

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

КАФЕДРА інформатики та програмної інженерії
(повна назва кафедри)

КУРСОВА РОБОТА

з дисципліни «Бази даних»
(назва дисципліни)

на тему: База даних для ведення реєстру земельних ділянок та їх
власників з можливістю побудови топологічної карти

Студента 2 курсу групи ПІ-02
спеціальності 121 «Інженерія програмного
забезпечення»

Глуценка О.В.
(прізвище та ініціали)

Керівник ст.вик. Марченко О.І.
(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 – 2021 рік

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

Факультет Інформатики та обчислювальної техніки
(повна назва)

Кафедра Інформатики та програмної інженерії
(повна назва)

Дисципліна Бази даних

Курс 2 Група ІІІ-02 Семестр I

**ЗАВДАННЯ
НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Глуценку Олександру Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи База даних для ведення реєстру земельних ділянок та їх власників з можливістю побудови топологічної карти

керівник роботи ст.вик. Марченко О.І.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

2. Строк подання студентом роботи 24.12.2021

3. Вихідні дані до роботи Тексти програмного коду (Додаток А)

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Вступ, аналіз предметної області, постановка завдання, концептуальна модель бази даних, логічна модель бази даних, реалізація бази даних, висновки, перелік посилань

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

6. Дата видачі завдання 31.10.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання курсового проекту	Строк виконання етапів проекту	Примітка
1	Вивчення літератури	05.11.2021	
2	Аналіз предметного середовища	10.11.2021	
3	Побудова ER-діаграми	12.11.2021	
4	Побудова реляційної схеми	22.11.2021	
5	Створення бази даних	26.11.2021	
6	Створення користувачів бази даних	05.12.2021	
7	Імпорт даних з використанням засобів СУБД	10.12.2021	
8	Створення запитів до розробленої БД	17.12.2021	
9	Оптимізація роботи запитів	20.12.2021	

Студент

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

Керівник роботи

(підпис) Марченко О.І.
(прізвище та ініціали)

Зміст

ВСТУП.....	3
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	4
1.1 Сутності	4
1.2 Обмеження системи.....	6
1.3 Запити.....	6
2 ПОСТАНОВКА ЗАВДАННЯ	8
3 КОНЦЕПТУАЛЬНА МОДЕЛЬ БАЗИ ДАНИХ	9
3.1 Побудова ER-діаграми.....	9
3.2 Відношення між об'єктами бази даних.....	10
4 ЛОГІЧНА МОДЕЛЬ БАЗИ ДАНИХ	12
4.1 Схеми таблиць бази даних	12
4.2 Схеми бази даних	14
5 РЕАЛІЗАЦІЯ БАЗИ ДАНИХ.....	16
5.1 Створення таблиць.....	16
5.2 Обмеження значень полів	18
5.3 Заповнення даними	19
5.4 Створення запитів.....	21
5.5 Створення процедур	32
5.6 Оптимізація запитів за допомогою індексів.....	34
5.7 Створення ролей користувачів бази даних.....	35
ВИСНОВКИ	36
ПЕРЕЛІК ПОСИЛАНЬ.....	37
Додаток А Тексти програмного коду.....	38

ВСТУП

Ведення реєстру земельних ділянок завжди було актуальною для суспільства темою, адже питання привласнення, продажу та купівлі землі є надзвичайно важливим для держави. Оскільки цей процес достатньо складний і вимагає не тільки врахування великої кількості факторів, а й структурованого збереження інформації, щоб грамотно регулювати його необхідне створення бази даних. Саме таку буде розроблено в ході виконання курсової роботи.

Для реалізації поставленої задачі було обрано наступні інструменти:

1. Середовище JetBrains DataGrip для візуалізації даних, написання скриптів, а також для заповнення таблиць даними з файлів типу .csv. Було обрано саме це середовище, оскільки воно є дуже зручним для написання скриптів, а також підтримує більшість відомих
2. Система керування реляційними базами даних MySQL, оскільки вона є найпопулярнішою та однією з найбільш потужних серед аналогів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Сутності

Необхідно розробити базу даних для ведення реєстру земельних ділянок та їх власників. Відповідно, основною сутністю є Ділянка. Вона має наступні атрибути:

- Унікальний ключ
- Унікальний ключ власника ділянки
- Унікальний ключ типу використання ділянки
- Унікальний ключ регіону, де вона знаходиться

Власником ділянки може бути як фізична, так і юридична особа.

Юридична особа має такі атрибути:

- Унікальний ключ
- Назва установи

Фізична особа має такі атрибути:

- Унікальний ключ
- Ім'я
- Прізвище
- По батькові
- Дата народження
- Унікальний ключ юридичної установи (якщо фізична особа є її представником)

В залежності від регіону, де знаходиться земельна ділянка, вона може мати різну ціну. Тож існує сутність Регіон з наступними атрибутами:

- Унікальний ключ
- Назва
- Податок

Сутність Тип використання:

- Унікальний ключ
- Назва
- Податок

Сутність Натуральні ресурси:

- Унікальний ключ
- Назва
- Податок

Для більш зручного обчислення площі ділянки, а також її зображення на топографічній карті, було вирішено розбивати всю її територію на прямокутники. Тож було створено сутність Сегменти ділянки з наступними атрибутами:

- Унікальний ключ ділянки
- Довгота лівого верхнього кута
- Широта лівого верхнього кута
- Довгота правого нижнього кута
- Широта правого нижнього кута
- Унікальний ключ натурального ресурсу

