# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ

**УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА** **ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ**

**Кафедра цифрової економіки та бізнес-** **аналітики**

# КУРСОВА РОБОТА

**з навчальної дисципліни “Проектування та адміністрування БД і СД”**

**Тема:**

«Інформаційна система купівлі-продажу, оренди приміщень.»

# Науковий керівник: Виконавець:

к.ф.-м.н., доц. Депутат Б.Я. Балух Б.І.

(прізвище, ім’я, по-батькові) (прізвище, ім’я, по-батькові)

\_ (підпис) УФЕ-31с група

\_ \_ (підпис)

# “ \_” \_ \_ 2022 р. “ ” \_ 2022 р.

**Загальна кількість балів \_\_** **\_**

(підписи, ПІП членів комісії)

# Львів 2022

[ВСТУП 3](#_bookmark0)

[РОЗДІЛ 1. АНАЛІЗ ВИМОГ 5](#_bookmark1)

* 1. [Постановка завдання 5](#_bookmark2)
  2. [Use Case Diagram 8](#_bookmark4)
  3. [Бізнес модель Canvas 10](#_bookmark5)

[РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ 13](#_bookmark6)

[2.1 Опис моделі даних 13](#_bookmark7)

2.2 Нормалізація відношень…………………………………………………...14

* 1. [Визначення типів даних 18](#_bookmark8)
  2. [Обмеження цілісності даних 20](#_bookmark9)
  3. [Реалізація SQL-скрипту 22](#_bookmark10)

[ВИСНОВКИ 28](#_bookmark11)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 29](#_bookmark12)

[ДОДАТКИ 31](#_bookmark13)

**Вступ**

**Актуальність теми дослідження.** Купівля-продаж чи оренда приміщень було завжди актуальним питанням як зараз так і 100 років тому. В данний момент часу питання де можна орендувати або викупити якесь приміщення звучить часто в звязку з війною.

Чи актуальний в даний момент бізнес по продажу або оренді приміщень , певно актуальний як ніщо інше. Вибір сайтів де можна придбати або орендувати будівлю не великий , а якщо шукати зручний з великою кількістю фільтрів для будь-яких забаганок покупця , то такі можна перерахувати пальцями однієї руки. Досить багато компаній які займаються будівлями працює офлайн і я вважаю що це велике упущення що вони навіть не пробують створити свій сайт щоб люди могли зручно і прямо з дому подивится на можливі варіанти, підібрати собі по забаганкам те чого їм хочеться.

Саме маркетингова частинка такого бізнесу дуже страдає. Враховуючи що ми живемо в 21 столітті власний веб-сайт для такого роду бізнесу є життєво необхідним як мінімум з ціллю популяризацію свого товару і своєї торгової марки для подальшого розвитку іміджу компанії

**Мета і завдання дослідження.**  Метою курсової роботи є дослідження та аналіз актуальності інформаційної системи для купівлі-продажу та оренди нерухомості в Україні

Для досягнення мети в роботі поставленно й вирішено такі теоретичні та практичні завдання:

* Визначити особливості створення веб сайту для Купілві/продажу приміщень в Україні
* Розробити базу данних для сайту
* Перевірити базу данних на наявність дефектів та можливостей покращення

***Обєктом дослідження*** *виступає Офлайн бізнес для якого створюється база данних та її забезпечення та оновлення в подальшому*

***Предметом є*** *застосування різноманітних технолоній та теоритичних аспектів для створення бази даних*

**Практичне значення отриманих результатів.**  Програма використовутиметься особою яка хоче розмітити інформацію про свій бізнес в мережі інтернет з метою популяризації своїх послуг

**Використане програмне забезпечення.** Під час розробки використовувалось середовище розробки Visual Studio Code, платформи Node.js, мова розмітки HTML, CSS, та мова програмування JavaScript. Для проектування бази даних було використано Microsoft SQL Server Management Studio.

**Структура роботи.** Курсова робота складається з вступу, трьох розділів, а саме «Аналіз вимог», «Розробка бази даних» та «Розробка веб-додатку», а також висновків, списку використаних джерел та додатів. Загальний обсяг роботи – 00 сторінок.

**Розділ 1 Аналіз вимог**

* 1. **Постановка завдання**

Основним завданням даної роботи полягає в тому, щоб зробити аналіз актуальності створення інформаційної системи для Купівлі\Продажу\Оренди приміщень в Україні

Найважливішою складовою управлінської інформації є інформація економічна, її властивості визначають науково-технічну і економічну доцільність використання засобів обчислювальної техніки. Саме інформаційні системи забезпечують збір, зберігання, обробку, пошук, передачу економічної інформації, необхідної в процесі прийняття рішень апаратом управління. Для економічної інформації є характерними такі риси: великі обсяги, повторення циклів її отримання і обробки, різноманіття її джерел і споживачів та значна питома вага логічних операцій при її обробці. Інформаційна система (ІС) є системою інформаційного обслуговування апарату управління, вона формується та функціонує в регламенті, який визначається методами та структурою управлінської діяльності, на конкретному економічному об’єкті, реалізує цілі та завдання, які перед ним поставлені. Призначення ІС полягає в описі економічного об’єкта, його стану, взаємодії, що виражається через економічні показники. Вона покликана своєчасно надавати органам управління необхідну і достатню інформацію для прийняття рішень, якість яких забезпечує високоефективну діяльність об’єкта управління.

Отож, в даній роботі нам слід створити базу даних веб-сайту.Дана база даних має допомогти офлайн бізнесу збілиши свій прибуток зарахунок популяризації своєї платформи в інтернеті . В перспективі підєднавши до цієї бази даних сайт , люди отримають набагато зручнішу платформу ніж ті які є зараз на ринку . Основними інструментами в досягнені цього є розробка бази даних , використання селектів та erd діаграми

**1.2 Use Case Diagram**

Діаграма Прецедентів візуально зображає різноманітні сценарії взаємодії між акторами (користувачами) і прецедентами (випадками використання); описує функціональні аспекти системи (бізнес логіку). Діаграми Прецедентів відіграють важливу роль не тільки у комунікації між збирачами вимог до проекту і потенційними користувачами. Діаграми Прецедентів дописані бізнес логікою і детальними специфікаціями прецедентів, як джерельна інформація, успішно використовують учасники розробки проекту на всіх його фазах (зародження, дизайн, програмування, тестування, документування). Добре продумані і завершені специфікації прецедентів легко перевтілюються у Тестові Випадки.

Елементи Діаграми Прецедентів:

* Актор — користувач.
* Прецедент — випадок використання, дія. Позначається овалом.
* Граничні межі системи — охоплюють усі випадки використання у системі.

Елементи взаємодії Діаграми Прецедентів:

* Використовує — користувач виконує дію.
* Включає — один прецедент використовує іншого.
* Розширює — представлення дочірніх прецедентів.
* Вимагає — наступний прецедент вимагає виконання попереднього.
* Схожий — прецеденти подібні, але описують різну функціональність.
* Рівнозначний — подібна функціональність, але користувач сприймає, як різну.

