

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
(повна назва інституту/факультету)

КАТЕДРА інформатики та програмної інженерії
(повна назва кафедри)

КУРСОВА РОБОТА

з дисципліни «Бази даних»
(назва дисципліни)

на тему: «База даних агенції нерухомости»

Студента 2 курсу ІІ-11 групи
спеціальности 121 «Інженерія програмного
забезпечення»

Лесіва В. І.
(прізвище та ініціали)

Керівник ас. Ліщук О. В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____

Кількість балів: _____ Оцінка ECTS _____

Члени комісії

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

Київ – 2023 рік

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет Інформатики та обчислювальної техніки
(повна назва)

Катедра Інформатики та програмної інженерії
(повна назва)

Дисципліна Бази даних

Курс 2 Група ІІІ-11 Семестр 3

**З А В Д А Н Н Я
НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Лесіву Владиславу Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи База даних агенції нерухомости

керівник роботи ас. Ліщук О. В.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

2. Строк подання студентом роботи 05.01.2023

3. Вихідні дані до роботи завдання на розробку бази даних для агенції, що працює з продавцями та покупцями нерухомости й допомагає їм у здійсненні огляду, купівлі-продажу, завірення договору тощо.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1)

Аналіз предметного середовища

2) Побудова ER-моделі

3) Побудова реляційної схеми з ER-моделі

4) Створення бази даних, у форматі обраної системи управління базою даних

5) Створення користувачів бази даних

6) Імпорт даних з використанням засобів СУБД в створену базу даних

7) Створення мовою SQL запитів

8) Оптимізація роботи запитів

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Дата видачі завдання 08.11.2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання курсового проєкту	Строк виконання етапів проєкту	Примітка
1	Аналіз предметного середовища	21.11.2022	
2	Побудова ER-моделі	21.11.2022	
3	Побудова реляційної схеми з ER-моделі	05.01.2023	
4	Створення бази даних, у форматі обраної системи управління базою даних	05.01.2023	
5	Створення користувачів бази даних	05.01.2023	
6	Імпорт даних з використанням засобів СУБД в створену базу даних	05.01.2023	
7	Створення мовою SQL запитів	05.01.2023	
8	Оптимізація роботи запитів	05.01.2023	
9	Оформлення пояснювальної записки	05.01.2023	
10	Захист курсової роботи		

Студент

(підпис)
Лесів В.І.
(прізвище та ініціали)

Керівник роботи

(підпис)
Ліщук О.В.
(прізвище та ініціали)

АНОТАЦІЯ

Пояснювальна записка до курсової роботи: 47 сторінок, 32 рисунки, 1 таблиця, 4 посилання.

Об'єкт дослідження: база даних агенції нерухомости.

Мета роботи: закріплення навичок з проєктування, реалізації реляційних баз даних та їх використання для практичних задач.

Проведено аналіз предметного середовища, визначено сутності, атрибути та зв'язки між об'єктами. Побудовано ER-модель предметного середовища, реляційну схему бази даних, виділено первинні та зовнішні ключі, визначено обмеження для підтримки цілісності даних. Розроблено скрипти для побудови спроектованої бази даних, імпортовано дані в неї, виконано різноманітні запити до неї.

Виконана програмна реалізація бази даних агенції нерухомости (варіант 18).

ЗМІСТ

ВСТУП.....	7
1. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ	8
2. ПОСТАНОВКА ЗАВДАННЯ	9
3. ER-ДІАГРАМА.....	10
3.1. Бізнес правила	10
3.2. Вибір сутностей	10
3.3. Набори атрибутів сутностей.....	11
4. РЕЛЯЦІЙНА МОДЕЛЬ БАЗИ ДАНИХ.....	14
4.1. Побудова необхідних відношень та визначення первинних та зовнішніх ключів.	14
4.2. Визначення обмежень цілісності для спроектованих відношень	14
5. РЕАЛІЗАЦІЯ БАЗИ ДАНИХ.....	16
5.1. Створення бази даних у форматі системи управління базою даних MySQL. 16	
5.2. Імпортування даних в таблицю	21
6. СТВОРЕННЯ КОРИСТУВАЧІВ БАЗИ ДАНИХ.....	27
6.1. Створення адміністратора. Адміністратор володіє всіма правами і можливостями.....	27
6.2. Створення працівника агенції. Працівник володіє селектом на всі таблиці й редагуванням на деякі конкретні.....	27
7. SQL ЗАПИТИ	28
7.1. SQL-запити.....	28
7.1.1. Кількість нерухомості за власником.	28
7.1.2. Нерухомість з паркомісцем.	28
7.1.3. Вільні працівники.....	29
7.1.4. Найближче до парку й зупинки.	29
7.1.5. Клієнти, які мають успішні огляди.....	30
7.1.6. Скільки угод уклав нотаріус.....	30
7.1.7. Номери телефонів відвідувачів.....	31
7.1.8. Скільки номерів телефонів є в БД.	31

7.1.9. Кількість оглядів нерухомости.	32
7.1.10. Продана нерухомість за містами	32
7.1.11. Середня виручка відділу.....	33
7.1.12. Відсоток від продажу працівниками.	34
7.1.13. Дзвінок від адміністратора	34
7.1.14. Скільки відсотків від типу продано.....	35
7.2. Функції й збережені процедури.	35
7.2.1. Кількість оглядів за роком.....	35
7.2.2. Нерухомість за ідентифікатором відділу.	36
7.2.3. Різниця між ціною й угодою	37
7.2.4. Ціна за квадратний метр.	38
7.3. Тригери.	39
7.3.1. Тригер на вставлення й перевірку дати.....	39
7.3.2. Тригер на оновлення й зміну ціни у відсотках.....	40
7.3.3. Тригер на видалення й кількість відповідальности	41
7.4. Представлення.....	42
7.4.1. Розширення огляду.....	43
7.4.2. Представлення нерухомости.	44
7.5. Індокси.	45
ВИСНОВОК.....	46
ПЕРЕЛІК ПОСИЛАНЬ	47

ВСТУП

На сучасному етапі бази даних стали невід'ємною частиною нашого повсякденного життя. Вони введені й успішно використовуються практично в усіх галузях людської діяльності. У сучасних умовах однією з найбільш розповсюджених задач, що виникає при користуванні комп'ютером, є використання інформаційних систем, заснованих на базах даних.

Обрана мною тема є актуальною, адже щодня в агентствах нерухомости по всьому світу відбуваються безліч процесів, оглядів у угод, які зручно утримувати й отримувати за допомогою бази даних.

Дана робота присвячена вивченню розробки програмного забезпечення з SQL, і стосується проєктування бази даних для збереження інформації в агенції нерухомости, що допоможе працівникам й адміністраторам таких агенцій. Задача полягає в аналізі предметної області, побудови ER-діаграми та реляційної схеми бази даних, написанні SQL-скриптів та запитів до неї.

Виконання роботи проведено з допомогою MySQL, оскільки вона є потужним засобом побудови БД, гнучкою, підтримує різні ОС, а також вона є доволі захищеною, що особливо важливо для чутливих даних агенції.

1. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

Основним об'єктом дослідження предметної області є агенція нерухомости та процеси, що відбуваються в ній.

Агенція нерухомости займається продажем нерухомости. Агентство має декілька відділів. Власники нерухомости звертаються до одного з відділів і подають нерухомість для продажу. Нерухомість закріплюється за конкретним працівником агентства, який супроводжує її до моменту угоди. В анкеті нерухомости заповнюються тип, вартість, адреса, площа, кількість кімнат тощо, а також додаткові вигоди, як-то паркомісце чи зелений двір.

Клієнти - потенційні покупці - звертаються до агентства зі своїми бажаннями й вимогами до нерухомости, відповідно до чого агентом підбирається й пропонується на огляд те чи інше житло.

Якщо клієнт вважає, що житло його задовільняє і він хоче придбати його, то в результаті огляду відбувається угода купівлі-продажу. Угода затверджується довіреним нотаріусом агентства, який вносить угоду в державний реєстр.