Для покупця важливо знати, які блага містить ділянка. Тому існує сутність Комунікації, що має наступні атрибути:

- Електроенергія
- Водопровід
- Каналізація
- Унікальний ключ ділянки

Для реєстрації купівлі-продажу ділянки було створено окрему сутність Акт. Вона містить такі атрибути:

- Унікальний ключ ділянки
- Унікальний ключ продавця
- Унікальний ключ покупця

- Унікальний ключ реєстратора
- Дата здійснення операції

Для збереження даних реєстратора також існує окрема сутність
Реєстратор:

- Унікальний ключ реєстратора
- Ім'я реєстратора

1.2 Обмеження системи

- 1) Вік клієнта не має бути меншим за 18 років.
- 2) Довгота лівого верхнього кута має бути меншою за довготу правого нижнього кута.
- 3) Широта лівого верхнього кута має бути меншою за широту правого нижнього кута.
- 4) Податок має бути невід'ємним числом.
- 5) Усі ключі мають бути унікальними.
- 6) Дата здійснення акту купівлі-продажу не має бути більшою за поточну дату.
- 7) Довгота не може бути від'ємною.

1.3 Запити

Для реалізації функціональної частини БД необхідно розробити наступні запити на мові SQL:

1. Відображення імені представника компанії та назви компанії.
2. Відображення кількості актів, здійснених юридичною особою.
3. Відображення площі ділянки.
4. Відображення осіб з найбільшою площею ділянок.
5. Відображення загальної площі ділянок різного типу.
6. Відображення суми вартості ділянок фізичних осіб.
7. Відображення площі ділянок в різних регіонах.
8. Відображення кількості ділянок з електропостачанням в різних регіонах.

9. Відображення актів, що відбулись в період з 01.01.2001 по 01.01.2020.
10. Відображення загальної площі ділянок якими володіє компанія.
11. Відображення натуральних ресурсів якими володіє особа.
12. Відображення ділянок з власниками старше 70 років.
13. Відображення ділянок без комунікацій.
14. Відображення регіону, де проживають особи.
15. Відображення актів, що відбулись з ділянкою з заданим id.
16. Відображення актів здійснених заданим реєстратором.
17. Відображення сумарної площі натуральних ресурсів на всіх ділянках.
18. Відображення реєстраторів, що співпрацювали з юридичною особою.
19. Відображення кількості ділянок з проведеною каналізацією в регіоні.
20. Відображення кількості ділянок з проведеним водопостачанням в регіоні.

2 ПОСТАНОВКА ЗАВДАННЯ

Необхідно спроектувати базу даних для ведення реєстру земельних ділянок. Для цього потрібно виконати такі кроки:

- побудувати ER-діаграму;
- побудувати схему БД;
- створити таблиці та зв'язки між ними;
- заповнити БД записами;
- розробити скрипти за запити на мові SQL;
- оптимізувати роботу скриптів.

Також необхідними умовами є використання тригерів, генераторів та представлень.

Кількість таблиць, які використовуються в кожному з запитів, має бути більше одного, а загальна кількість сутностей не меншою за 10.

Отже, було визначено основні кроки та вимоги до проектування бази даних для ведення реєстру земельних ділянок.

3 КОНЦЕПТУАЛЬНА МОДЕЛЬ БАЗИ ДАНИХ

3.1 Побудова ER-діаграми

Створена база даних має 10 сутностей. Їх назви та призначення наведені в таблиці 3.1.

Таблиця 3.1 – Назви сутностей та їх призначення

Назва	Назва в БД	Призначення
Ділянка	land_plot	Інформує про тип ділянки, її власника та регіон, де вона розміщена
Сегменти ділянки	land_plot_chunks	Зберігає прямокутники, на які умовно поділена ділянка, їх координати на місцевості та натуральні ресурси всередині цих ділянок
Натуральні ресурси	natural_resource	Зберігає назву натурального ресурсу та податок на нього
Регіон	region	Інформує про назву регіону, в якому міститься ділянок, та податок в залежності від нього
Тип використання	usage_type	Інформує про тип використання ділянки та податок в залежності від нього
Комунікації	communications	Зберігає дані про зручності, які містить ділянка: наявність електроенергії, водопроводу та каналізації
Акт	action	Зберігає дані про акт купівлі-продажу, а саме унікальні ключі ділянки, продавця, покупця та реєстратора, а також дату здійснення акту

Продовження таблиці 3.1

Назва	Назва в БД	Призначення
Фізична особа	physical_person	Зберігає дані про фізичну особу – власника ділянки або покупця. Містить такі дані, як ПІБ, дата народження та ключ підприємства, якщо людина є його представником
Юридична особа	juridical_person	Зберігає унікальний ключ юридичної особи та назву
Реєстратор	registrator	Зберігає унікальний ключ реєстратора та його ім'я

3.2 Відношення між об'єктами бази даних

Зв'язки між сутностями наведені в таблиці 3.2.

Таблиця 3.2 – Зв'язки між сутностями

Перша сутність	Друга сутність	Тип зв'язку
LandPlot	LandPlotChunks	Один до багатьох
PhysicalPerson	LandPlot	Один до багатьох
JuridicalPerson	PhysicalPerson	Один до одного
PhysicalPerson	Actions	Один до багатьох
Registrator	Actions	Один до багатьох
LandPlot	Actions	Один до багатьох
UsageType	LandPlot	Один до одного
Region	LandPlot	Один до одного
Communications	LandPlot	Один до одного
NaturalResource	LandPlotChunks	Один до багатьох

UML-діаграму наведено на рис 3.1.

