включно з їх іменами, датами народження, статтю та контактними номерами.

```
1    SELECT * FROM patients;
```

	id	full_name	birth_date	gender	phone
1	1	Іван Петренко	1990-05-14	Ч	+380501112233
2	2	Олена Коваль	1985-11-02	Ж	+380671234567
3	3	Марія Шевченко	1978-03-21	Ж	+380931234567

Рисунок 3.13 – Перегляд усіх пацієнтів

Перегляд лікарів за спеціальністю (рис. 3.14). Запит показує лікарів кардіологів, з контактною інформацією, що допомагає швидко знайти потрібного спеціаліста.

```
1  SELECT full_name, specialty, phone FROM doctors WHERE specialty = 'Кардіолог';
2
```

	full_name	specialty	phone
1	Світлана Білик	Кардіолог	+380672223344

Рисунок 3.14 – Перегляд лікарів за спеціальністю

Перегляд прийомів конкретного пацієнта (наприклад, patient_id = 1) (рис. 3.15). Об’єднання таблиць «appointments» і «doctors», щоб побачити дату прийому, причину, ім'я лікаря і його спеціалізацію для обраного пацієнта. Результати відсортовані за датою прийому у спадному порядку.

```
1  SELECT a.appointment_date, a.reason, d.full_name AS doctor_name, d.specialty
2  FROM appointments a
3  JOIN doctors d ON a.doctor_id = d.id
4  WHERE a.patient_id = 1
5  ORDER BY a.appointment_date DESC;
6
```

	appointment_date	reason	doctor_name	specialty
1	2024-05-01	ГРВІ	Андрій Іванов	Терапевт

Рисунок 3.15 – Перегляд прийомів конкретного пацієнта

Перегляд медичних записів пацієнта (рис. 3.16): запит показує історію медичних записів пацієнта, включно з датою запису, діагнозом і лікуванням, з відсортованими найновішими записами зверху.

```
1  SELECT record_date, diagnosis, treatment
2  FROM medical_records
3  WHERE patient_id = 1
4  ORDER BY record_date DESC;
5
```

	record_date	diagnosis	treatment
1	2025-05-10	Гостре респіраторне вірусне …	Призначено противірусні препарати, …

Рисунок 3.16 – Перегляд медичних записів пацієнта

3.4 Висновки до практичної частини

У ході виконання практичної частини було успішно реалізовано просту, але повноцінну базу даних для умовної медичної інформаційної системи. Для цього було використано легку та зручну у використанні реляційну СУБД – SQLite, а також графічний інтерфейс DB Browser for SQLite, що значно полегшив процес проєктування, наповнення та візуалізації структури бази даних.

Було створено чотири взаємопов’язані таблиці:

1. «patients». Зберігає інформацію про пацієнтів;
2. «doctors». Містить дані про лікарів і їх спеціалізації;
3. «appointments». Фіксує факти прийомів пацієнтів у лікарів;
4. «medical_records». Медична історія пацієнтів, діагнози та призначене лікування.

Усі таблиці були наповнені тестовими даними за допомогою INSERT-запитів. Було також виконано низку SQL-запитів для демонстрації можливостей витягування, об'єднання та фільтрації даних. Зокрема, були продемонстровані запити на отримання прийомів конкретного пацієнта, перегляд лікарів за спеціальністю та історії медичних записів. Результати були задокументовані скріншотами, що підтверджують коректність реалізації.

Переваги використаного реляційного підходу. Використання реляційної моделі даних дало змогу:

– організувати чітку структуру з логічними зв’язками між таблицями за допомогою зовнішніх ключів;

– забезпечити цілісність даних, зменшивши ймовірність дублювання або неконсистентної інформації;

– ефективно виконувати запити для витягування пов’язаних даних з кількох таблиць;

– масштабувати модель у майбутньому без необхідності її повної перебудови;

– простоту імплементації – SQLite не потребує окремої інсталяції сервера, що ідеально підходить для прототипування та невеликих проєктів.

Крім того, SQLite – це файлова база, яка ідеально підходить для вбудованих систем і локальних застосунків, де важливі швидкість, легкість розгортання і мінімальні вимоги до ресурсів.