Серед іншого адміністрація агентства може видавати статистику про середню вартість нерухомости, відсоток будинків і квартир до нерухомости в загальному тощо.

2. ПОСТАНОВКА ЗАВДАННЯ

Метою даної роботи є розробка бази даних для збереження даних в процесі роботи агенції нерухомости. Тобто організація даних в базі має бути такою, аби можна було зручно отримувати доступ до даної нерухомости, клієнтів, продавців, а також структурувати сам процес продажу.

Виділімо основні задачі:

Адміністрація агенції може:

- Змінювати склад працівників агенції, відділів, довірених нотаріусів агенції;
- Виконувати технічні дії щодо нерухомости, тобто додавати, видаляти, змінювати дані в базі;
- Видавати статистику щодо продажів.

Працівники агенції можуть:

- Додавати\видаляти клієнтів, нерухомості, її власників, а також записи до оглядів нерухомостей;
- Переглядати дані з усієї бази, тобто дізнаватися інформацію щодо адміністрації, угод, нотаріусів та відділів;

3. ER-ДІАГРАМА

Після аналізу було виділено такі сутності та зв'язки між ними.

3.1. Бізнес правила:

- Нерухомість може бути двох видів: «Flat» (Квартира) й «House» (Будинок);
- Вид угоди може бути лише двох видів: «Sale» (Продаж) й «Rent» (Оренда);
- Результат огляду може бути двох видів: «OK» й «No» (Ні);
- Дата огляду може бути як внесена по факту, так і запланована, та якщо дата більша за сьогоднішню, видається попередження;
- Нерухомість може мати лише одного власника (власницю);
- Працівник (-ця) може працювати лише в одному відділі.

3.2. Вибір сутностей:

- Адміністрація агентства;
- Додаткові вигоди;
- Клієнт;
- Угоди;
- Відділ;
- Довірені нотаріуси;
- Нерухомість;
- Власники нерухомості;
- Огляд нерухомості;
- Працівники агентства.

3.3. Набори атрибутів сутностей:

Таблиця 3.1 – Сутності та їхні атрибути.

Сутність	Атрибути
Адміністрація агентства (administrators)	ПІБ (fullName) Робочий телефон (phone)
Додаткові вигоди (amenities)	Наявність паркомісця (parking) Відстань до зупинки (busStopDistance) Двір, вільний від авто (garden) Відстань до зеленої зони (parkDistance) Доплата за поточні меблі (keepFurniturePrice)
Клієнт (clients)	ПІБ (fullName) Номер телефону (phone) Вимоги до нерухомости (requirements)
Угоди (deals)	Тип угоди (dealType) Дата угоди (dealDate) Сума угоди (dealTotal) Номер документу про оплату (receiptNumber)
Відділ (departments)	Назва (depName) Місто (city) Адреса (address) Номер телефону (phone) E-mail (email)

Продовження таблиці 3.1.

Довірені нотаріуси (notaries)	Номер ліцензії (license) ПІБ (fullName) Номер телефону (phone) E-mail (email)
Нерухомість (realties)	Тип нерухомості (realtyType) Вид угоди (dealType) Затверджена вартість (price) Місто (city) Адреса (address) Площа (area) Кількість кімнат (rooms)
Власники нерухомості (realtyOwners)	ПІБ (fullName) Номер телефону (phone)
Огляд нерухомості (surveys)	Дата проведення (surveyDate) Час проведення (surveyTime) Результат (result)
Працівники агентства (workers)	ПІБ (fullName) Номер телефону (phone)

Сутність **Відділ** пов'язана *один до багатьох* із сутністю **Працівники агентства**, адже декілька працівників можуть працювати в одному відділі.

Сутність **Працівники агентства** пов'язана *один до багатьох* із сутністю **Огляд нерухомості**, адже декілька оглядів можуть проводитися одним працівником.

Сутність **Клієнт** пов'язана *один до багатьох* із сутністю **Огляд нерухомості**, адже декілька оглядів можуть відбуватися для одного клієнта.

Сутність **Нерухомість** пов'язана *один до багатьох* із сутністю **Огляд нерухомості**, адже декілька оглядів можуть стосуватися однієї нерухомості.

Сутність **Працівники агентства** пов'язана *один до багатьох* із сутністю **Нерухомість**, адже один працівник може бути відповідальним за декілька нерухомостей.

Сутність **Додаткові вигоди** пов'язана *один до одного* із сутністю **Нерухомість**, адже для однієї нерухомості може бути лише один набір вигод.

Сутність **Адміністрація агентства** пов'язана *один до багатьох* із сутністю **Нерухомість**, адже один адміністратор може головувати щодо декількох нерухомостей.

Сутність **Власники нерухомості** пов'язана *один до багатьох* із сутністю **Нерухомість**, адже один власник може мати декілька нерухомостей.

Сутність **Нерухомість** пов'язана *один до багатьох* із сутністю **Угоди**, адже одна нерухомість може мати декілька угод упродовж якогось періоду часу.

Сутність **Довірені нотаріуси** пов'язана *один до багатьох* із сутністю **Угоди**, адже один нотаріус може укладати декілька угод.

ER-діаграма:

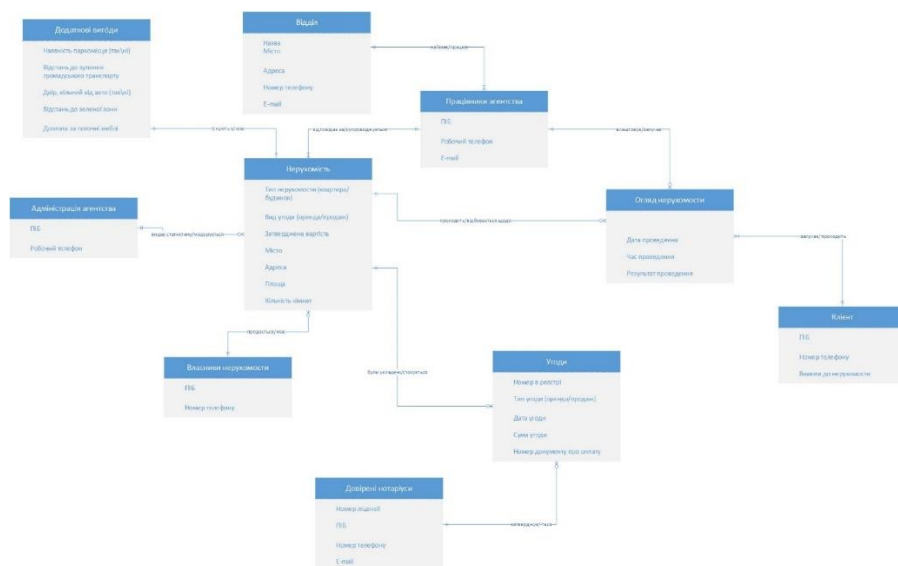


Рисунок 3.1 – ER-діаграма

4. РЕЛЯЦІЙНА МОДЕЛЬ БАЗИ ДАНИХ

4.1. Побудова необхідних відношень та визначення первинних та зовнішніх ключів.

На рисунку можна побачити, що база даних знаходиться у третій нормальній формі, оскільки поля таблиці декомпозовані, атрибути функційно повно залежать від первинного ключа, кожен неключовий атрибут не є транзитивно залежним від первинного ключа.