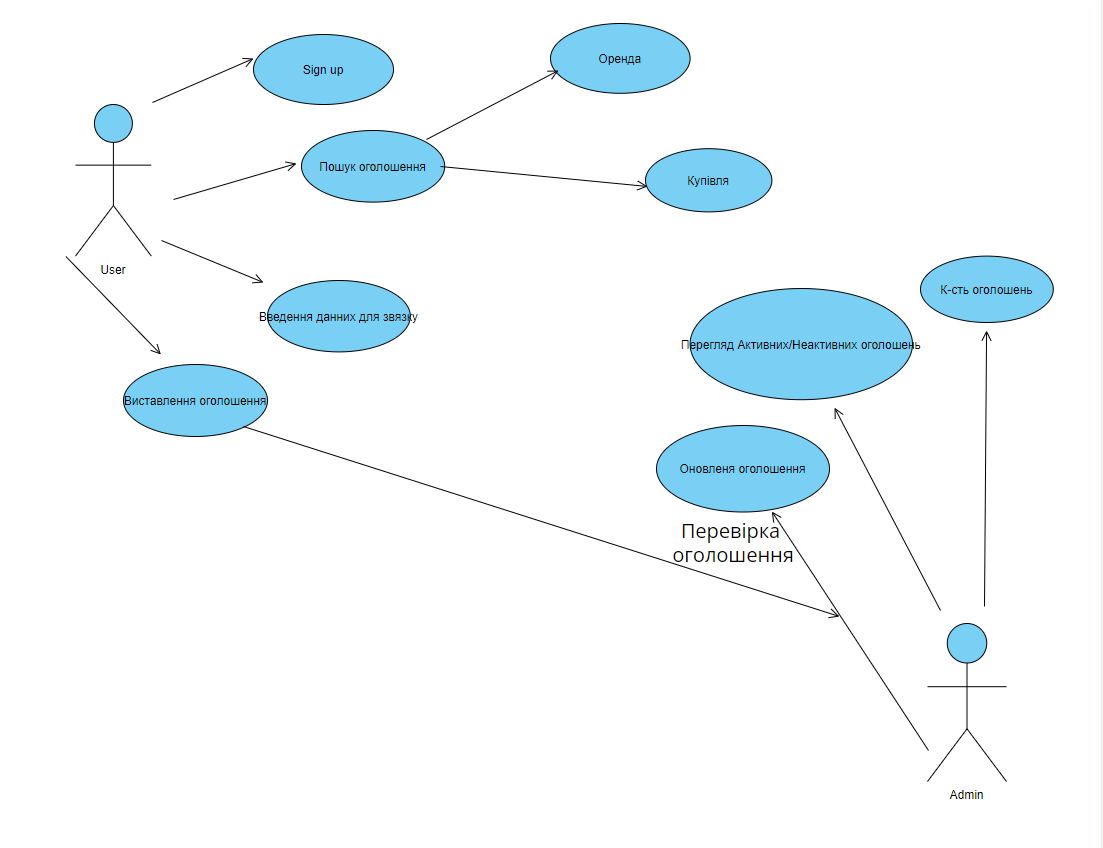


Рис. 1.1 Use case діаграма

В моїй діаграмі є два учасника це юзер та адмін . Для юзера надані такі можливі use кейси як:

1. Sign up
2. Пошук оголошення
3. Виставлення оголошення
4. Введення данних для звязку
5. Можливість вибору між Орендою та купівлею

Адміну надані такі можливі use кейси як :

1. Перевірка оголошення
2. Оновлення оголошення
3. Перегляд активних і неактивних оголошень
4. Та перед к-сть оголошень

**1.3 Бізнес модель Canvas**

Канва бізнес-моделі - один з інструментів стратегічного управління для опису бізнес-моделей нових підприємств, що вже діють. Являє собою односторінкову схему, що описує всі бізнес-процеси компанії: пропозицію, інфраструктуру, споживачів та фінанси Крім того, канва бізнес-моделі може використовуватися організаціями для планування, оцінки або виконання нових моделей взагалі. Таким чином, модель підкреслює ключові моменти і гарантує, що жодні життєво важливі фактори не будуть забуті. Це чудовий інструмент, який допоможе зрозуміти бізнес-модель у простий структурований спосіб. Використання такого полотна дасть змогу дізнатися про клієнтів, яких ви обслуговуєте, які цінні пропозиції через які канали пропонуються та як ваша компанія заробляє гроші. Це так візуальна діаграма з елементами, що описують ціннісні пропозиції фірми або продукту, інфраструктуру, клієнтів та фінанси, допомогаючи підприємствам узгоджувати свою діяльність, ілюструючи потенційні компроміси.

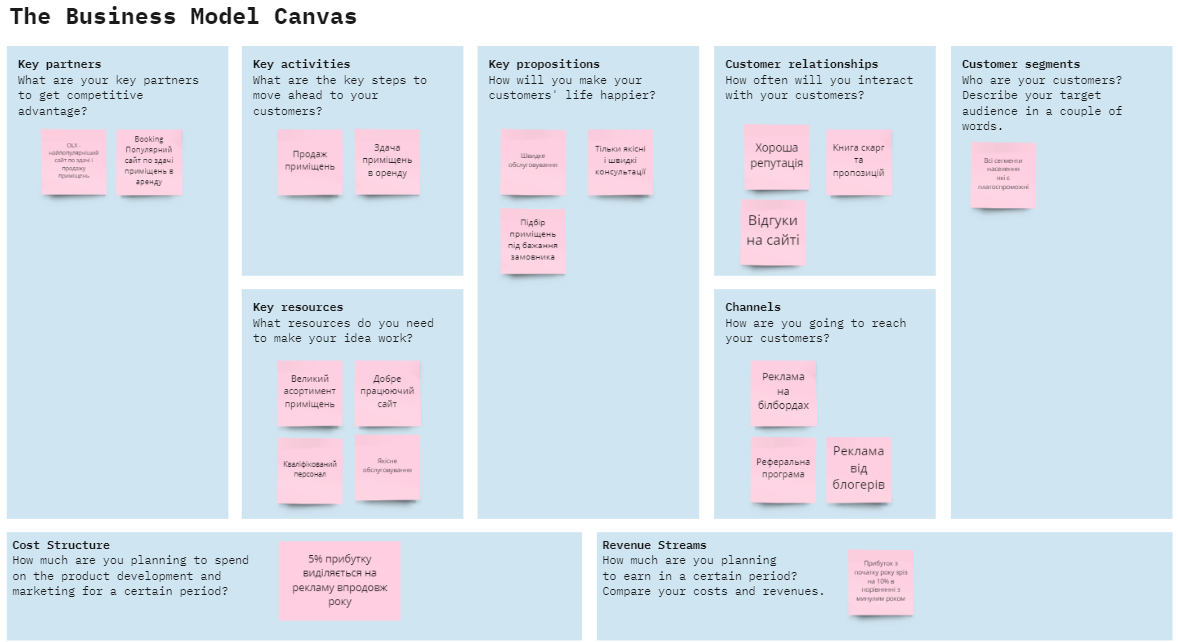


Рис 1.2 Модель Купівлі оренди приміщень

На рис 1.2 видно модель Купівлі оренди приміщень на якій описано всі бізнес-процеси компанії: пропозицію, інфраструктуру, споживачів та фінанси

# 

# РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ

# 2.1 Опис моделі даних

Існують самі різні бази даних, але всі вони створюються на основі більш поширених і фундаментальних моделей. Класифікація інформаційних моделей даних підрозділяється також на безліч різних видів. Наведемо найбільш часто використовувані категорії:

* ієрархічна модель;
* мережна схема;
* реляційна модель;
* об’єктно-орієнтовані схеми.