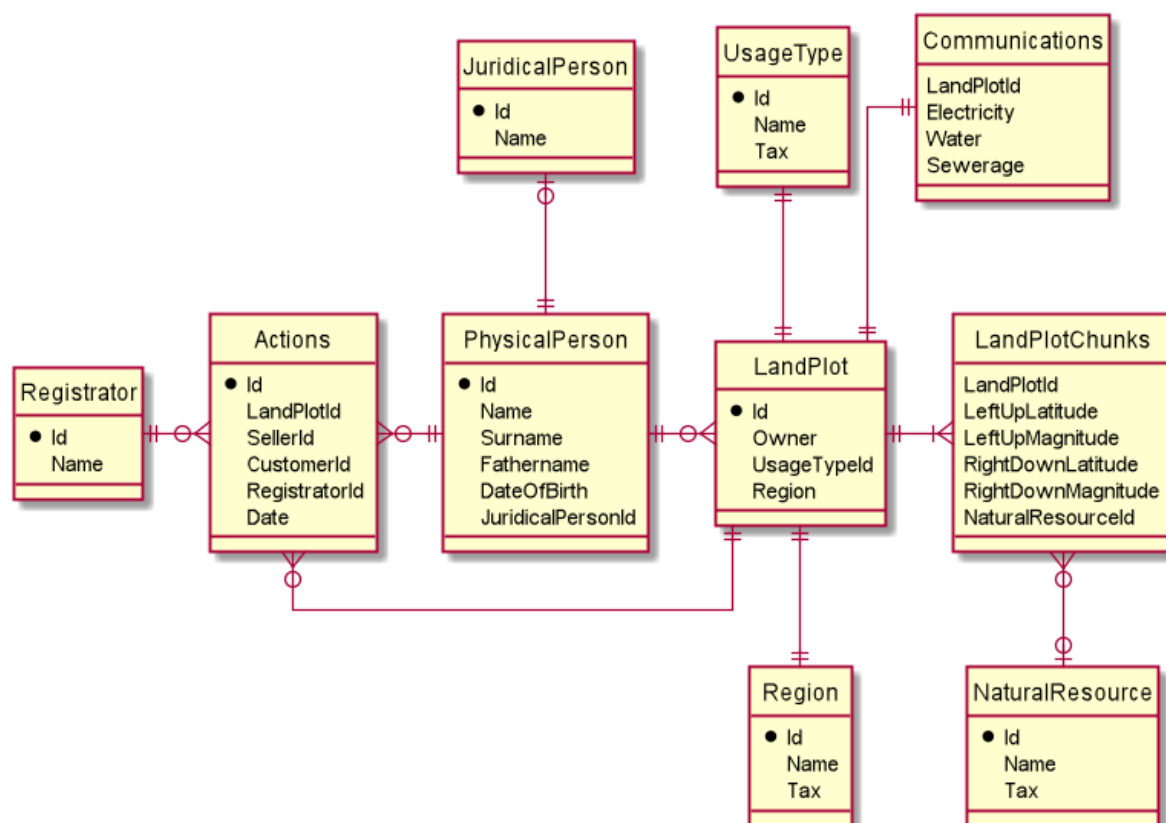


Рис. 3.1 – UML-діаграма

4 ЛОГІЧНА МОДЕЛЬ БАЗИ ДАНИХ

4.1 Схеми таблиць бази даних

Схеми таблиць зображено в таблицях 4.1 – 4.10

Таблиця 4.1 – Типи даних атрибутів таблиці land_plot

Назва атрибуту	Тип даних
(PK) Id	int
(FK) OwnerId	int
(FK) UsageTypeId	int
(FK) RegionId	int

Таблиця 4.2 – Типи даних атрибутів таблиці region

Назва атрибуту	Тип даних
(PK) Id	int
Name	varchar(50)
Tax	float

Таблиця 4.3 – Типи даних атрибутів таблиці usage_type

Назва атрибуту	Тип даних
(PK) Id	int
Name	varchar(50)
Tax	float

Таблиця 4.4 – Типи даних атрибутів таблиці natural_resource

Назва атрибуту	Тип даних
(PK) Id	int
Name	varchar(50)
Tax	float

Таблиця 4.5 – Типи даних атрибутів таблиці land_plot_chunks

Назва атрибуту	Тип даних
(FK) LandPlotId	int
LeftUpLatitude	float
LeftUpMagnitude	float
RightDownLatitude	float
RightDownMagnitude	float
NaturalResourcesId	int

Таблиця 4.6 – Типи даних атрибутів таблиці communications

Назва атрибуту	Тип даних
(FK) LandPlotId	int
Electricity	tinyint(1)
Water	tinyint(1)
Sewerage	tinyint(1)

Таблиця 4.7 – Типи даних атрибутів таблиці action

Назва атрибуту	Тип даних
(PK) Id	int
(FK) LandPlotId	int
(FK) SellerId	int
(FK) CustomerId	int
(FK) RegistrarorId	int
Date	date

Таблиця 4.8 – Типи даних атрибутів таблиці physical_person

Назва атрибуту	Тип даних
(PK) Id	int
Name	varchar(50)
Surname	varchar(50)
Fathername	varchar(50)
(FK) JuridicalPersonId	int
DateOfBirth	date

Таблиця 4.9 – Типи даних атрибутів таблиці juridical_person

Назва атрибуту	Тип даних
(PK) Id	int
Name	varchar(100)

Таблиця 4.10 – Типи даних атрибутів таблиці registrator

Назва атрибуту	Тип даних
(PK) Id	int
Name	varchar(30)

4.2 Схеми бази даних

Після створення схеми таблиць та ER-моделі було спроектовано схему бази даних (рис. 4.1).