4.2. Визначення обмежень цілісності для спроектованих відношень

1. Додаємо до таблиць обмеження відповідно до того, як логічно поєднані таблиці. В одних випадках при видаленні батьківського рядка зовнішні ключі варто видалити каскадно, в інших – установити значення NULL. До прикладу,

```
FOREIGN KEY (adminId) REFERENCES administrators(id) ON  
UPDATE CASCADE ON DELETE SET NULL,  
FOREIGN KEY (ownerId) REFERENCES realtyOwners(id) ON UPDATE  
CASCADE ON DELETE CASCADE;
```

2. Обов'язкові атрибути таблиць мають обмеження NOT NULL для запобігання помилок при роботі з даними, ті, що мають мати унікальні значення – обмеження UNIQUE;

3. Обмежуємо текстові поля максимальною кількістю символів, до прикладу, VARCHAR(20);

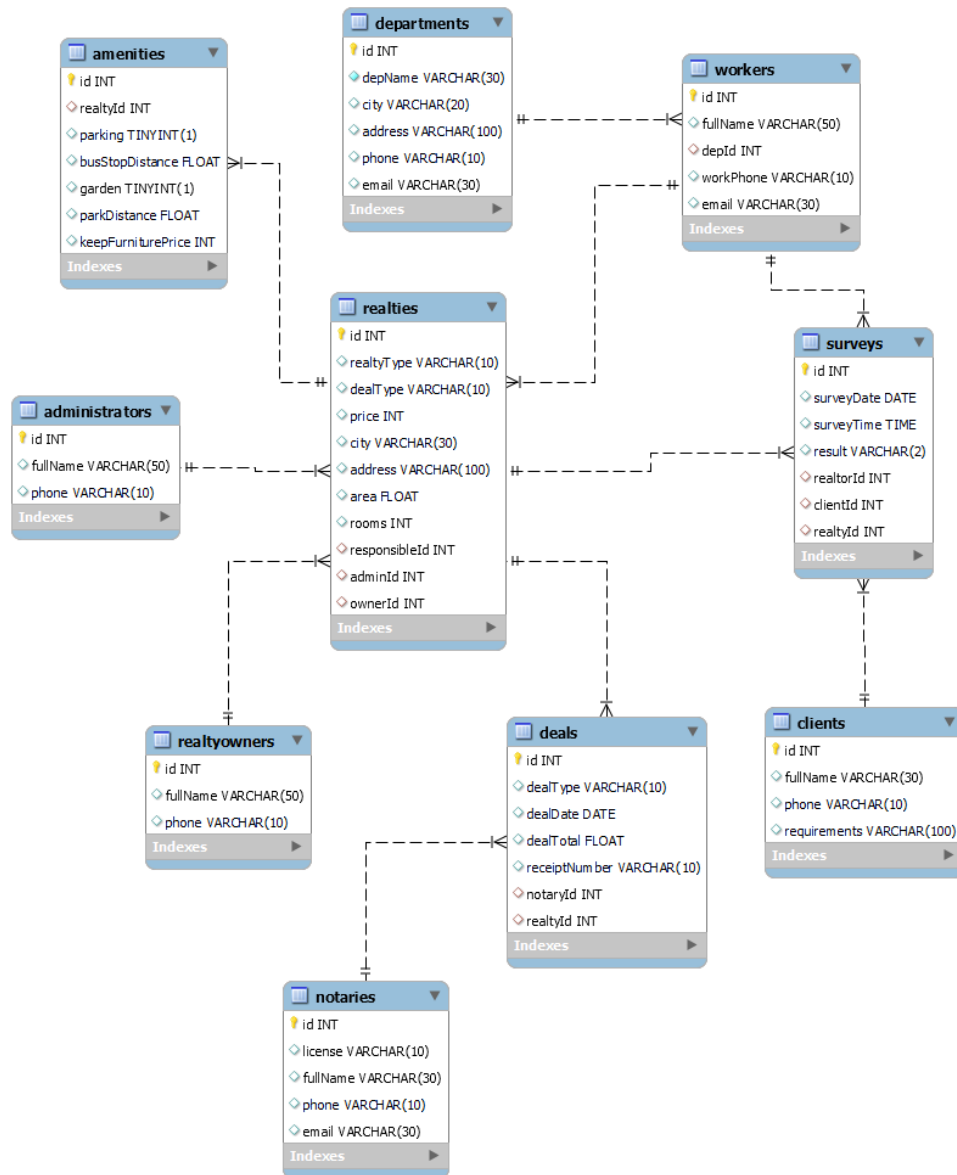


Рисунок 4.1 – Реляційна модель бази даних

5. РЕАЛІЗАЦІЯ БАЗИ ДАНИХ

5.1. Створення бази даних у форматі системи управління базою даних MySQL.

```
CREATE SCHEMA `realEstateAgency` ;
```

```
USE realEstateAgency;
```

```
CREATE TABLE departments(  
id INT PRIMARY KEY AUTO_INCREMENT,  
depName VARCHAR(30) NOT NULL,  
city VARCHAR(20),  
address VARCHAR(100),  
phone VARCHAR(10),  
email VARCHAR(30)  
);
```

```
CREATE TABLE workers(  
id INT PRIMARY KEY AUTO_INCREMENT,  
fullName VARCHAR(50),  
depId INT,  
workPhone VARCHAR(10),  
email VARCHAR(30),  
FOREIGN KEY (depId) REFERENCES departments(id) ON UPDATE  
CASCADE ON DELETE CASCADE
```


);

```
CREATE TABLE administrators(  
id INT PRIMARY KEY AUTO_INCREMENT,  
fullName VARCHAR(50),  
phone VARCHAR(10)  
);
```

```
CREATE TABLE realtyOwners(  
id INT PRIMARY KEY AUTO_INCREMENT,  
fullName VARCHAR(50),  
phone VARCHAR(10)  
);
```

```
CREATE TABLE realties(  
id INT PRIMARY KEY AUTO_INCREMENT,  
realtyType VARCHAR(10),  
dealType VARCHAR(10),  
price INT,  
city VARCHAR(30),  
address VARCHAR(100),  
area FLOAT,
```

```

rooms INT,

responsibleId INT,

adminId INT,

ownerId INT,

FOREIGN KEY (responsibleId) REFERENCES workers(id) ON UPDATE
CASCADE ON DELETE SET NULL,

FOREIGN KEY (adminId) REFERENCES administrators(id) ON UPDATE
CASCADE ON DELETE SET NULL,

FOREIGN KEY (ownerId) REFERENCES realtyOwners(id) ON UPDATE
CASCADE ON DELETE CASCADE

);

```

```

ALTER TABLE realties ADD CONSTRAINT realType CHECK
(realType="House" OR realType="Flat");

```

```

ALTER TABLE realties ADD CONSTRAINT deaType CHECK
(dealType="Sale" OR dealType="Rent");

```

```

CREATE TABLE amenities(

id INT PRIMARY KEY AUTO_INCREMENT,

realtyId INT UNIQUE,

parking BOOL,

busStopDistance FLOAT,

garden BOOL,

parkDistance FLOAT,

```

```
keepFurniturePrice INT,  
  
FOREIGN KEY (realtyId) REFERENCES realties(id) ON UPDATE CASCADE  
ON DELETE CASCADE  
  
);
```

```
CREATE TABLE clients(  
  
id INT PRIMARY KEY AUTO_INCREMENT,  
  
fullName VARCHAR(30),  
  
phone VARCHAR(10),  
  
requirements VARCHAR(100)  
  
);
```

```
CREATE TABLE surveys(  
  
id INT PRIMARY KEY AUTO_INCREMENT,  
  
surveyDate DATE,  
  
surveyTime TIME,  
  
result VARCHAR(2),  
  
realtorId INT,  
  
clientId INT,  
  
realtyId INT,  
  
FOREIGN KEY (realtorId) REFERENCES workers(id) ON UPDATE CASCADE  
ON DELETE SET NULL,
```

FOREIGN KEY (clientId) REFERENCES clients(id) ON UPDATE CASCADE
ON DELETE SET NULL,

FOREIGN KEY (realtyId) REFERENCES realties(id) ON UPDATE CASCADE
ON DELETE CASCADE

);

ALTER TABLE surveys ADD constraint res CHECK (result="OK" OR
result="NO");

CREATE TABLE notaries(

id INT PRIMARY KEY AUTO_INCREMENT,

license VARCHAR(10) UNIQUE,

fullName VARCHAR(30),

phone VARCHAR(10),

email VARCHAR(30)