Всі ці види моделей даних розрізняються між собою за характером подання і зберігання в них інформації.

## **Критерії вибору потрібної моделі**

Користувач може створити базу даних з будь-яким з описаних вище видів. Однак варто зазначити, що вибір моделі даних визначає залежність від деяких факторів.

Самий головний критерій полягає в тому, підтримує використовується клієнтом СУБД конкретну модель. Більшість СУБД побудовані так, що користувачеві представлена модель даних, яка повинна бути використана, однак деякі з них підтримують кілька різних аналогів відразу.

**Ієрархічна модель**

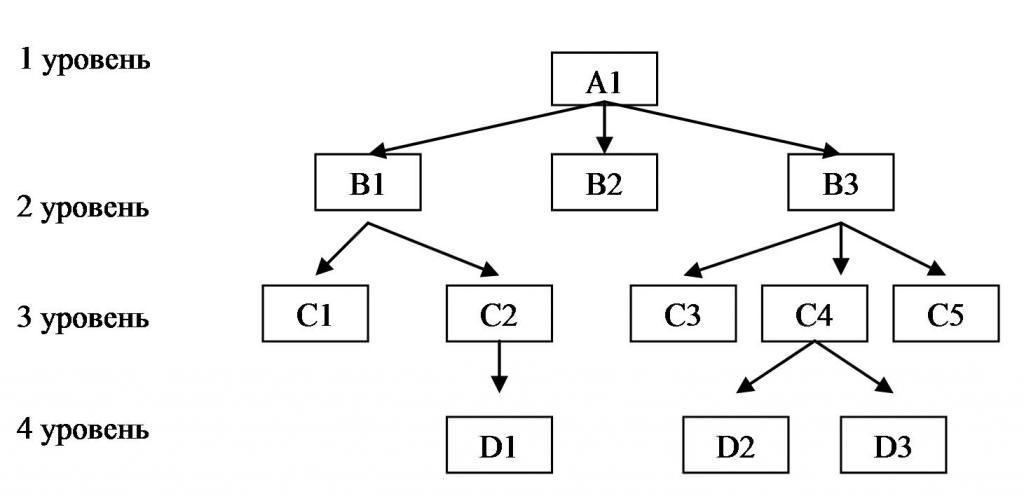


Рис. 2.1 Ієрархічна модель

Є одним з типів моделей представлення даних, організовуючи їх у вигляді сукупності елементів, які розташовані в порядку від загального до приватного.

Структура – перевернуте дерево. Для доступу до одного конкретного файлу є один шлях.

Ієрархічна модель повинна задовольняти трьом основним умовам:

* Кожен вузол нижчого рівня може бути пов’язаний тільки з одним вузлом, що знаходиться на більш високому рівні.
* В ієрархії є всього один головний кореневий вузол, який не підпорядковується жодному іншому вузлу і знаходиться на самому верхньому рівні.
* До будь-якого вузла ієрархії існує тільки один шлях від кореневого вузла.

## **Мережева модель**

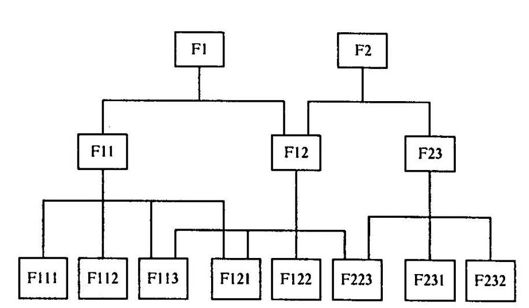


Рис. 2.2 Мережева модель

Вона багато в чому спирається на ієрархічну, маючи з нею досить багато спільного. Головною відмінністю між ними служить тип зв’язку, який передбачає відношення “багато до багатьох”, тобто зв’язки можуть існувати між різними вузлами.

Перевагою мережевої моделі вважається те, що вона витрачає менше ресурсів ПК в плані пам’яті і оперативності, ніж інші моделі.

Недолік такої схеми полягає в тому, що якщо вам знадобиться змінити структуру даних, що зберігаються, то доведеться змінити всі програми, що працюють на основі мережної моделі, так як така структура не є незалежною.