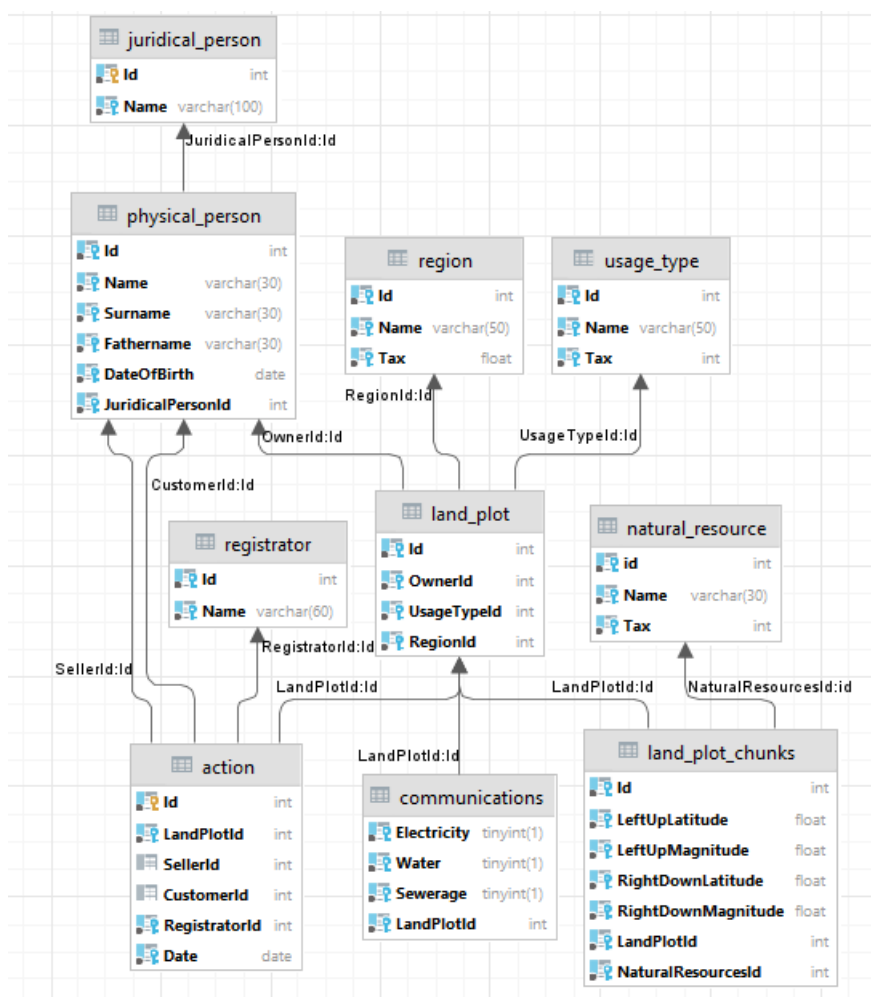


Рис 4.1 – Схема бази даних

5 РЕАЛІЗАЦІЯ БАЗИ ДАНИХ

5.1 Створення таблиць

Для реалізації бази даних необхідно в першу чергу створити таблиці. Для цього необхідно написати скрипт, вказавши назву таблиці, її поля та їх типи. Також можна одразу вказати первинний ключ таблиці та деякі обмеження. Наприклад, на рис. 5.1 створено таблицю Сегменти ділянки

```
create table if not exists land_plot_chunks
(
    Id                int auto_increment
    primary key,
    LeftUpLatitude    float null,
    LeftUpMagnitude   float null,
    RightDownLatitude float null,
    RightDownMagnitude float null,
    LandPlotId        int null,
    NaturalResourcesId int null,
    constraint check_latitude check ( LeftUpLatitude < RightDownLatitude ),
    constraint check_magnitude check ( LeftUpMagnitude < RightDownMagnitude)
);
```

Рис 5.1 – Створення таблиці

Після створення пов'язаних таблиць можна розширити скрипти, пов'язавши їх поля. Для цього необхідно додати обмеження foreign key та вказати тип оновлення і видалення полів. Наприклад, на рис. 5.2 зображено створення таблиці Сегменти ділянки з полями, пов'язаними з таблицями Ділянка та Натуральні ресурси.