Можливості розширення моделі. Реалізована база даних має просту, але логічну структуру, яку можна легко адаптувати до більш складних задач. Зокрема, її можна розширити наступним чином:

– додати таблицю «Користувачі системи» для розмежування прав доступу (адміністратор, лікар, реєстратор);

– впровадити історію змін записів для забезпечення відстеження редагувань у медичних даних;

– створити таблиці для призначених медикаментів, лабораторних аналізів, скарг пацієнтів або щеплень;

– зв'язати базу з інтерфейсом користувача (наприклад, через Python або інші мови), що дозволить створити повноцінну систему управління медичною інформацією.

Такі очевидні можливості розширення дають підстави стверджувати, що навіть проста реалізація на SQLite може стати основою для функціональної та масштабованої медичної інформаційної системи.

ВИСНОВКИ

У межах кваліфікаційної роботи було досліджено особливості побудови та використання баз даних у медичних інформаційних системах, а також реалізовано власну спрощену модель такої системи на базі реляційної СУБД SQLite. Актуальність обраної теми зумовлена зростаючими обсягами медичної інформації та потребою у її структурованому, надійному й безпечному зберіганні та обробці.

У теоретичній частині роботи було проаналізовано основні вимоги до медичних інформаційних систем, розглянуто переваги та недоліки використання реляційних, нереляційних та хмарних баз даних для зберігання клінічної інформації, а також типові моделі даних, які використовуються в медичних закладах. Окрема увага була приділена поняттям масштабованості, безпеки даних і забезпечення конфіденційності інформації.

Практична частина кваліфікаційної роботи передбачала створення бази даних із чотирма таблицями: пацієнти, лікарі, прийоми та медичні записи. Була реалізована логічна модель з відповідними зовнішніми ключами, що забезпечує зв’язок між сутностями. Для реалізації використано СУБД SQLite та графічний інтерфейс DB Browser for SQLite, що дало змогу ефективно візуалізувати структуру даних, наповнити таблиці тестовими записами та виконати запити для демонстрації витягування й аналізу інформації.

Отримані результати підтверджують доцільність застосування реляційної моделі у сфері охорони здоров’я, особливо на рівні локальних інформаційних систем. Побудована база даних має просту та гнучку структуру, яку можна масштабувати під більші проєкти. Вона може слугувати основою для подальшого розвитку повноцінного застосунку, інтегрованого з електронною медичною картою, системою запису на прийом, аналітикою та звітністю.

У підсумку, поставлені в роботі цілі були досягнуті, а її результати можуть бути використані для розробки навчальних, прототипних або реальних медичних інформаційних систем з обмеженим обсягом даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Malachynska M. J. Медичні інформаційні системи. Medical informatics and engineering. 2015. № 4. URL: https://doi.org/10.11603/mie.1996-1960.2015.4.5472 (дата звернення: 23.03.2025).

2) Kizian M., Opara A., Bovsunovskaya K. Medical information systems in the outpatient: improving the provision of medical aid. International scientific journal "Internauka". 2023. No. 9(143). URL: https://doi.org/10.25313/2520-2057-2023-9-8914 (дата звернення: 23.03.2025).

3) Stout Q. F., Woodworth P. A. Relational databases. The american mathematical monthly. 1983. Vol. 90, no. 2. P. 101. URL: https://doi.org/10.2307/2975809 (дата звернення: 24.03.2025).

4) eHealth 2008 (2008 London, England). Electronic healthcare: first international conference, ehealth 2008, london, UK, september 8-9, 2008 : revised selected papers. Berlin : Springer, 2009. 222 p.

5) The security rule. HHS.gov. URL: https://www.hhs.gov/hipaa/for-professionals/security/index.html (дата звернення: 24.03.2025).

6) Next-Gen Hospital Information Systems : White Paper / *IEEE Ukraine Section*. - 2022. - 45 p. URL: https://ieee.org.ua/whitepapers

7) Healthcare analytics reference architecture. URL: https://docs.aws.amazon.com/wellarchitected/latest/healthcare-industry-lens/healthcare-analytics-reference-architecture.html (дата звернення: 25.03.2025).

8) Розбудова електронної системи охорони здоров'я в Україні. Міністерство охорони здоров'я України. URL: https://moz.gov.ua/uk/rozbudova-esoz-yiyi-metodologichna-ta-tehnichna-arhitektura (дата звернення: 26.03.2025).