);

CREATE TABLE deals(

id INT PRIMARY KEY AUTO_INCREMENT,

dealType VARCHAR(10),

dealDate DATE,

dealTotal FLOAT,

receiptNumber VARCHAR(10),

notaryId INT,

realtyId INT,

FOREIGN KEY (notaryId) REFERENCES notaries(id) ON UPDATE CASCADE
ON DELETE SET NULL,

FOREIGN KEY (realtyId) REFERENCES realties(id) ON UPDATE CASCADE
ON DELETE CASCADE

);

ALTER TABLE deals ADD CONSTRAINT deaTyp CHECK (dealType="Sale"
OR dealType="Rent");

#	Time	Action	Message	Duration / Fesck
1	14:16:05	drop schema if exists 'realEstateAgency'	10 row(s) affected	0.078 sec
2	14:16:05	CREATE SCHEMA 'realEstateAgency'	1 row(s) affected	0.000 sec
3	14:16:05	USE realEstateAgency	0 row(s) affected	0.000 sec
4	14:16:05	CREATE TABLE departments(id INT PRIMARY KEY AUTO_INCREMENT, depName VARCHAR(30) NOT NULL...	0 row(s) affected	0.016 sec
5	14:16:05	CREATE TABLE workers(id INT PRIMARY KEY AUTO_INCREMENT, fullName VARCHAR(50), depId INT, work...	0 row(s) affected	0.015 sec
6	14:16:05	CREATE TABLE administrators(id INT PRIMARY KEY AUTO_INCREMENT, fullName VARCHAR(50), phone VA...	0 row(s) affected	0.047 sec
7	14:16:05	CREATE TABLE realtyOwners(id INT PRIMARY KEY AUTO_INCREMENT, fullName VARCHAR(50), phone VA...	0 row(s) affected	0.016 sec
8	14:16:05	CREATE TABLE realties(id INT PRIMARY KEY AUTO_INCREMENT, realtyType VARCHAR(10), dealType VAR...	0 row(s) affected	0.031 sec
9	14:16:05	ALTER TABLE realties ADD CONSTRAINT realType CHECK (realtyType="House" OR realtyType="Flat")	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
10	14:16:05	ALTER TABLE realties ADD CONSTRAINT deaType CHECK (dealType="Sale" OR dealType="Rent")	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
11	14:16:05	CREATE TABLE amenities(id INT PRIMARY KEY AUTO_INCREMENT, realtyId INT UNIQUE, parking BOOL, bu...	0 row(s) affected	0.015 sec
12	14:16:05	CREATE TABLE clients(id INT PRIMARY KEY AUTO_INCREMENT, fullName VARCHAR(30), phone VARCHAR...	0 row(s) affected	0.000 sec
13	14:16:05	CREATE TABLE surveys(id INT PRIMARY KEY AUTO_INCREMENT, surveyDate DATE, surveyTime TIME, res...	0 row(s) affected	0.031 sec
14	14:16:05	ALTER TABLE surveys ADD constraint res CHECK (result="OK" OR result="NO")	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
15	14:16:05	CREATE TABLE notaries(id INT PRIMARY KEY AUTO_INCREMENT, license VARCHAR(10) UNIQUE, fullName...	0 row(s) affected	0.016 sec
16	14:16:05	CREATE TABLE deals(id INT PRIMARY KEY AUTO_INCREMENT, dealType VARCHAR(10), dealDate DATE, d...	0 row(s) affected	0.031 sec
17	14:16:05	ALTER TABLE deals ADD CONSTRAINT deaTyp CHECK (dealType="Sale" OR dealType="Rent")	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec

Рисунок 5.1.1 – Створення бази даних

5.2. Імпортування даних в таблицю

Для імпортування даних у СУБД MySQL існує два методи:

1. Імпортувати дані для кожної таблиці з .csv-файлів

USE realEstateAgency;

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/departments.csv'

```

INTO TABLE departments
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
ESCAPED BY '"'
LINES TERMINATED BY ';'
(depName,city,address,phone,email);

```

```

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/workers.csv'
INTO TABLE workers
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
ESCAPED BY '"'
LINES TERMINATED BY ';'
(fullName,depId,workPhone,email);

```

```

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/administrators.csv'
INTO TABLE administrators
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
ESCAPED BY '"'
LINES TERMINATED BY ';'

```

(fullName,phone);

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/realtyOwners.csv'
```

```
INTO TABLE realtyOwners
```

```
FIELDS TERMINATED BY ','
```

```
ENCLOSED BY ''''
```

```
ESCAPED BY ''''
```

```
LINES TERMINATED BY ';'

```

(fullName,phone);

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/realties.csv'
```

```
INTO TABLE realties
```

```
FIELDS TERMINATED BY ','
```

```
ENCLOSED BY ''''
```

```
ESCAPED BY ''''
```

```
LINES TERMINATED BY ';'

```

(realtyType,dealType,price,city,address,area,rooms,responsibleId,adminId,ownerId);

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/amenities.csv'
```

INTO TABLE amenities

FIELDS TERMINATED BY ','

ENCLOSED BY ''''

ESCAPED BY ''''

LINES TERMINATED BY ';'

(realtyId,parking,busStopDistance,garden,parkDistance,keepFurniturePrice);

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/clients.csv'

INTO TABLE clients

FIELDS TERMINATED BY ','

ENCLOSED BY ''''

ESCAPED BY ''''

LINES TERMINATED BY ';'

(fullName,phone,requirements);

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/surveys.csv'

INTO TABLE surveys

FIELDS TERMINATED BY ','

ENCLOSED BY ''''

ESCAPED BY ''''

LINES TERMINATED BY ';'

(surveyDate,surveyTime,result,realtorId,clientId,realtyId);

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/notaries.csv'
```

```
INTO TABLE notaries
```

```
FIELDS TERMINATED BY ','
```

```
ENCLOSED BY '"'
```

```
ESCAPED BY '"'
```

```
LINES TERMINATED BY ';'

```

```
(license,fullName,phone,email);
```

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/deals.csv'
```

```
INTO TABLE deals
```

```
FIELDS TERMINATED BY ','
```

```
ENCLOSED BY '"'
```

```
ESCAPED BY '"'
```

```
LINES TERMINATED BY ';'

```

```
(dealType,dealDate,dealTotal,receiptNumber,notaryId,realtyId);
```

#	Time	Action	Message
21	14:17:01	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/administrators.csv' INTO TABLE ad...	3 row(s) affected Records: 3 Deleted: 0 Skipped: 0 Warnings: 0
22	14:17:01	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/realtyOwners.csv' INTO TABLE real...	4 row(s) affected Records: 4 Deleted: 0 Skipped: 0 Warnings: 0
23	14:17:01	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/realities.csv' INTO TABLE realities FI...	4 row(s) affected Records: 4 Deleted: 0 Skipped: 0 Warnings: 0
24	14:17:01	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/amenities.csv' INTO TABLE ameniti...	3 row(s) affected Records: 3 Deleted: 0 Skipped: 0 Warnings: 0
25	14:17:01	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/clients.csv' INTO TABLE clients FIE...	4 row(s) affected Records: 4 Deleted: 0 Skipped: 0 Warnings: 0
26	14:17:01	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/surveys.csv' INTO TABLE surveys ...	5 row(s) affected Records: 5 Deleted: 0 Skipped: 0 Warnings: 0
27	14:17:01	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/notaries.csv' INTO TABLE notaries ...	3 row(s) affected Records: 3 Deleted: 0 Skipped: 0 Warnings: 0
28	14:17:01	LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/deals.csv' INTO TABLE deals FIEL...	2 row(s) affected Records: 2 Deleted: 0 Skipped: 0 Warnings: 0

Рисунок 5.2.1 – Імпортування даних з CSV-файлу

2. Імпортувати дані зі скрипта

```
INSERT INTO administrators(fullName,phone)
```

```
VALUES ("Joe Biden","0987654321"),
```

```
("Yaremchuk Nazariy","0372037203");
```

```
INSERT INTO notaries(license,fullName,phone,email)
```

```
VALUES ("df-0009-rt","Ivasiuk  
Volodymyr","0372564676","rutaa@gmail.cv.ua");
```

60	14:19:18	INSERT INTO administrators(fullName,phone) VALUES ("Joe Biden","0987654321"), ("Yaremchuk Nazariy","0372037203");	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0
61	14:19:18	INSERT INTO notaries(license,fullName,phone,email) VALUES ("df-0009-rt","Ivasiuk Volodymyr","0372564676","rutaa@gmail.cv.ua");	1 row(s) affected

Рисунок 5.2.2 – Імпортування даних за допомогою скрипту

6. СТВОРЕННЯ КОРИСТУВАЧІВ БАЗИ ДАНИХ

6.1. Створення адміністратора. Адміністратор володіє всіма правами і можливостями.