5.2 Обмеження значень полів

За допомогою конструкції `constraint check` було створено наступні обмеження:

1. Таблиця `natural_resource`

```
constraint tax_check check (Tax>=0),
```

Рис 5.4 – Податок має бути невід’ємним числом

2. Таблиця `region`

```
constraint tax_check check (Tax>=0)
```

Рис 5.5 – Податок має бути невід’ємним числом

3. Таблиця `usage_type`

```
constraint tax_check check (Tax>=0)
```

Рис 5.6 – Податок має бути невід’ємним числом

4, 5. Таблиця `land_plot_chunks`

```
constraint check_latitude check (LeftUpLatitude < RightDownLatitude),
constraint check_magnitude check (LeftUpMagnitude < RightDownMagnitude)
```

Рис 5.7 – Ліві координати мають бути меншими за відповідні праві

Оскільки використання `check` не допускає використання функції `currdate()`, було створено наступні тригери:

1. Особа, що продає або купує землю, має бути повнолітньою:

```
create trigger 'physical_person_before_insert' before insert on 'physical_person' for each row
begin
    if datediff(currdate(), NEW.DateOfBirth)/365 <=18 then
        signal sqlstate '45000'
        set MESSAGE_TEXT = 'Client is under 18 years';
    end if;
end;
```

Рис 5.8 – Тригер для перевірки віку особи

2. Дата здійснення акту купівлі-продажу не може бути більшою за поточну:

```
create trigger 'action_before_insert' before insert on 'action' for each row
begin
    if New.Date > CURDATE() then
        signal sqlstate '45000'
        set MESSAGE_TEXT = 'Date is in future';
    end if;
end;
```

Рис 5.9 – Тригер для перевірки дати здійснення акту купівлі-продажу

5.3 Заповнення даними

Для заповнення таблиць даними доцільно використовувати вбудований засіб DataGrip «Імпорт з файлу». Для цього необхідно натиснути ПКМ по таблиці, яку необхідно заповнити, та вибрати зі списку “Import Data from File” (рис. 5.10)

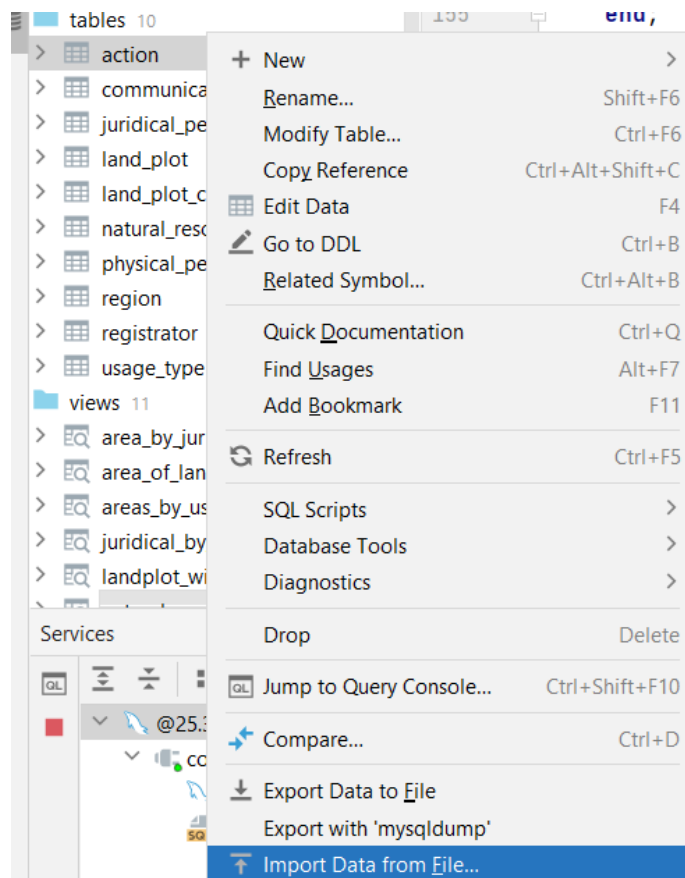


Рис 5.10 – Вибір команди “Import Data from File”

Обравши файл з якого здійснюється імпорт (рис. 5.11), необхідно призначити колонки таблиці відповідним колонкам файлу на натиснути “Import” (рис. 5.12)