## **Реляційна модель**

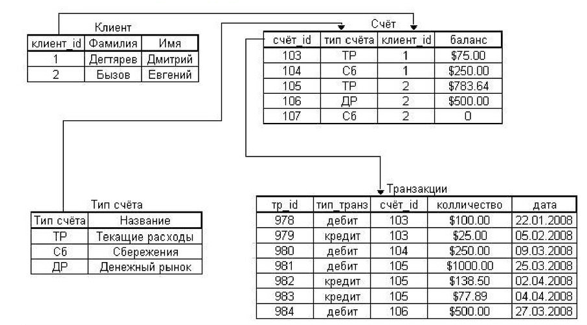


Рис 2.3 Реляційна модель

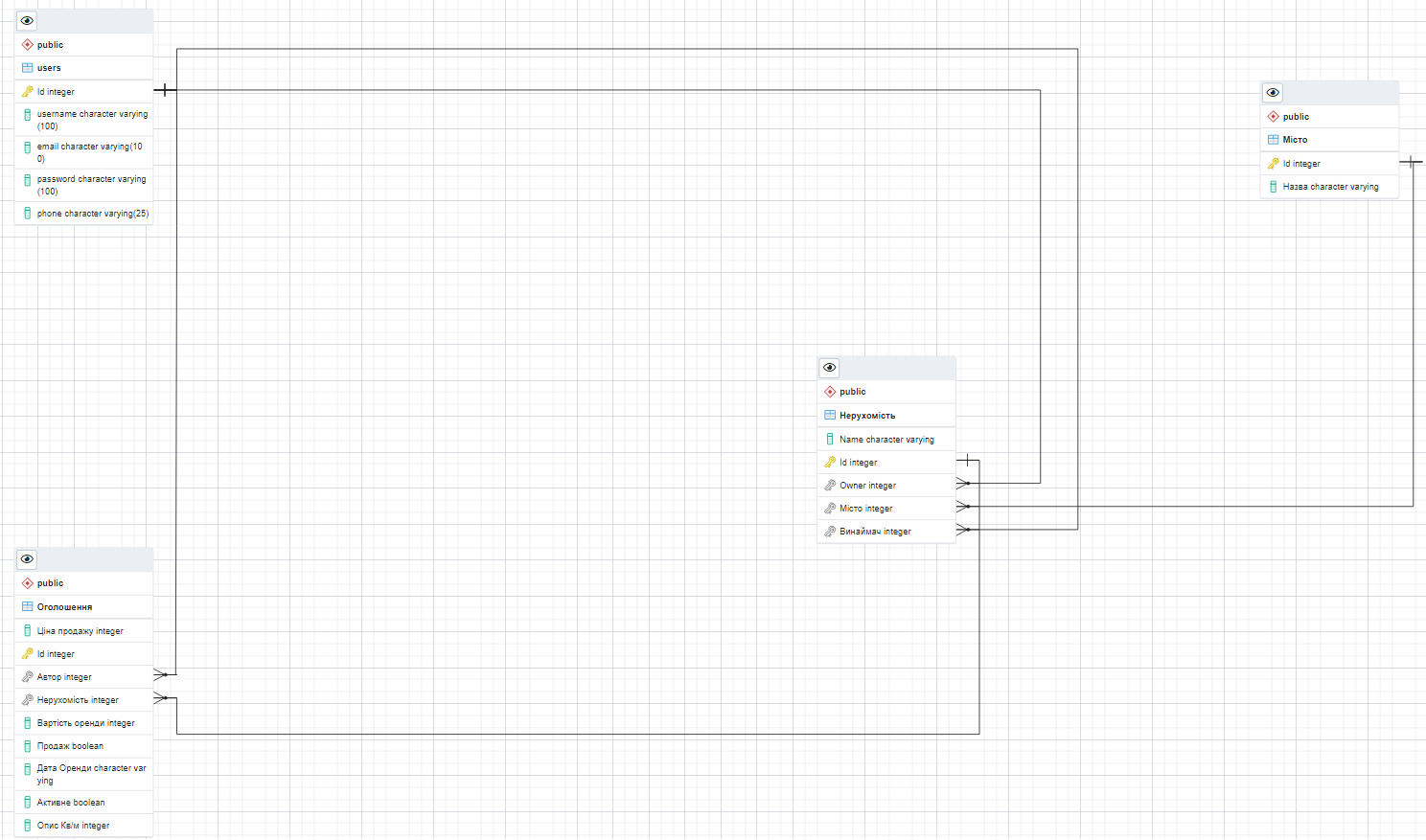
Є найбільш поширеною на сьогоднішній день. Об’єкти і зв’язки між ними при такій моделі даних представлені таблицями, причому зв’язку в них розглядаються як об’єкти. Стовпці в такій таблиці називаються полями, а рядки – записами. Кожна таблиця реляційної моделі повинна задовольняти наступним властивостям:

* Абсолютно всі її стовпці є однорідними, тобто всі елементи, розташовані в одному стовпці, повинні мати однаковий тип і максимально допустимий розмір.
* У кожного стовпця є своє унікальне ім’я.
* Однакових рядків у таблиці бути не повинно.
* Порядок, в якому рядки і стовпці слідують в таблиці, може бути довільним.

Реляційна модель також враховує типи зв’язків між таблицями, включаючи відносини «один до одного», «один до багатьох» і «багато до багатьох».

Бази даних, які створюються на основі табличної реляційної моделі, є гнучкими, адаптуємими і добре масштабованими. Кожний об’єкт даних розбивається на

При розробці моделі використовувалаяся реляційна модель данних



# Рис 2.4 Erd діаграма

# На малюнку 2.4 ми бачимо що дана модель складається з 4 таблиць

# Users таблиця яка містить інформацію про Користувачів сайту,полями таблиці є

# Id – унікальний номер кожного юзера

# Username – нікнейм юзера

# Email – поштова скринька юзера

# Password – пароль від акаунту

# Phone – номер телефону юзера

# Таблиця Нерухомість містить інформацію про власників нерухомості, тип нерухомості, унікальний номер кожної нерухомості, винаймача нерухомості та місце знаходження нерухомості

# Name – тип нерухомості

# Id – унікальний номер нерухомості

# Owner – власник нерухомості

# Місто – місцезнаходження нерухомості

# Винаймач – людина яка взяла в аренду якусь нерухомість

# Таблиця Місто містить інформацію про міста в яких може знаходитися нерухомість , містить наступні поля

# Id – унікальний номер міста

# Назва – Назва міста

# Таблиця Оголошення містить інформацію про ціну продажу якщо вона є , унікальний номер оголошення , автора тобто власника нерухомості , тип нерухомості, вартісь оренди якщо вона присутня , продаж це чи оренда , Дата оренди , чи оголошення активне чи неактивне та опис кв\м, містить такі поля

# Ціна продажу – якщо приміщення продається

# Id – унікальний номер оголошення

# Автор – Власник нерухомості

# Нерухомість – тип нерухомості яка здається чи продається

# Вартість оренди – якщо приміщення здається в оренду

# Продаж

# Дата оренди – якщо приміщення здається в оренду

# Активне – чи активне оголошення чи не активне

# Опис – Кв\м

# 2.2 Нормалізація відношень

Нормалізація відношень- покроковий процес розділення (декомпозиції) початкових відношень БД на простіші. Кроки цього процесу переводять схему відношення БД в послідовні нормальні форми. Кожна наступна форма володіє кращими властивостями ніж попередня. Кожній нормальній формі відповідає певний набір обмежень. При переведенні структури відношення у форми вищого порядку досягають видалення з таблиць надмірної описової інформації.Процес нормалізації заснований на понятті функціональної залежності атрибутів.

Крім завдання більш ефективного використання пам'яті нормалізація дозволяє знизити загрозу порушення цілісності БД через появу в ній внутрішніх суперечностей.

Вводиться поняття нормальних форм (НФ) відносин, кожної з яких відповідає певний набір обмежень. Відношення знаходиться в даній нормальній формі, якщо задовольняє зазначеним обмеженням. Кожна наступна НФ включає в себе вимоги всіх попередніх, тобто є більш суворим обмеженням.

У реляційних баз даних є кілька нормальних форм (НФ):

**Перша нормальна форма** (1НФ). Першим кроком нормалізації є приведення відношення до першої нормальної форми. Відношення в 1НФ повинно відповідати таким вимогам:

* усі атрибути відношення повинні бути унікальними, тобто не допускається їхнього дублювання, а також атомарними, тобто неподільними;
* усі рядки таблиці повинні мати однакову структуру;
* імена стовпців повинні бути різними, а значення однорідними (однакового типу і формату);
* порядок рядків у таблиці не істотний.