```
CREATE USER IF NOT EXISTS "admin1"@"localhost" IDENTIFIED BY "ad1pass";
```

```
GRANT ALL ON realestateagency.* TO "admin1"@"localhost";
```

✓	32	14:28:24	CREATE USER IF NOT EXISTS "admin1"@"localhost" IDENTIFIED BY "ad1pass"	0 row(s) affected
✓	33	14:28:24	GRANT ALL ON realestateagency.* TO "admin1"@"localhost"	0 row(s) affected

Рисунок 6.1.1 – Створення користувача адміністратора

6.2. Створення працівника агенції. Працівник володіє селектом на всі таблиці й редагуванням на деякі конкретні.

```
CREATE USER IF NOT EXISTS "worker1"@"localhost" IDENTIFIED BY "wk1pass";
```

```
GRANT SELECT ON realestateagency.* TO "worker1"@"localhost";
```

```
GRANT ALL ON realestateagency.amenities TO "worker1"@"localhost";
```

```
GRANT ALL ON realestateagency.clients TO "worker1"@"localhost";
```

```
GRANT ALL ON realestateagency.realities TO "worker1"@"localhost";
```

```
GRANT ALL ON realestateagency.realtyOwners TO "worker1"@"localhost";
```

```
GRANT ALL ON realestateagency.surveys TO "worker1"@"localhost";
```

✓	34	14:28:24	CREATE USER IF NOT EXISTS "worker1"@"localhost" IDENTIFIED BY "wk1pass"	0 row(s) affected
✓	35	14:28:24	GRANT SELECT ON realestateagency.* TO "worker1"@"localhost"	0 row(s) affected
✓	36	14:28:24	GRANT ALL ON realestateagency.amenities TO "worker1"@"localhost"	0 row(s) affected
✓	37	14:28:24	GRANT ALL ON realestateagency.clients TO "worker1"@"localhost"	0 row(s) affected
✓	38	14:28:24	GRANT ALL ON realestateagency.realities TO "worker1"@"localhost"	0 row(s) affected
✓	39	14:28:24	GRANT ALL ON realestateagency.realtyOwners TO "worker1"@"localhost"	0 row(s) affected
✓	40	14:28:24	GRANT ALL ON realestateagency.surveys TO "worker1"@"localhost"	0 row(s) affected

Рисунок 6.2.1 – Створення користувача працівника

7. SQL ЗАПИТИ

7.1. SQL-запити.

7.1.1. Кількість нерухомості за власником.

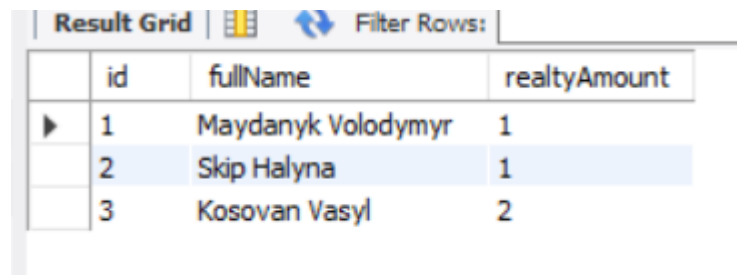
Запит на визначення того, скільки нерухомості має власник.

```
SELECT    realtyowners.id, realtyowners.fullName,    COUNT(ownerid)    as
realtyAmount

FROM realtyowners, realties

WHERE realtyowners.id=realties.ownerId

GROUP BY ownerId;
```



	id	fullName	realtyAmount
▶	1	Maydanyk Volodymyr	1
	2	Skip Halyna	1
	3	Kosovan Vasyl	2

Рисунок 7.1 – Кількість нерухомості за власником

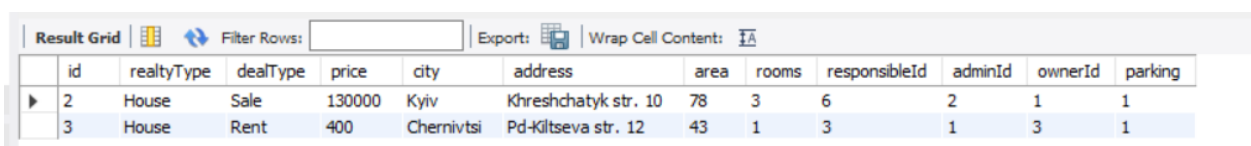
7.1.2. Нерухомість з паркомісцем.

Запит на визначення нерухомостей, які мають паркомісця.

```
SELECT realties.*, amenities.parking

FROM amenities, realties

WHERE parking=1 AND realties.id=amenities.realtyId;
```



	id	realtyType	dealType	price	city	address	area	rooms	responsibleId	adminId	ownerId	parking
▶	2	House	Sale	130000	Kyiv	Khreshchatyk str. 10	78	3	6	2	1	1
	3	House	Rent	400	Chernivtsi	Pd-Kiltseva str. 12	43	1	3	1	3	1

Рисунок 7.2 – Нерухомість з паркомісцем

7.1.3. Вільні працівники.

Запит на визначення працівників, які не відповідають за жодну нерухомість.

```
SELECT workers.id, fullName, depId, address
```

```
FROM workers, departments
```

```
WHERE workers.id NOT IN (SELECT responsibleId FROM realties) AND  
depId=departments.id;
```

	id	fullName	depId	address
►	2	Vlayko Anton	1	Politekhnichna str. 41
	7	Loshak Roman	4	Yaroslaviv Val str. 12
	5	Topala Tymur	6	Universytetska str. 1

Рисунок 7.3 – Вільні працівники

7.1.4. Найближче до парку й зупинки.

Запит на сортування нерухомостей з відомостями про відстані до парку й зупинки за найкращою сумою цих параметрів.

```
SELECT realties.id, realtyType, city, address, busStopDistance, parkDistance
```

```
FROM realties, amenities
```

```
WHERE realties.id=amenities.realtyId
```

```
ORDER BY (busStopDistance+parkDistance);
```

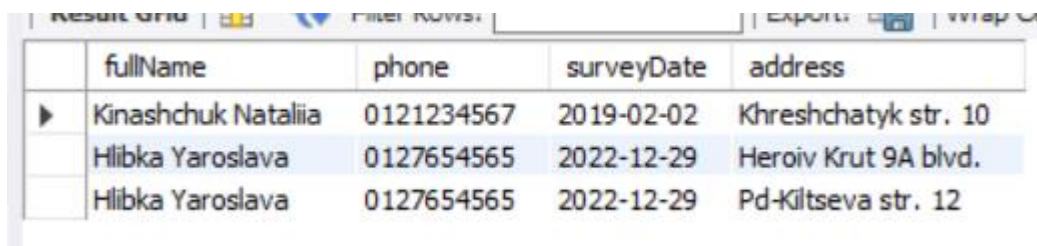
	id	realtyType	city	address	busStopDistance	parkDistance
►	1	Flat	Chernivtsi	Heroiv Krut 9A blvd.	100	450
	3	House	Chernivtsi	Pd-Kiltseva str. 12	570	344
	2	House	Kyiv	Khreshchatyk str. 10	1200	10

Рисунок 7.4 – Найближче до парку й зупинки

7.1.5. Клієнти, які мають успішні огляди.

Запит на вибірку клієнтів і їх номерів телефонів, яким сподобався огляд нерухомости, з метою, до прикладу, передзвонити і спитати про подальші дії.