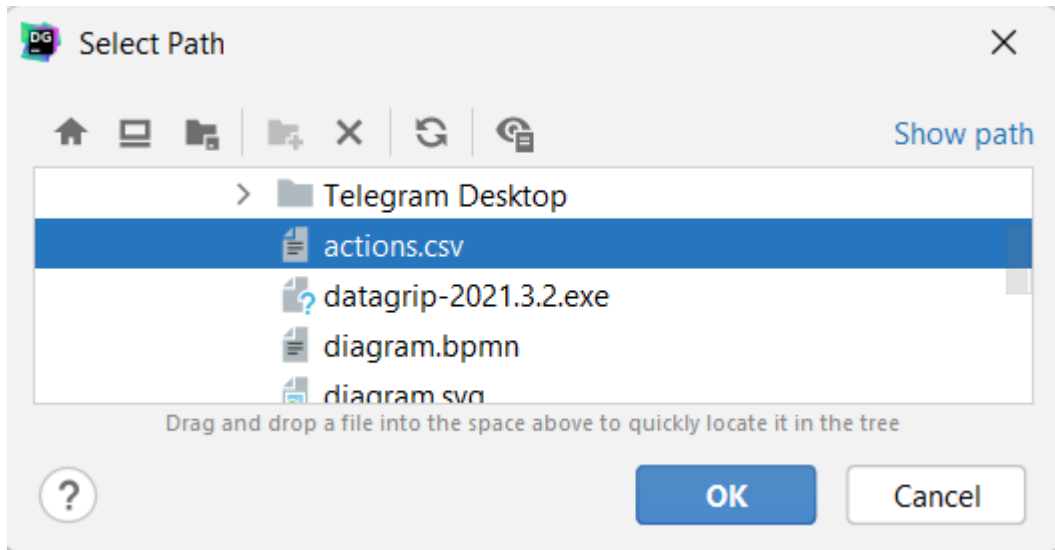


Рис 5.11 – Вибір файлу для імпорту

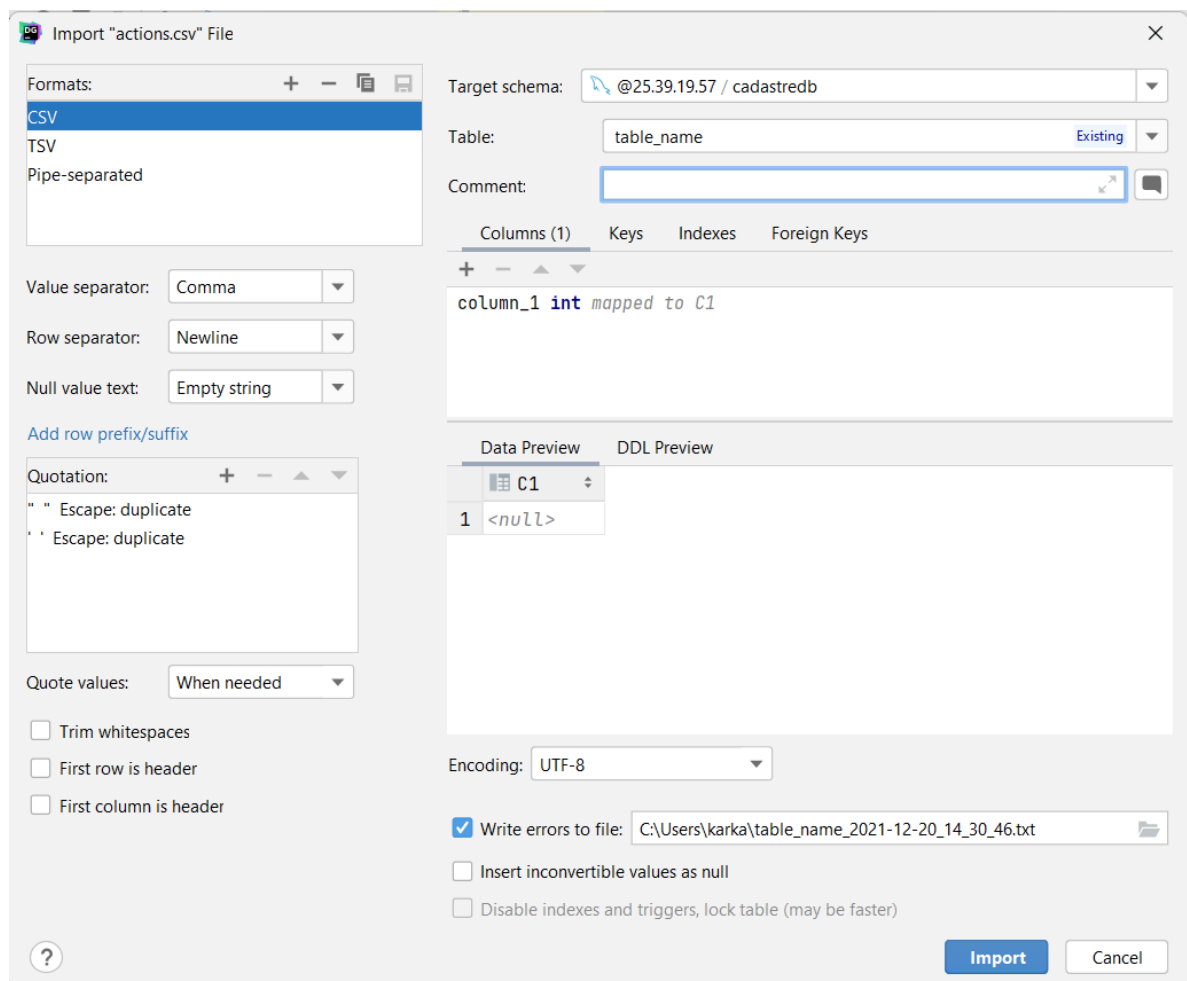


Рис 5.12 – Призначення колонкам таблиці відповідних колонок файлу

Таким чином було створено 10 таблиць розроблюваної бази даних. Файли .csv генерувались програмно засобами мови С#.

5.4 Створення запитів

Для забезпечення роботи потрібно реалізувати запити що відповідають вимогам у розділі «Аналіз предметної області». Деякі з них містять процедури, що детальніше описані в пункті 5.5. Список запитів наступний:

1. Відображення імені представника компанії та назви компанії

```
create view juridical_by_representative as
select pp.Surname, pp.Name, pp.Fathername ,jp.Name as CompanyName
from physical_person as pp join juridical_person as jp
on pp.JuridicalPersonId = jp.Id
```

Рис 5.13 – Скрипт для відображення імені представника компанії та назви компанії

	Surname	Name	Fathername	CompanyName
1	Йоне	Соломія	Матвіївна	ТОВ "Мета"
2	Дійкстра	Дарина	Артемівна	ТОВ "Леза"
3	Німкевич	Соломія	Давидівна	ПРАТ "Темне Братство"
4	Порошенко	Денис	Маркович	ТОВ "Моргтонг"
5	Немченко	Соломія	Юріївна	ПРАТ "Колегія Вінтерхолда"