Отже, будь-яка нормалізована таблиця буде знаходитись в першій нормальній формі (1НФ).

**Друга нормальна форма** (2НФ). Відношення знаходиться у другій нормальній формі тоді і тільки тоді, коли відношення перебуває в 1НФ і всі його неключові атрибути функціонально повно залежать від первинного ключа. Неключовий атрибут – це атрибут, що не входить до складу первинного ключа.

Перевага 2НФ – це зручність внесення змін у базу даних. 2НФ повністю виключає можливість виникнення протиріччя даних, а також економить пам'ять.

**Третя нормальна форма** (3НФ). Відношення знаходиться у третій нормальній формі тоді і тільки тоді, коли воно перебуває у 2НФ і всі його неключові атрибути взаємно функціонально незалежні. Переваги ЗНФ – це те, що виключається надлишкове дублювання інформації. Тому можна зробити ще такий висновок: відношення знаходиться в ЗНФ, якщо зміна значення будь-якого його атрибута (крім тих, що входять у первинний ключ) не призведе до необхідності зміни значень інших полів.

**Нормальна форма Бойса – Кодда**. Нормальна форма Бойса–Кодда – це підсилена ЗНФ, у якій вивчаються залежності ключових атрибутів від неключових. Відношення перебуває в НФБК, якщо воно перебуває в ЗНФ і в ньому відсутні залежності ключових атрибутів від неключових атрибутів. Але відношення в ЗНФ не завжди можна привести до нормальної форми Бойса–Кодда, не втративши залежності між його атрибутами.

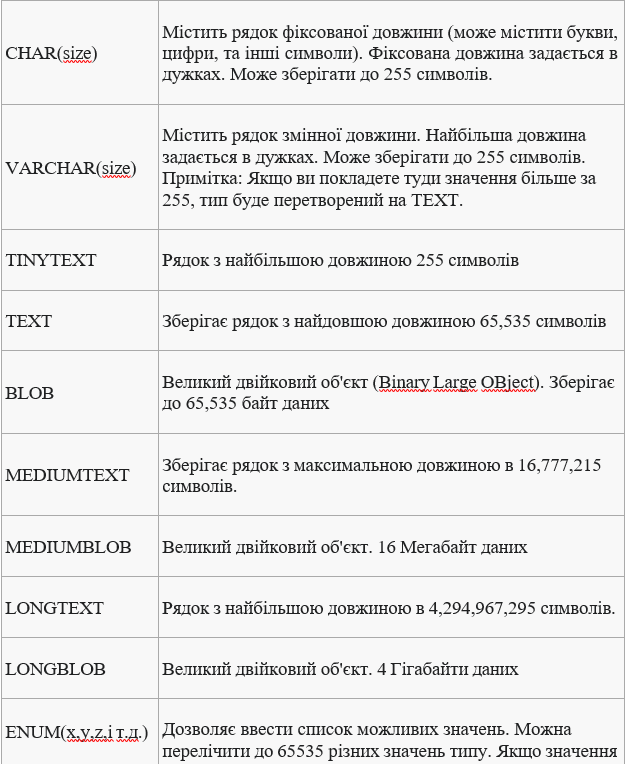
**Четверта нормальна форма** (4НФ). Якщо відношення має багатозначні залежності між атрибутами, то виконують його декомпозицію й отримують 4НФ.

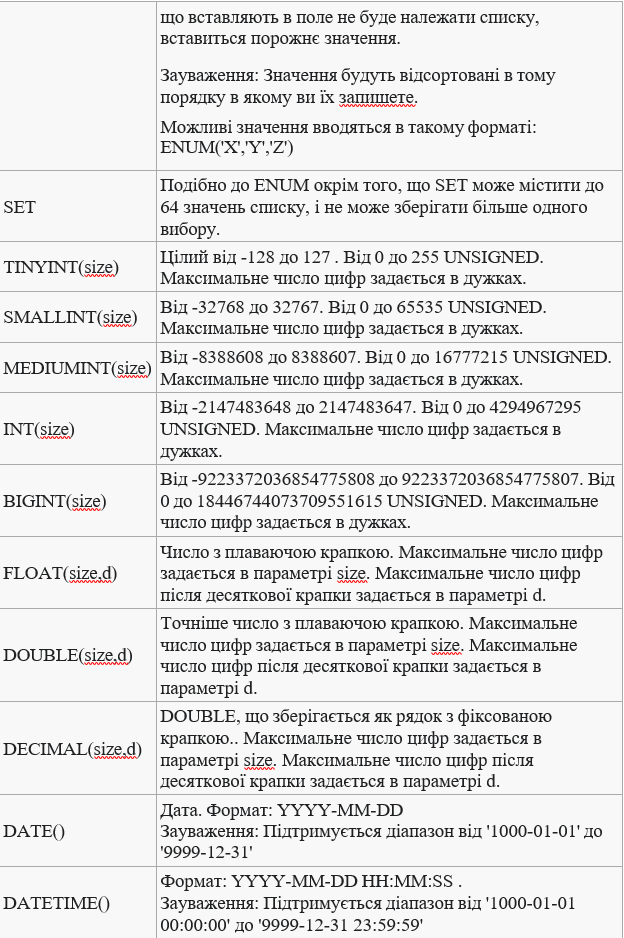
**П'ята нормальна форма** (5НФ). Відношення знаходиться в п'ятій нормальній формі тоді і тільки тоді, коли воно знаходиться в 4НФ і кожна нетривіальна залежність визначається її потенційним ключем.

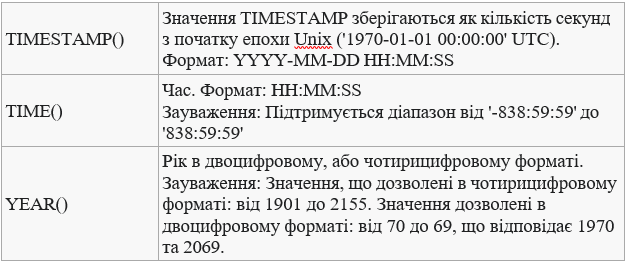
**2.3 Визначення типів даних**

Перед створенням кожної таблиці бази даних необхідно визначити пити даних кожного стовпця. В sql є безліч типів даних.

Нище в таблицях наведено всі типи даних в sql.