9) Електронна система охорони здоров'я. Міністерство охорони здоров'я України. URL: https://ehealth.gov.ua/ (дата звернення: 26.03.2025).

10) Nisa B. U. A Comparison between Relational Databases and NoSQL Databases. International Journal of Trend in Scientific Research and Development. 2018.

Volume-2, Issue-3. P. 845–848. URL: https://doi.org/10.31142/ijtsrd11214 (дата звернення: 28.03.2025).

11) EHealth – що це та як підключитися?. URL: https://blog.h24.ua/uk/ehealth-shho-tse-ta-yak-pidklyuchytysya/ (дата звернення: 28.03.2025).

12) НСЗУ Національна служба здоров'я України. URL: https://www.facebook.com/nszu.ukr/posts/ehealthелектронна-медична-картка-емк-це-сукупність-персональних-та-медичних-дани/595638516086764/ (дата звернення: 28.03.2025).

13) Огляд моделей розвитку eHealth та наявних медичних інформаційних систем. Науковий вісник НЛТУ України. URL: https://nv.nltu.edu.ua/Archive/2017/27_10/29.pdf (дата звернення: 29.03.2025).

14) Li G., Dong H., Zhang C. Cloud databases. Proceedings of the VLDB Endowment. 2022. Vol. 15, no. 12. P. 3758–3761. URL: https://doi.org/10.14778/3554821.3554893 (дата звернення: 29.03.2025).

15) Регламент (ЄС) 2016/679 Європейського Парламенту і Ради від 27 квітня 2016 р. про захист фізичних осіб у зв'язку з обробкою персональних даних і про вільне переміщення таких даних (Загальний регламент про захист даних, GDPR). Офіційний вісник ЄС. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679 (дата звернення: 29.03.2025).

16) Health Insurance Portability and Accountability Act of 1996 (HIPAA). U.S. Government Publishing Office. URL: https://www.govinfo.gov/content/pkg/PLAW-104publ191/pdf/PLAW-104publ191.pdf (дата звернення: 30.03.2025).

17) Про захист персональних даних : Закон України від 01.06.2010 №2297-VI. URL: https://zakon.rada.gov.ua/laws/show/2297-17#Text (дата звернення: 30.03.2025).

18) Conceptual eHealth Digital System of Ukraine Architecture. ResearchGate. URL: https://www.researchgate.net/figure/Conceptual-eHealth-Digital-System-of-Ukraine-Architecture_fig1_376468160 (дата звернення: 01.04.2025).

19) HIPAA Compliance Infographic. Compliancy Group. URL:

https://compliancy-group.com/hipaa-compliance-infographic/ (дата звернення: 01.04.2025).

20) Laybats C., Davies J. GDPR. Business Information Review. 2018. Vol. 35, no. 2. P. 81–83. URL: https://doi.org/10.1177/0266382118777808 (дата звернення: 01.04.2025).

21) Мельник К. В. Моделювання процесу інтелектуальної обробки медичних даних. Системи обробки інформації. 2017. № 4(150). С. 237–244. URL: https://doi.org/10.30748/soi.2017.150.48 (дата звернення: 01.04.2025).

22) Implementing HL7 FHIR / C. Jaffe et al. Journal of Healthcare Management Standards. 2023. Vol. 2, no. 1. P. 1–8. URL: https://doi.org/10.4018/jhms.316144 (дата звернення: 02.04.2025).

23) URM Consulting. GDPR Principles. URL: https://www.urmconsulting.com/blog/gdpr-back-to-basics (date of access: 22.05.2025).

24) Ліпкан В. А. Національна безпека України: нормативно-правові аспекти забезпечення : Монографія. Київ : Текст, 2003. 180 с.

25) Теремецький В. Загальна характеристика охорони бази персональних даних за законодавством України. Підприємництво, господарство і право. 2015. № 10 (238). С. 3–5.

26) Healthcare Data Breach Statistics. HIPAA Journal. URL: https://www.hipaajournal.com/healthcare-data-breach-statistics/ (дата звернення: 03.04.2025).

27) Hipp D. R. SQLite Home Page. URL: https://www.sqlite.org/index.html (дата звернення: 04.04.2025).

28) Newman C. SQLite. Upper Saddle River : Pearson Education, 2004.