```
SELECT fullName, phone, surveyDate, address
FROM clients, surveys, realties
WHERE clientId=clients.id AND result="OK" AND realtyId=realties.id
ORDER BY surveyDate;
```



	fullName	phone	surveyDate	address
▶	Kinashchuk Nataliia	0121234567	2019-02-02	Khreshchatyk str. 10
	Hlibka Yaroslava	0127654565	2022-12-29	Heroiv Krut 9A blvd.
	Hlibka Yaroslava	0127654565	2022-12-29	Pd-Kiltseva str. 12

Рисунок 7.5 - Клієнти, які мають успішні огляди.

7.1.6. Скільки угод уклав нотаріус.

Запит на з'ясування того, скільки угод уклав нотаріус, з метою знаходження найпопулярнішого, до прикладу.

```
SELECT notaries.id, fullName, COUNT(notaryId) AS successDeals
FROM notaries, deals
WHERE notaries.id=notaryId
GROUP BY notaryId
ORDER BY COUNT(notaryId) DESC;
```

	id	fullName	successDeals
▶	2	Lana Del Rey	1
	3	Trisha Paytas	1

Рисунок 7.6 – Скільки угод уклав нотаріус

7.1.7. Номери телефонів відвідувачів.

Запит на вибірку номерів телефонів покупців і продавців з метою якоїсь розсилки для клієнтів загалом.

```
SELECT fullName,phone
```

```
FROM realtyOwners
```

```
UNION
```

```
SELECT fullName,phone
```

```
FROM clients;
```

	fullName	phone
▶	Maydanyk Volodymyr	076543212
	Skip Halyna	0765678789
	Kosovan Vasyl	0768900909
	Halytska Yuliia	0761111111
	Kinashchuk Nataliia	0121234567
	Hlibka Yaroslava	0127654565
	Kharyton Olena	3453452312
	Lohvynenko Bohdan	0131233333

Рисунок 7.7 - Номери телефонів відвідувачів

7.1.8. Скільки номерів телефонів є в БД.

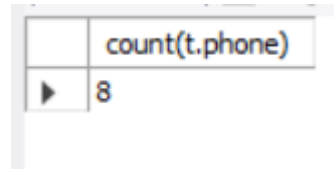
Запит, що рахує кількість номерів телефонів з попереднього запити.

```
SELECT count(t.phone)
```

```

FROM (SELECT fullName,phone
FROM realtyOwners
UNION
SELECT fullName,phone
FROM clients) AS t;

```



	count(t.phone)
▶	8

Рисунок 7.8 – Скільки номерів телефонів є в БД

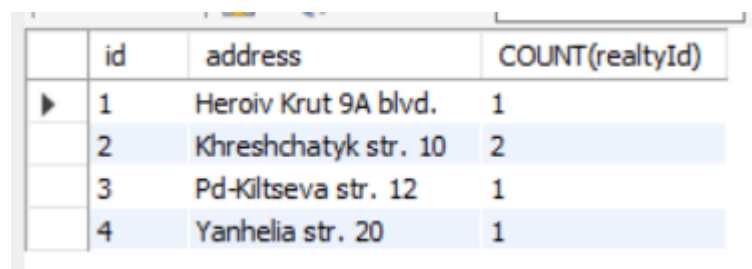
7.1.9. Кількість оглядів нерухомості.

Запит, що показує, скільки разів була оглянута кожна нерухомість.

```

SELECT realties.id, address, COUNT(reactyId)
FROM realties, surveys
WHERE realties.id=reactyId
GROUP BY reactyId;

```



	id	address	COUNT(reactyId)
▶	1	Heroiv Krut 9A blvd.	1
	2	Khreshchatyk str. 10	2
	3	Pd-Kiltseva str. 12	1
	4	Yanhelia str. 20	1

Рисунок 7.9 – Кількість оглядів нерухомості.

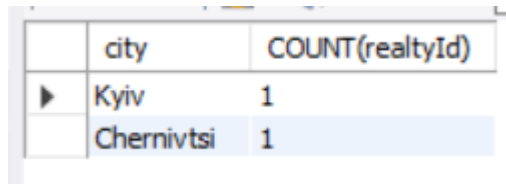
7.1.10. Продана нерухомість за містами.

Запит, що показує, де і скільки було продано нерухомості.


```

SELECT city, COUNT(realtyId)
FROM deals,realties
WHERE realtyId=realties.id
GROUP BY city;

```



	city	COUNT(realtyId)
►	Kyiv	1
	Chernivtsi	1

Рисунок 7.10 – Продана нерухомість за містами.

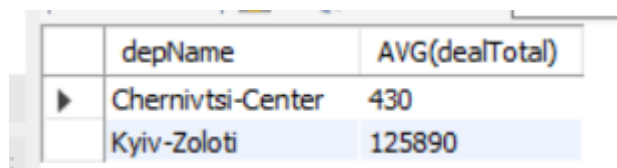
7.1.11. Середня виручка відділу.

Запит, що показує, скільки в середньому було виручки від продажів працівників за їх відділом.

```

SELECT depName,AVG(dealTotal)
FROM deals,departments,realties,workers
WHERE realtyId=realties.id AND responsibleId=workers.id AND
depId=departments.id
GROUP BY depId;

```



	depName	AVG(dealTotal)
►	Chernivtsi-Center	430
	Kyiv-Zoloti	125890

Рисунок 7.11 – Середня виручка відділу

7.1.12. Відсоток від продажу працівниками.

Запит, що показує, скільки працівник заробив на продажі нерухомості, якщо відсоток від продажу становить 10%.

```
SELECT workers.fullName, depName, 0.1*dealTotal AS vyruchka
FROM workers,departments, deals, realties
WHERE realtyId=realties.id AND responsibleId=workers.id AND
depId=departments.id;
```

	fullName	depName	vyruchka
►	Boklach Ihor	Chernivtsi-Center	43
	Bespalova Iryna	Kyiv-Zoloti	12589

Рисунок 7.12 – Відсоток від продажу працівниками

7.1.13. Дзвінок від адміністратора.

Запит, що показує власників нерухомості, номер і адміністратора, що головує над записом нерухомості: до прикладу, власникам дарують якісь бонуси або повідомляють щось дуже термінове.

```
SELECT administrators.fullName
adminName,realtyOwners.fullName,realtyOwners.phone
FROM administrators, realtyOwners,realties
WHERE administrators.id=adminId AND realtyOwners.id=ownerId;
```

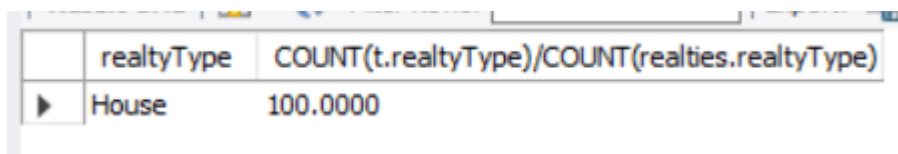
	adminName	fullName	phone
►	Zaluzhnyi Valeriy	Maydanyk Volodymyr	076543212
	Zaluzhnyi Valeriy	Skip Halyna	0765678789
	Zelenskyi Volodymyr	Kosovan Vasyl	0768900909
	Budanov Kyrylo	Kosovan Vasyl	0768900909

Рисунок 7.13 – Дзвінок від адміністратора

7.1.14. Скільки відсотків від типу продано.

Запит показує, скільки відсотків нерухомости від наявної продано за типом (Квартира\Будинок).