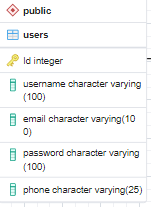




Таблиця 2.5 Типи данних SQL

В SQL Server в кожного стопвця є певний тип даних. Тип даних – це атрибут, який визначає, які дані можуть зберігатися з об’єкті: цілі числа, символи, дата і час і т.д

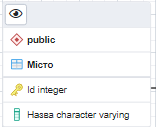
Типи даних які використовувалися в таблицях проекту:



Таблиця 2.6 тип даних таблиці users

В цій таблиці використовувалися такі типи данних як :

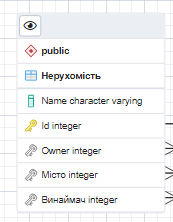
1. Integer
2. Character varying
3. Character varying
4. Character varying
5. Character varying



Таблиця 2.7 тип даних таблиці Місто

В цій таблиці використовувалися такі типи данних як :

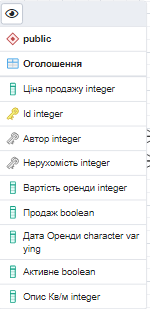
1. Integer
2. Character varying



Таблиця 2.8 тип даних таблиці Місто

В цій таблиці використовувалися такі типи данних як :

1. Character varying
2. Integer
3. Integer
4. Integer
5. Integer



Таблиця 2.9 тип даних таблиці Оголошення

В цій таблиці використовувалися такі типи данних як :

1. Integer
2. Integer
3. Integer
4. Integer
5. Integer
6. Boolean
7. Character varying
8. Boolean
9. Integer

**2.4 Обмеження цілісності даних**

Обмеження цілісності - це правила, які обмежують усі можливі стани бази даних, а також переходи з одного стану в інший. Таким чином, обмеження цілісності визначають множину «допустимих» станів і переходів між ними. База даних перебуває в цілісному стані, якщо вона відповідає всім визначеним для неї вимогам цілісності

Цілісність бази даних– це відповідність інформації, що знаходиться в базі даних, її внутрішній логіці, структурі і всім явно заданим правилам.

Типи правил цілісності:

* CHECK - Контроль допустимих значень атрибутів.
* NOT NULL/NULL - Заборона/ дозвіл на використання не заданих або не визначених значень.
* UNIQUE - Контроль унікальності значень атрибутів.
* PRIMARY KEY - Первинний ключ.
* FOREIGN KEY - Зовнішній ключ.

PRIMARY KEY (PK) використовується для ідентифікації рядків таблиці, в нього є певні особливості:

* В одній таблиці БД може бути тільки один PK.
* Рядки, які мають це правило, не можуть мати не визначені або не задані дані.
* Будучи призначеним для декількох стовпців(складений ключ), це обмеження ставить унікальність комбінацій відповідних значень, хоча окреме значення в кожному стовпці складеного ключа не обов'язково має бути унікальним.

FOREIGN KEY (FK) використовується для того, щоб зв'язать дві таблиці, в нього є такі особливості :

* Поле у головній таблиці, на яке буде адресуваться FG повинне бути оголошене PK або UNIQUE.
* Поле у головній таблиці і поле з якої відбувається адресація повинні мати один й теж самий тип даних

Під час виконання данного завдання було використано декілька (PK/FK)

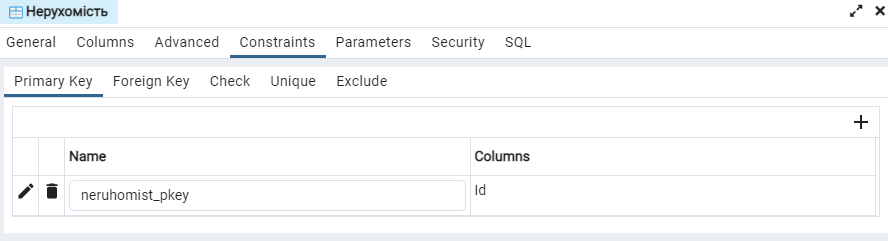
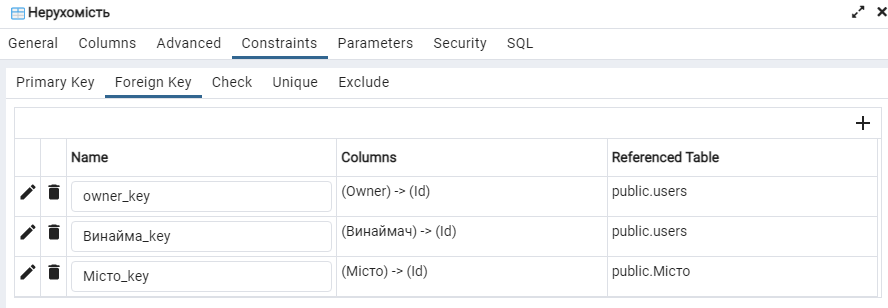
 

Рис. 2.10 Використані вторинні і первинні ключі в табличці “Нерухомість”

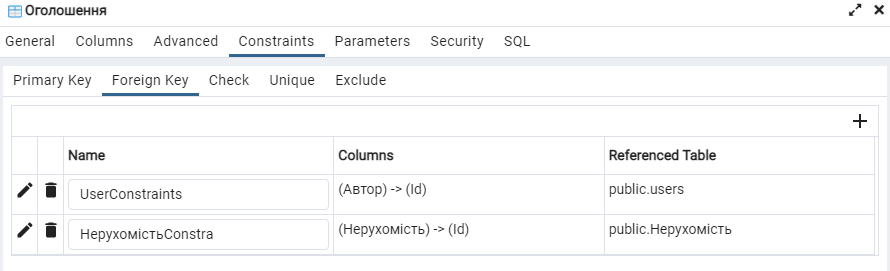
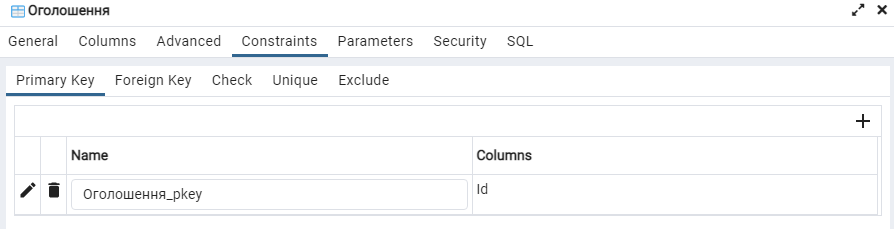


Рис 2.11 Використані первинні і вторинні ключі в таблиці “Оголошення”

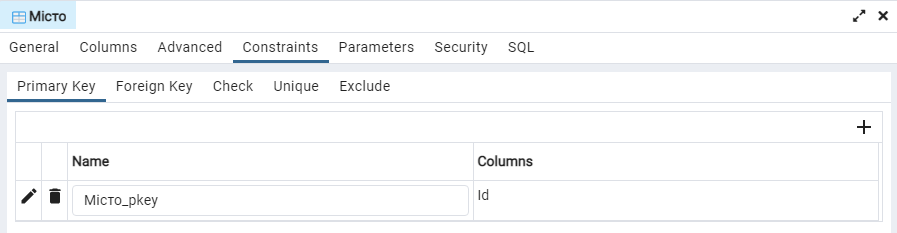


Рис 2.12 Використаний первинний ключ в таблиці “Місто”