29) Singh S. Security Analysis of MongoDB. International Journal for Digital Society. 2019. Vol. 10, no. 4. P. 1556–1561. URL: https://doi.org/10.20533/ijds.2040.2570.2019.0193 (дата звернення: 05.04.2025).

30) DB Browser for SQLite. URL: https://sqlitebrowser.org/about/ (дата звернення: 06.04.2025).

Додаток А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ



Навчально-науковий інститут Інформаційних технологій
Кафедра Комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА
на тему: «Бази даних в медичних інформаційних системах»

Виконав студент групи КІД-41
Вашкевич Іван Євгенович

Керівник кваліфікаційної роботи:
Торошанко Ярослав Іванович, доцент кафедри КІ, к.т.н., с.н.с.

Київ – 2025

Мета роботи – дослідження теоретичних засад та практична реалізація медичної бази даних.

Об`єкт дослідження – інформаційні системи, призначені для обробки медичних даних.

Предмет дослідження – база даних як ключовий компонент медичної інформаційної системи.

Актуальність обраної теми

Актуальність обраної теми обумовлена зростаючими обсягами медичної інформації та потребою у її структурованому, надійному й безпечному зберіганні та обробці в контексті розвитку електронних систем охорони здоров'я в Україні, зокрема державної системи eHealth, яка покликана реформувати галузь шляхом впровадження єдиного інформаційного простору.

Поняття та класифікація медичних інформаційних систем

Електронна системи охорони здоров'я (ЕСОЗ) та її складові





Схема МІС

Типи баз даних

Реляційна БД

student

id	name	surname	age
1	John	Brown	19
2	Emma	Carly	23

Нереляційна БД

Файл student.json:

```
[
  {
    "id": 1,
    "name": "John",
    "surname": "Brown",
    "age": 19,
  },
  {
    "id": 2,
    "name": "Emma",
    "surname": "Carly",
    "age": 23,
  }
]
```

Рисунок 1.2 - Різниця між реляційними і нереляційними БД



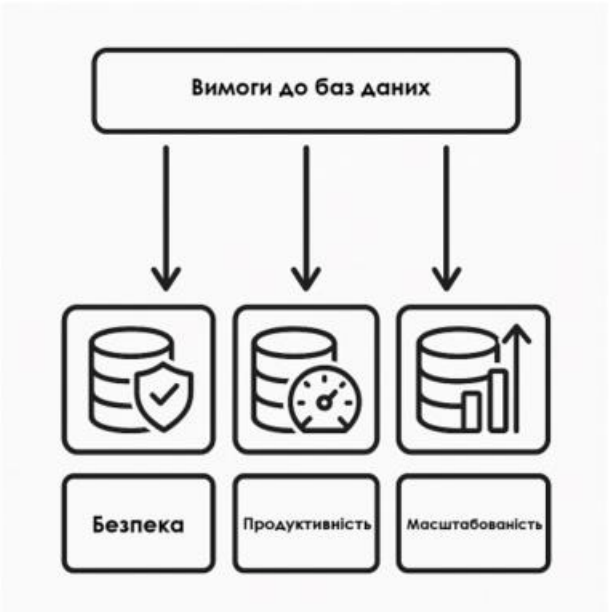
Рисунок 1.1 - Реляційна база даних



Рисунок 1.3 - Хмарні бази даних

Вимоги до баз даних у медицині

- **Безпека медичних даних** - формує основу довіри до електронних систем охорони здоров'я.
- **Продуктивність системи** - визначає якість медичного обслуговування.
- **Масштабованість системи** - відображає здатність системи адаптуватися до зростання обсягів даних та кількості користувачів без втрати продуктивності.



Нормативно-правове регулювання захисту медичних даних

- **Національне законодавство України у сфері захисту медичних даних.** В Україні ключовим нормативним актом є Закон «Про захист персональних даних», який запроваджує базові принципи обробки даних.
- **GDPR (General Data Protection Regulation).** Європейська модель захисту персональних даних.
- **HIPAA (Health Insurance Portability and Accountability Act).** Стандарти безпеки медичних даних, ухвалених у США.