```
SELECT t.realtyType,COUNT(t.realtyType)/COUNT(realities.realtyType)*100
FROM realities,(SELECT realtyType,realities.id,deals.dealType
FROM realities
LEFT OUTER JOIN deals
ON realtyId=realities.id) AS t
WHERE t.dealType IS NOT NULL
GROUP BY t.realtyType;
```



	realtyType	COUNT(t.realtyType)/COUNT(realities.realtyType)
▶	House	100.0000

Рисунок 7.14 – Скільки відсотків продано

7.2. Функції й збережені процедури.

7.2.1. Кількість оглядів за роком.

Функція, яка показує, скільки оглядів було проведено заданого року. Як параметр передається необхідний рік.

```
DELIMITER //

CREATE FUNCTION surveysByYear(yearr INT)

RETURNS INT DETERMINISTIC

BEGIN

DECLARE c INT DEFAULT 0;
```

```

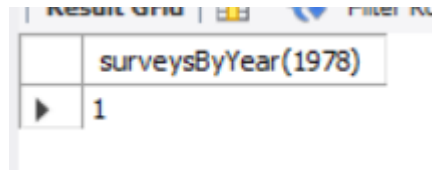
SELECT COUNT(surveys.id) INTO c
FROM surveys
WHERE YEAR(surveyDate)=yearr;
RETURN c;
END;

```

```

DELIMITER ;
SELECT surveysByYear(1978);

```



surveysByYear(1978)	
▶	1

Рисунок 7.15 – Огляди за роком проведення

7.2.2. Нерухомість за ідентифікатором відділу.

Процедура, яка показує нерухомість за заданим як параметр id відділу.

```

DELIMITER //
CREATE PROCEDURE realtyByDep(IN depId INT)
BEGIN
SELECT realties.*
FROM realties, departments
WHERE realties.city=departments.city AND departments.id=depId;
END;
DELIMITER ;

```

CALL realtyByDep(3);

	id	realtyType	dealType	price	city	address	area	rooms	responsibleId	adminId	ownerId
▶	1	Flat	Sale	23000	Chernivtsi	Heroiv Krut 9A blvd.	53.5	2	1	2	2
	3	House	Rent	400	Chernivtsi	Pd-Kiltseva str. 12	43	1	3	1	3

Рисунок 7.16 – Нерухомість за відділом

7.2.3. Різниця між ціною й угодою.

Функція, яка повертає різницю між сумою початково заданих цін на нерухомість та сумою того, що було продано.

DELIMITER //

CREATE FUNCTION bargainDiffer()

RETURNS INT

DETERMINISTIC

BEGIN

DECLARE setPrice INT DEFAULT 0;

DECLARE endPrice INT DEFAULT 0;

SELECT SUM(dealTotal) INTO endPrice

FROM deals;

SELECT SUM(T.price) INTO setPrice

FROM (SELECT price

FROM realties

INNER JOIN deals

WHERE deals.realtyId=realties.id) AS T;

```

RETURN ABS(setPrice-endPrice);

END;//

DELIMITER ;

SELECT bargainDiffer();

```

	bargainDiffer()
▶	4080

Рисунок 7.17 – Різниця між ціною й угодою

7.2.4. Ціна за квадратний метр.

Процедура, яка показує ціни нерухомості за квадратний метр.

```

DELIMITER //

CREATE PROCEDURE priceForSqM()

BEGIN

SELECT      T.id,      T.address,IF(T.keepFurniturePrice      IS      NOT
NULL,(T.price+T.keepFurniturePrice)/T.area,T.price/T.area) AS pricee
FROM (SELECT realties.id,address,price,keepFurniturePrice,area
FROM realties
LEFT OUTER JOIN amenities
ON realties.id=realtyId) AS T;

END;//

DELIMITER ;

CALL priceForSqM();

```

	id	address	pricee
►	1	Heroiv Krut 9A blvd.	571.9626168224299
	2	Khreshchatyk str. 10	1711.5384615384614
	3	Pd-Kiltseva str. 12	9.30232558139535
	4	Yanhelia str. 20	20

Рисунок 7.18 – Ціна за квадратний метр

7.3. Тригери.

7.3.1. Тригер на вставлення й перевірку дати.

Тригер, що після вставлення запису в таблицю угод, порівнює дату угоди з поточною, і якщо вона більша за поточну, то в змінну записується попередження. Інакше записується добре повідомлення.

```
DELIMITER //
```

```
CREATE FUNCTION correctDeal(dat DATE)
```

```
RETURNS VARCHAR(100)
```

```
DETERMINISTIC
```

```
BEGIN
```

```
DECLARE msg VARCHAR(100);
```

```
SET msg = IF (dat>curdate(),"The entered date is after today`s date. You may want  
to change it to the right one.",
```

```
"The entered date is valid, keep going.");
```

```
RETURN msg;
```

```
END; //
```

```
DELIMITER ;
```

```

SET @res="";

CREATE TRIGGER correctDealTrigger

AFTER INSERT

ON deals

FOR EACH ROW

SELECT correctDeal(new.dealDate) INTO @res;

SELECT @res;

INSERT INTO
deals(dealType,dealDate,dealTotal,receiptNumber,notaryId,realtyId)

VALUES ("Sale","2023-09-09",12,"qwert",1,1);

```

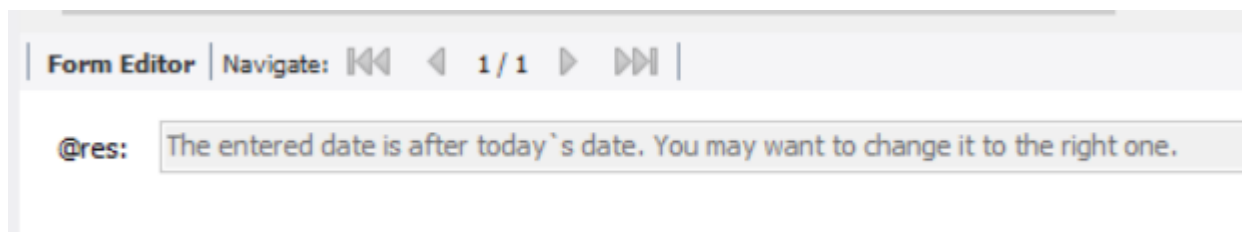


Рисунок 7.19 – Виконання триггеру на вставлення

7.3.2. Тригер на оновлення й зміну ціни у відсотках.

Для цього триггеру й триггеру на видалення створюємо додаткову таблицю. Цей тригер після оновлення в таблиці нерухомості підраховує й записує в нову таблицю, на скільки у відсотках змінилася ціна.

```

CREATE TABLE llog(id INT auto_increment PRIMARY KEY, objectId INT,
actionn VARCHAR(5),result FLOAT);

DELIMITER //

CREATE TRIGGER updatePrice

```



```

AFTER UPDATE

ON realties

FOR EACH ROW

BEGIN

DECLARE c FLOAT default 0;

SELECT

100*(NEW.price+keepFurniturePrice)/(OLD.price+keepFurniturePrice)-100

FROM amenities

WHERE OLD.id=realtyId

INTO c;

INSERT INTO llog(objectId,actionn,result)

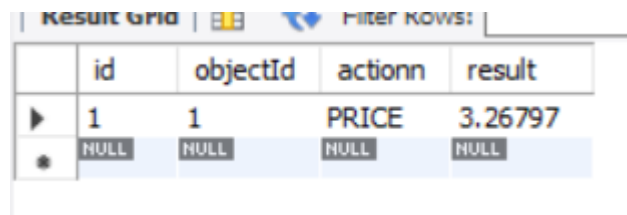
VALUES (OLD.id,"PRICE",c);

END;//

DELIMITER ;

UPDATE realties SET price=24000 WHERE id=1;

```



	id	objectId	actionn	result
▶	1	1	PRICE	3.26797
•	NULL	NULL	NULL	NULL

Рисунок 7.20 – Виконання тригера на оновлення.

7.3.3. Тригер на видалення й кількість відповідальности.

Цей тригер після видалення працівника підраховує, за яку кількість нерухомості він був відповідальний і записує запис у таблицю.