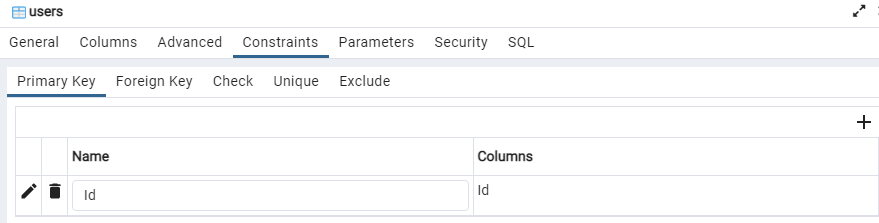


Рис 2.13 Використаний первинний ключ в таблиці “Місто”

# 2.5 Реалізація SQL-скрипту

# Реалізація:

# Запит №1 Вільні приміщення для оренди

# select

# "Нерухомість"."Name",

# "Оголошення"."Вартість оренди",

# "Оголошення"."Дата Оренди",

# "Нерухомість"."Винаймач"

# from "Оголошення"

# inner join "Нерухомість" ON "Нерухомість"."Id" = "Оголошення"."Нерухомість"

# where "Нерухомість"."Винаймач" IS Null and "Оголошення"."Вартість оренди" IS NOT Null;

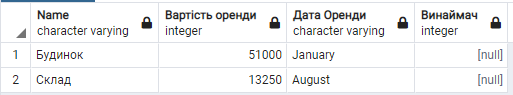


Рис 2.14 Виконання селекту “Вільні приміщення для оренди”

**Запит №2 Активні оголошення продажу**

select "Оголошення"."Активне", users.username, users.phone, "Нерухомість"."Name", "Місто"."Назва","Оголошення"."Ціна продажу" from "Нерухомість"

left join "Оголошення" ON "Оголошення"."Нерухомість" = "Нерухомість"."Id"

inner join "Місто" ON "Місто"."Id" = "Нерухомість"."Місто"

inner join users ON users."Id" = "Нерухомість"."Owner"

where "Оголошення"."Активне" IS True and "Оголошення"."Вартість оренди" is Null;

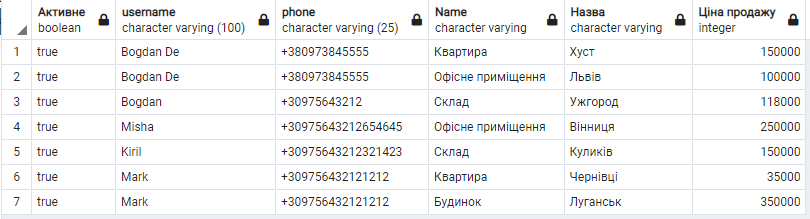


Рис 2.15 Виконання селекту “Активні оголошення продажу”

**Запит №3 Неактивні оголошення оренди**

select "Оголошення"."Активне", users.username, users.phone, "Нерухомість"."Name", users2.username, "Місто"."Назва" from users

inner join "Нерухомість" ON "Нерухомість"."Винаймач" = users."Id"

left join "Оголошення" ON "Оголошення"."Нерухомість" = "Нерухомість"."Id"

inner join "Місто" ON "Місто"."Id" = "Нерухомість"."Місто"

inner join users users2 ON users2."Id" = "Нерухомість"."Owner"

where "Оголошення"."Активне" IS False and "Оголошення"."Продаж" is False

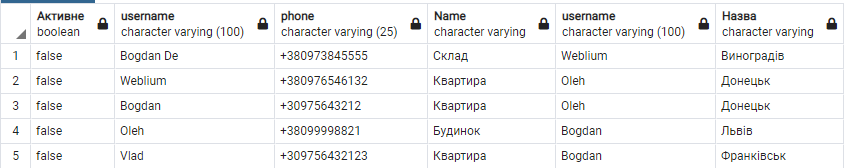


Рис 2.16 Виконання селекту “Неактивні оголошення оренди”

**Запит №4 Типи приміщення власники та місцезнаходження**

select "Нерухомість"."Name", "Місто"."Назва" as "Місто", users.username from "Нерухомість"

INNER JOIN "Місто" ON "Місто"."Id" = "Нерухомість"."Місто"

INNER JOIN users ON users."Id" = "Нерухомість"."Owner"

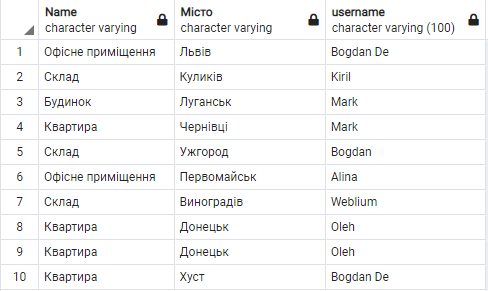


Рис 2.17 Виконання селекту “Типи приміщення власники та місцезнаходження ”

**Запит №5 Всі юзери які не є власниками або винаймачами**

select \* from users left join "Нерухомість" on users."Id" = "Нерухомість"."Owner" where "Нерухомість"."Owner" IS NULL;

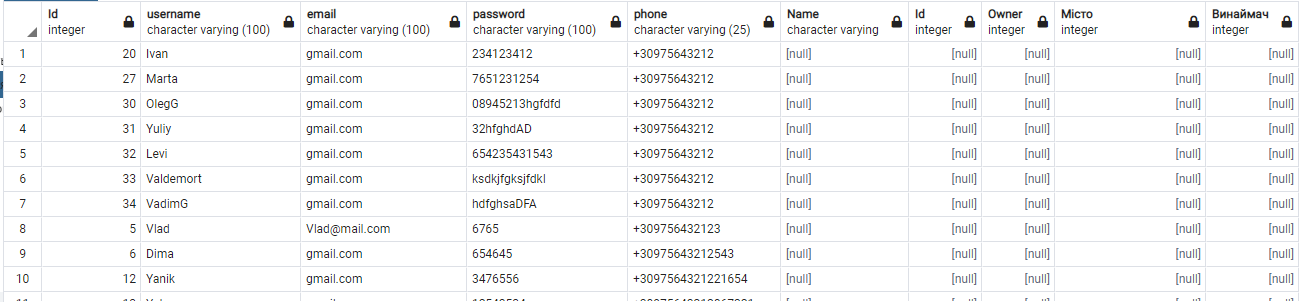


Рис 2.18 Виконання селекту “Всі юзери які не є власниками або винаймачами ”

**Запит №6 Приміщення які здані в оренду на певний місяць**

select

"Оголошення"."Продаж",

"Оголошення"."Вартість оренди",

"Нерухомість"."Name",

"Оголошення"."Дата Оренди",

"users"."username" AS "Орендар"

from "Оголошення"

inner join "Нерухомість" ON "Нерухомість"."Id" = "Оголошення"."Нерухомість"

inner join users as users on "Нерухомість"."Винаймач" = users."Id"

where "Оголошення"."Продаж" IS False

AND

"Оголошення"."Дата Оренди" Like 'August';