Оцінка продуктивності, масштабованості та безпеки різних типів баз даних

Таблиця 2.1 - Порівняння баз даних

Критерій	Реляційні БД (SQL)	Нереляційні БД (NoSQL)	Хмарні БД
Продуктивність	Висока для структурованих даних і складних запитів; оптимізовані для транзакцій	Висока при роботі з великими обсягами неструктурованих даних	Залежить від провайдера; зазвичай висока завдяки автоматичному масштабуванню
Масштабованість	Вертикальна; масштабування обмежене апаратними ресурсами	Горизонтальна; добре масштабуються за рахунок розподіленої архітектури	Автоматична горизонтальна масштабованість, гнучка конфігурація
Безпека	Розвинені механізми контролю доступу, транзакційність, шифрування	Можуть мати обмежену підтримку складної безпеки, залежить від реалізації	Зазвичай включають вбудовані засоби шифрування, аудит, але залежать від постачальника

Проєктування та вибір інструментів

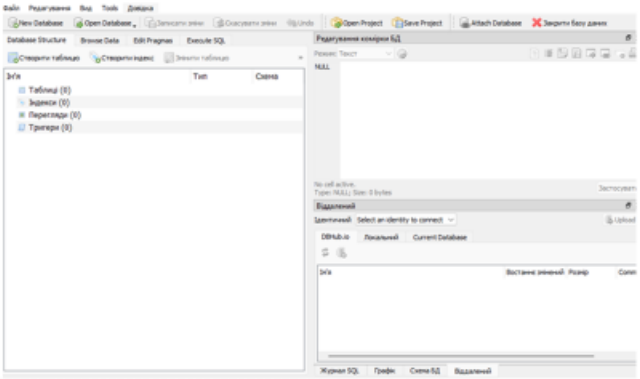
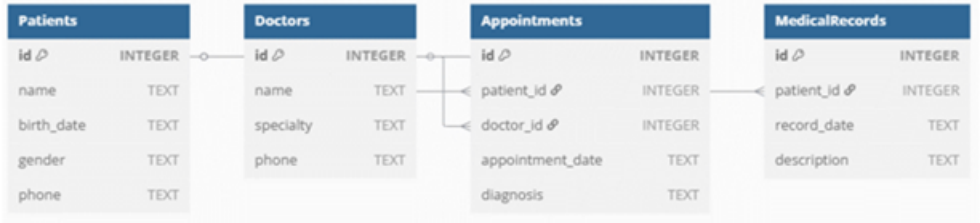


Рисунок 3.1 - Інтерфейс SQLite

Рисунок 3.7 - Візуальна діаграма зв'язків між сутностями бази



Створення бази даних і наповнення тестовими даними

```sql
CREATE TABLE patients (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    full_name TEXT NOT NULL,
    birth_date DATE,
    gender TEXT,
    phone TEXT
);
```

Рисунок 3.2 - Створення таблиці «patients»

```sql
CREATE TABLE doctors (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    full_name TEXT NOT NULL,
    specialty TEXT,
    phone TEXT
);
```

Рисунок 3.3 - Створення таблиці «doctors»

```sql
CREATE TABLE appointments (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    patient_id INTEGER,
    doctor_id INTEGER,
    appointment_date DATE,
    reason TEXT,
    FOREIGN KEY (patient_id) REFERENCES patients(id),
    FOREIGN KEY (doctor_id) REFERENCES doctors(id)
);
```

Рисунок 3.4 - Створення таблиці «appointments»

```sql
CREATE TABLE medical_records (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    patient_id INTEGER,
    record_date DATE,
    diagnosis TEXT,
    treatment TEXT,
    FOREIGN KEY (patient_id) REFERENCES patients(id)
);
```

Рисунок 3.5 - Створення таблиці «medical_records»

```sql
INSERT INTO Patients (full_name, birth_date, gender, phone) VALUES
('Іван Петренко', '1990-05-14', 'Ч', '+380501112233'),
('Олена Коваль', '1985-11-02', 'Ж', '+380671234567'),
('Марія Шевченко', '1978-03-21', 'Ж', '+380931234567');
```

Рисунок 3.8 – Заповнення таблиці «patients»

```sql
INSERT INTO Doctors (full_name, specialty, phone) VALUES
('Андрій Іванов', 'Терапевт', '+380501112244'),
('Світлана Білик', 'Кардіолог', '+380672223344'),
('Олег Степаненко', 'Педіатр', '+380931234888');
```