```

DELIMITER //

CREATE TRIGGER fireAWorker

BEFORE DELETE

ON workers

FOR EACH ROW

BEGIN

DECLARE c INT DEFAULT 0;

SELECT COUNT(responsibleId)

FROM realties

WHERE responsibleId=OLD.id

INTO c;

INSERT INTO llog(objectId,actionn,result)

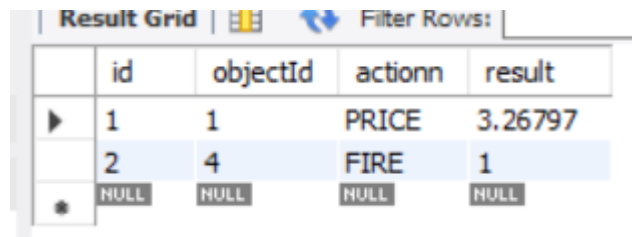
VALUES (OLD.id, "FIRE",c);

END;//

DELIMITER ;

DELETE FROM workers WHERE id=4;

```



	id	objectId	actionn	result
▶	1	1	PRICE	3.26797
	2	4	FIRE	1
★	NULL	NULL	NULL	NULL

Рисунок 7.21 – Виконання триггеру на видалення.

7.4. Представлення.

7.4.1. Розширення огляду.

Представлення, що пояснює й доповнює таблицю огляду, замінюючи значення id на значення, що зрозумілі для звичайного користувача.

```

CREATE                                                    VIEW
presentSurvey(id,surveyDate,surveyTime,result,workerName,workDep,clientName,realType,address) AS

SELECT surveys.id,

        surveyDate,

        surveyTime,

        result,

        workers.fullName,

        depName,

        clients.fullName,

        realties.realtyType,

        realties.address

FROM surveys, workers,departments,clients,realties

WHERE  realtorId=workers.id  AND  workers.depId=departments.id  AND
clientId=clients.id AND realtyId=realties.id;

```

	id	surveyDate	surveyTime	result	workerName	workDep	clientName	realType	address
►	4	2019-02-02	12:00:00	OK	Bespalova Iryna	Kyiv-Zoloti	Kinashchuk Natalia	House	Khreshchatyk str. 10
	1	2022-12-29	12:00:00	OK	Brozynskyy Oleh	Chernivtsi-Hraviton	Hlibka Yaroslava	Flat	Heroiv Krut 9A blvd.
	2	2022-12-29	14:30:00	OK	Brozynskyy Oleh	Chernivtsi-Hraviton	Hlibka Yaroslava	House	Pd-Kiltseva str. 12
	3	2023-01-05	14:45:00	NO	Bespalova Iryna	Kyiv-Zoloti	Hlibka Yaroslava	House	Khreshchatyk str. 10
	5	1978-09-09	17:23:34	NO	Bespalova Iryna	Kyiv-Zoloti	Kharyton Olena	Flat	Yanhelia str. 20

Рисунок 7.22 – Представлення для огляду

7.4.2. Представлення нерухомости.

Представлення, яке розширює таблицю нерухомости, додає кількість оглядів, ціну за квадратний метр.

```
CREATE VIEW realRealty(id,
realtyType,dealType,city,address,pricePerSqM,surveyCount,adminContact,owner
Name) AS

SELECT realties.id,

        realties.realtyType,

        realties.dealType,

        city,

        address,

        price/area,

        COUNT(realtyId),

        administrators.phone,

        realtyowners.fullName

FROM realties,surveys, administrators,realtyowners

WHERE    realties.id=realtyId    AND    adminId=administrators.id    AND

ownerId=realtyOwners.id

GROUP BY realtyId;
```

	id	realtyType	dealType	city	address	pricePerSqM	surveyCount	adminContact	ownerName
►	1	Flat	Sale	Chernivtsi	Heroiv Krut 9A blvd.	448.5981308411215	1	0897654321	Skip Halyna
	2	House	Sale	Kyiv	Khreshchatyk str. 10	1666.6666666666667	2	0897654321	Maydanyk Volodymyr
	3	House	Rent	Chernivtsi	Pd-Kiltseva str. 12	9.30232558139535	1	0891234567	Kosovan Vasyl
	4	Flat	Rent	Kyiv	Yanhelia str. 20	20	1	0893453456	Kosovan Vasyl

Рисунок 7.23 – Представлення нерухомости

7.5. Індокси.

Виконаємо простий SQL-запит. Спочатку без використання індексу.

EXPLAIN ANALYZE

SELECT departments.id,depName,fullName

FROM departments

INNER JOIN workers

ON departments.id=depId;

EXPLAIN:

```
-> Nested loop inner join (cost=2.95 rows=6) (actual time=0.037..0.047 rows=6 loops=1)
-> Filter: (workers.depId is not null) (cost=0.85 rows=6) (actual time=0.024..0.028 rows=6 loops=1)
-> Table scan on workers (cost=0.85 rows=6) (actual time=0.023..0.026 rows=6 loops=1)
-> Single-row index lookup on departments using PRIMARY (id=workers.depId) (cost=0.27 rows=1) (actual time=0.003..0.003 rows=1 loops=6)
```

Рисунок 7.24 – Час виконання запиту без індексу

CREATE INDEX depIndex ON departments(id,depName);

EXPLAIN:

```
-> Nested loop inner join (cost=3.40 rows=7) (actual time=0.021..0.030 rows=6 loops=1)
-> Filter: (workers.depId is not null) (cost=0.95 rows=7) (actual time=0.012..0.015 rows=6 loops=1)
-> Table scan on workers (cost=0.95 rows=7) (actual time=0.012..0.014 rows=6 loops=1)
-> Single-row index lookup on departments using PRIMARY (id=workers.depId) (cost=0.26 rows=1) (actual time=0.002..0.002 rows=1 loops=6)
```

Рисунок 7.25 – Час виконання запиту з індексом

Бачимо, що ефективність збільшилася у 1.77 разів. Ця різниця стає ще помітнішою на великих базах даних. Отже, індекси є важливими у роботі з базами даних.

ВИСНОВОК

Метою курсової роботи було проектування бази даних агенції нерухомості. Для виконання курсової роботи були проведені всі необхідні дослідження, що стосуються розробки. Після цього була побудована концептуальна модель. Для цього була використана мова ER-опису предметної області, яка базується на концепції, що інформаційна модель будь-якої предметної області може бути описана із застосування таких понять, як сутність, атрибут, зв'язок. Крім того, ця мова є суттєво графічною, що дає можливість наочно представляти концептуальну модель предметної області. При побудові концептуальної моделі неявно використовувалися результати теорії нормалізації, у зв'язку з цим побудована модель представлена у третій нормальній формі.

Логічне та фізичне проектування бази даних складалося з конвертації концептуальної моделі предметної області у реляційну модель даних. Після цього реляційна база даних була представлена у вигляді команд створення таблиць бази даних у мові SQL. Крім того, у мові SQL описані запити.

В даній курсовій роботі була розроблена база даних «Агенція нерухомості» в системі управління базами даних MySQL.

Таким чином, створення бази даних «Агенція нерухомості» є досить актуальним і корисним.

ПЕРЕЛІК ПОСИЛАНЬ

1. MySQL Select Statement. // MySQL [Інтернет-ресурс]. URL: <https://dev.mysql.com/doc/refman/8.0/en/select.html>
2. MySQL Trigger Syntax and Examples. // MySQL. URL: <https://dev.mysql.com/doc/refman/8.0/en/trigger-syntax.html>
3. База даних MySQL. // Promoter. URL: <https://promoter.net.ua/articles/baza-danix-mysql.html>
4. Модель «Сутність-зв'язок». // Вікіпедія. URL: https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%C2%AB%D1%81%D1%83%D1%82%D0%BD%D1%96%D1%81%D1%82%D1%8C_%E2%80%94%D0%B7%D0%B2%27%D1%8F%D0%B7%D0%BE%D0%BA%C2%BB