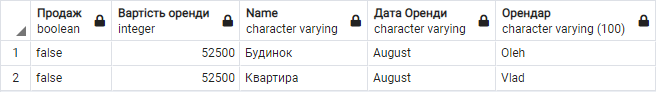


Рис 2.19 Виконання селекту “Приміщення які здані в оренду на певний місяць ”

**Висновки**

В результаті виконаної роботи було реалізовано базу даних для купівлі продажу та оренди приміщень з метою переведення бізнесу з офлайн режиму в онлайн для збільшення прибутку

Також була пророблена use case діаграма для двох користувачів яка наявно показує як працює база данних і полегшує орієнтацію по бізнес моделі

В тому числі була достеменно пророблена Ear- діаграма в якій присутні 4 таблиці і заповнені різними типами данних з різними звязками . Вона показує як звязані таблички одна з одною з метою полегшення вибору інформації для подальшого юзера

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мартин Грабер :SQL; Видавництво «Лори», 2016. -643 с.
2. [Моррисон М.](http://padabum.com/search.php?author=%D0%9C%D0%BE%D1%80%D1%80%D0%B8%D1%81%D0%BE%D0%BD%20%D0%9C) : Изучаем JavaScript; Видавництво «O`REILLY», 2012. – 606 c.
3. Веб-дизайн, структура сайту [Електронний ресурс]. – Режим доступу: <https://webstudio2u.net/ua/design-web/403-structure-models.html>.
4. Bootstrap [Електронний ресурс]. – Режим доступу: <https://timeweb.com/ru/community/articles/plyusy-i-minusy-bootstrap-1>.
5. Переваги MySQl [Електронний ресурс]. – Режим доступу: https://studfile.net/preview/5607354/page:3/ .
6. Переваги PhP [Електронний ресурс]. – Режим доступу: [https://blog.ithillel.ua/ua/articles/chomu-vybyraiut-7-perevah-movy-](https://blog.ithillel.ua/ua/articles/chomu-vybyraiut-7-perevah-movy-prohramuvannia-php) [prohramuvannia-php](https://blog.ithillel.ua/ua/articles/chomu-vybyraiut-7-perevah-movy-prohramuvannia-php).
7. Моделі даних [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%96_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) [D1%96\_%D0%B1%D0%B0%D0%B7\_%D0%B4%D0%B0%D0%BD%D0%B8](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%96_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)

[%D1%85](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%96_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85).

1. Типи даних MySQL [Електронний ресурс]. – Режим доступу: [https://uk.wikibooks.org/wiki/SQL/%D0%A2%D0%B8%D0%BF%D0%B8\_%D0](https://uk.wikibooks.org/wiki/SQL/%D0%A2%D0%B8%D0%BF%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85_MySQL)

[%B4%D0%B0%D0%BD%D0%B8%D1%85\_MySQL](https://uk.wikibooks.org/wiki/SQL/%D0%A2%D0%B8%D0%BF%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85_MySQL).

9. Інформаційні системи [Eлектронний ресурс]. — Режим доступу: <https://pidru4niki.com/1222090547713/informatika/informatsiyni_sistemi>

10. Моделі баз даних [Eлектронний ресурс]. — Режим доступу: [https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%B](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%96_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) [B%D1%96\_%D0%B1%D0%B0%D0%B7\_%D0%B4%D0%B0%D0%BD%D](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%96_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) [0%B8%D1%85](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%96_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)

11. НОРМАЛІЗАЦІЯ ВІДНОШЕНЬ [Електронний ресурс]. — Режим доступу: <https://life-prog.ru/ukr/1_331_normalizatsiya-vidnoshen.html>

12. Що таке діаграма варіантів використання [Електронний ресурс]. — Режим доступу: https://[www.visual-paradigm.com/guide/uml-unified-modeling-](http://www.visual-paradigm.com/guide/uml-unified-modeling-) language/what-is-use-case-diagram/

13. Що таке нормалізація бази даних у SQL Server? [Електронний ресурс]. — Режим доступу: [https://www.sqlshack.com/what-is-database-normalization-](https://www.sqlshack.com/what-is-database-normalization-in-sql-server/) [in-sql-server/](https://www.sqlshack.com/what-is-database-normalization-in-sql-server/)

**Додатки**

Лістинг А.1 Згенерований SQL

CREATE TABLE IF NOT EXISTS public.users

(

"Id" integer NOT NULL,

username character varying(100) COLLATE pg\_catalog."default" NOT NULL,

email character varying(100) COLLATE pg\_catalog."default" NOT NULL,

password character varying(100) COLLATE pg\_catalog."default" NOT NULL,

phone character varying(25) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Id" PRIMARY KEY ("Id")

);

CREATE TABLE IF NOT EXISTS public."Місто"

(

"Id" integer NOT NULL,

"Назва" character varying COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT "Місто\_pkey" PRIMARY KEY ("Id")

);

CREATE TABLE IF NOT EXISTS public."Нерухомість"

(

"Name" character varying COLLATE pg\_catalog."default" NOT NULL,

"Id" integer NOT NULL,

"Owner" integer,

"Місто" integer,

"Винаймач" integer,

CONSTRAINT neruhomist\_pkey PRIMARY KEY ("Id")

);

CREATE TABLE IF NOT EXISTS public."Оголошення"

(

"Ціна продажу" integer,

"Id" integer NOT NULL,

"Автор" integer NOT NULL,

"Нерухомість" integer NOT NULL,

"Вартість оренди" integer,

"Продаж" boolean NOT NULL,

"Дата Оренди" character varying COLLATE pg\_catalog."default",

"Активне" boolean,

"Опис Кв/м" integer,

CONSTRAINT "Оголошення\_pkey" PRIMARY KEY ("Id")

);

ALTER TABLE IF EXISTS public."Нерухомість"

ADD CONSTRAINT owner\_key FOREIGN KEY ("Owner")

REFERENCES public.users ("Id") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

CREATE INDEX IF NOT EXISTS fki\_f

ON public."Нерухомість"("Owner");

ALTER TABLE IF EXISTS public."Нерухомість"

ADD CONSTRAINT "Винайма\_key" FOREIGN KEY ("Винаймач")

REFERENCES public.users ("Id") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public."Нерухомість"

ADD CONSTRAINT "Місто\_key" FOREIGN KEY ("Місто")

REFERENCES public."Місто" ("Id") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public."Оголошення"

ADD CONSTRAINT "UserConstraints" FOREIGN KEY ("Автор")

REFERENCES public.users ("Id") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

ALTER TABLE IF EXISTS public."Оголошення"

ADD CONSTRAINT "НерухомістьConstra" FOREIGN KEY ("Нерухомість")

REFERENCES public."Нерухомість" ("Id") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID;

END;