Рисунок 3.9 – Заповнення таблиці «doctors»

```sql
INSERT INTO Appointments (patient_id, doctor_id, appointment_date, reason) VALUES
(1, 1, '2024-05-01', 'ГРВІ'),
(2, 2, '2024-05-02', 'Артеріальна гіпертензія'),
(3, 3, '2024-05-03', 'Планový огляд');
```

Рисунок 3.10 – Заповнення таблиці «appointments»

```sql
INSERT INTO medical_records (patient_id, record_date, diagnosis, treatment) VALUES
(1, '2025-05-10', 'Гостре респіраторне вірусне захворювання (ГРВІ)', 'Призначено прот
(2, '2025-05-12', 'Артеріальна гіпертензія', 'Корекція харчування, призначення антигі
(3, '2025-05-15', 'Плановий огляд здоров'я', 'Рекомендовано вітаміни, загальний огляд
```

Рисунок 3.11 – Заповнення таблиці «medical_records»

Приклади витягування даних з бази

```
SELECT a.appointment_date, a.reason, d.full_name AS doctor_name, d.specialty
FROM appointments a
JOIN doctors d ON a.doctor_id = d.id
WHERE a.patient_id = 1
ORDER BY a.appointment_date DESC;
```

appointment_date	reason	doctor_name	specialty
2024-05-01	ГРВІ	Андрій Іванов	Терапевт

Рисунок 3.15 - Перегляд прийомів конкретного пацієнта

```
SELECT record_date, diagnosis, treatment
FROM medical_records
WHERE patient_id = 1
ORDER BY record_date DESC;
```

record_date	diagnosis	treatment
2025-05-10	Гостре респіраторне вірусне …	Призначено противірусні препарати, …

Рисунок 3.16 - Перегляд медичних записів пацієнта

```
SELECT full_name, specialty, phone FROM doctors WHERE specialty = 'Кардіолог';
```

full_name	specialty	phone
Світлана Білик	Кардіолог	+380672223344

Рисунок 3.14 - Перегляд лікарів за спеціальністю

```
SELECT * FROM patients;
```

id	full_name	birth_date	gender	phone
1	Іван Петренко	1990-05-14	Ч	+380501112233
2	Олена Коваль	1985-11-02	Ж	+380671234567
3	Марія Шевченко	1978-03-21	Ж	+380931234567

Рисунок 3.13 - Перегляд інформації про усіх пацієнтів

ВИСНОВКИ

У межах дипломної роботи було досліджено особливості побудови та використання баз даних у медичних інформаційних системах, а також реалізовано власну спрощену модель такої системи на базі реляційної СУБД SQLite. Актуальність обраної теми зумовлена зростаючими обсягами медичної інформації та потребою у її структурованому, надійному й безпечному зберіганні та обробці.

У теоретичній частині роботи було проаналізовано основні вимоги до медичних інформаційних систем, розглянуто переваги та недоліки використання реляційних, нереляційних та хмарних баз даних для зберігання клінічної інформації, а також типові моделі даних, які використовуються в медичних закладах. Окрема увага була приділена поняттям масштабованості, безпеки даних і забезпечення конфіденційності інформації.

ВИСНОВКИ

Практична частина дипломної роботи передбачала створення бази даних із чотирма таблицями: пацієнти, лікарі, прийоми та медичні записи. Була реалізована логічна модель з відповідними зовнішніми ключами, що забезпечує зв'язок між сутностями. Для реалізації використано СУБД SQLite та графічний інтерфейс DB Browser for SQLite, що дало змогу ефективно візуалізувати структуру даних, наповнити таблиці тестовими записами та виконати запити для демонстрації витягування й аналізу інформації.

Отримані результати підтверджують доцільність застосування реляційної моделі у сфері охорони здоров'я, особливо на рівні локальних інформаційних систем. Побудована база даних має просту та гнучку структуру, яку можна масштабувати під більші проєкти. Вона може слугувати основою для подальшого розвитку повноцінного застосунку, інтегрованого з електронною медичною картою, системою запису на прийом, аналітикою та звітністю.

У підсумку, поставлені в роботі цілі були досягнуті, а її результати можуть бути використані для розробки навчальних, прототипних або реальних медичних інформаційних систем з обмеженим обсягом даних.