Рис 5.14 – Результат виконання скрипту

2. Відображення кількості актів, здійснених юридичною особою

```
create view registrators_by_amount_of_actions as
select rg.Name, COUNT(*) as Actions
from registrar as rg join action a
on rg.Id = a.RegistratorId
group by rg.Name
```

Рис 5.15 – Скрипт для відображення кількості актів, здійснених юридичною особою

	Name	Actions
1	Вайтранська Реєстраційна Комісія	3
2	Маркартська Реєстраційна Комісія	3
3	Соліт'юдська Центральна Реєстраційна Комісія	2
4	Виндхельмська Реєстраційна Комісія	3

Рис 5.16 – Результат виконання скрипту

3. Відображення площі ділянки

```

create view area_of_landplot as
select lp.Id, sum((RightDownLatitude - LeftUpLatitude)*(RightDownMagnitude-LeftUpMagnitude)) as Area
from land_plot_chunks
join land_plot lp 1..n<->1: on lp.Id = land_plot_chunks.LandPlotId
group by lp.Id

```

Рис 5.17 – Скрипт для відображення площі ділянки

	Id	Area
1	21	0.004091815004358068
2	22	0.0025418775621801615
3	23	0.002304062742041424
4	24	0.002849893950042315
5	25	0.01019233837723732
6	26	0.002450138854328543
7	27	0.0016021511546568945
8	28	0.0002488637692295015
9	29	0.0009800982661545277
10	30	0.0012089763768017292
11	31	0.017266157898120582
12	32	0.0031448597728740424
13	33	0.0006159526674309745
14	34	0.0019082155195064843

Рис 5.18 – Результат виконання скрипта

4. Відображення людей з найбільшою площею ділянок

```

create view Top5_landplot_owners_by_area as
select Surname, Name, Fathename, sum(Area) as AreaSum from physical_person
join land_plot lp 1<->1..n: on physical_person.Id = lp.OwnerId
join area_of_landplot on lp.Id = area_of_landplot.Id
group by (physical_person.Id)
order by AreaSum desc
LIMIT 5

```

Рис 5.19 – Скрипт для відображення людей з найбільшою площею ділянок

	Surname	Name	Fathername	AreaSum
1	Немченко	Святослав	Владиславович	0.017266157898120582
2	Гончар	Олексій	Олегович	0.012100553896743804
3	Чичка	Максим	Святославович	0.004091815004358068
4	Ковш	Христина	Максимівна	0.0040522900089854375
5	Порошенко	Дарина	Святославівна	0.0031448597728740424

Рис 5.20 – Результат виконання скрипта

5. Відображення загальної площі ділянок різного типу

```
create view areas_by_usage_type as
select usage_type.Name, sum(area_of_landplot.Area) as TotalArea from usage_type
join land_plot lp 1<->1..n: on usage_type.Id = lp.UsageTypeId
join area_of_landplot on area_of_landplot.Id = lp.Id
group by usage_type.Id
order by TotalArea desc
```

Рис 5.21 – Скрипт для відображення загальної площі ділянок різного типу

	Name	TotalArea
1	Вирощування	0.03465641607181169
2	Випасання худоби	0.009540737519273534
3	Забудова	0.007208248323877342

Рис 5.22 – Результат виконання скрипта

6. Відображення суми вартості ділянок фізичних осіб

```
create view taxes_by_person as
select physical_person.Surname,
       physical_person.Name,
       physical_person.Fathername,
       IFNULL(sum(nr.Tax) * Area * 1000, 0) + sum(r.Tax) + sum(ut.Tax) * 1000 * Area as Tax
from physical_person
join land_plot lp 1<->1..n: on physical_person.Id = lp.OwnerId
join area_of_landplot on area_of_landplot.Id = lp.Id
join region r 1..n<->1: on lp.RegionId = r.Id
join land_plot_chunks lpc 1<->1..n: on lp.Id = lpc.LandPlotId
left join natural_resource nr 1..n<->0..1: on lpc.NaturalResourcesId = nr.id
join usage_type ut 1..n<->1: on lp.UsageTypeId = ut.Id
group by physical_person.Id
```

Рис 5.23 – Скрипт для відображення суми вартості ділянок фізичних осіб

	Surname	Name	Fathername	Tax
1	Чичка	Максим	Святославович	80.37722506537102
2	Коші	Ангеліна	Артемівна	206.76391910389066
3	Німкевич	Соломія	Давидівна	164.72345081227832
4	Порошенко	Денис	Маркович	72.9978790008463
5	Гончар	Олексій	Олегович	834.6177166700363
6	Коші	Христина	Максимівна	224.76041407464072
7	Глуценко	Злата	Святославівна	42.50245665386319
8	Дійкстра	Дарина	Артемівна	61.31417318806052
9	Немченко	Святослав	Владиславович	1069.969473887235
10	Порошенко	Дарина	Святославівна	65.17289659311064

Рис 5.24 – Результат виконання скрипта

7. Відображення площі ділянок в різних регіонах

```

create view regions_by_area as
select region.Name, sum(Area) as TotalArea from region
join land_plot lp 1<->1..n: on region.Id = lp.RegionId
join area_of_landplot on area_of_landplot.Id = lp.Id
group by region.Id
order by TotalArea desc

```

Рис 5.25 – Скрипт для відображення площі ділянок в різних регіонах

	Name	TotalArea
1	Хаафінгар	0.017266157898120582
2	Вінтерхолд	0.012642477231565863
3	Вайтран	0.004758109469548799
4	Фолкріт	0.004740910706459545
5	Данстар	0.004091815004358068
6	Маркарт	0.0028111275314586237
7	Рифт	0.0025418775621801615
8	Х'ялмарк	0.002304062742041424
9	Істмарк	0.0002488637692295015

Рис 5.26 – Результат виконання скрипта

8. Відображення кількості ділянок з електропостачанням в різних регіонах

```

create view regions_by_electrified_landplots as
select Name, count(lp.Id) as Electrified from region
join land_plot lp 1<->1..n: on region.Id = lp.RegionId
join communications c 1<->1: on lp.Id = c.LandPlotId
where Electricity = 1
group by region.Id

```

Рис 5.27 – Скрипт для відображення кількості ділянок з електропостачанням в різних регіонах

	Name	Electrified
1	Х'ялмарк	1
2	Вайтран	2
3	Вінтерхолд	1
4	Істмарк	1
5	Хаафінгар	1
6	Данстар	1

Рис 5.28 – Результат виконання скрипта

9. Відображення актів, що відбулись в період з 01.01.2001 по 01.01.2020

```

drop procedure if exists GetActionsBetween;
create procedure GetActionsBetween(start date, end date)
begin
select (CONCAT(seller.Name, ' ', seller.Surname)) as Seller,
       (CONCAT(customer.Name, ' ', customer.Surname)) as Customer,
       a.Date
from action a
join physical_person customer 1..n<->1: on customer.Id = a.CustomerId
join physical_person seller 1..n<->1: on seller.Id = a.SellerId
where a.Date > start and a.Date < end;

end;
call GetActionsBetween( start: '2001-1-1', end: '2020-1-1')

```

Рис 5.29 – Скрипт для відображення актів, що відбулись в період з 01.01.2001 по 01.01.2020

	Seller	Customer	Date
1	Дмитро Дірихле	Христина Коші	2001-07-19
2	Соломія Капшук	Ангеліна Коші	2005-04-27
3	Олексій Гончар	Дарина Порошенко	2009-08-18
4	Денис Порошенко	Злата Глущенко	2003-07-05
5	Соломія Мирончук	Соломія Німкевич	2012-02-22
6	Роман Левенко	Святослав Немченко	2001-02-24
7	Яна Мирончук	Соломія Німкевич	2004-05-23

Рис 5.30 – Результат виконання скрипта

10. Відображення загальної площі ділянок якими володіє компанія

```

create view area_by_juridical as
select juridical_person.Name, sum(Area) as TotalArea from juridical_person
join physical_person pp 1<->1: on juridical_person.Id = pp.JuridicalPersonId
join land_plot lp 1<->1..n: on pp.Id = lp.OwnerId
join area_of_landplot on area_of_landplot.Id = lp.Id
group by juridical_person.Id
order by TotalArea desc

```

Рис 5.31 – Скрипт для відображення загальної площі ділянок якими володіє компанія

	Name	TotalArea
1	ПРАТ "Темне Братство"	0.0029200154094723985
2	ТОВ "Моргтонг"	0.002849893950042315
3	ТОВ "Леза"	0.0012089763768017292

Рис 5.32 – Результат виконання скрипта

11. Відображення натуральних ресурсів якими володіє особа

```

create view natural_resource_by_person as
select physical_person.Surname, physical_person.Name as Name,
       physical_person.Fathername, nr.Name as NaturalResource
from physical_person
       join land_plot lp 1<->1..n: on physical_person.Id = lp.OwnerId
       join land_plot_chunks lpc 1<->1..n: on lp.Id = lpc.LandPlotId
       join natural_resource nr 1..n<->1: on lpc.NaturalResourcesId = nr.id

```

Рис 5.33 – Скрипт для відображення натуральних ресурсів якими володіє особа

	Surname	Name	Fathername	NaturalResource
1	Коші	Ангеліна	Артемівна	Ліс
2	Гончар	Олексій	Олегович	Геотермальні джерела
3	Коші	Христина	Максимівна	Корисні копалини
4	Коші	Ангеліна	Артемівна	Геотермальні джерела
5	Дійкстра	Дарина	Артемівна	Ліс
6	Немченко	Святослав	Владиславович	Ліс
7	Німкевич	Соломія	Давидівна	Водойма

Рис 5.34 – Результат виконання скрипта

12. Відображення ділянок з власниками старше 70 років

```

drop procedure if exists GetLandplotWithOwnerOlder;
create procedure GetLandplotWithOwnerOlder(age int)
begin
    select land_plot.Id, pp.Surname, pp.Name, pp.DateOfBirth from land_plot
    join physical_person pp 1..n<->1: on pp.Id = land_plot.OwnerId
    where DATEDIFF(CURDATE(), pp.DateOfBirth)/365 > age;
end;
call GetLandplotWithOwnerOlder( age: 70);

```

Рис 5.35 – Скрипт для відображення ділянок з власниками старше 70 років

	Id	Surname	Name	DateOfBirth
1	23	Німкевич	Соломія	1949-08-08
2	33	Німкевич	Соломія	1949-08-08
3	30	Дійкстра	Дарина	1948-03-21
4	29	Глущенко	Злата	1951-02-23

Рис 5.36 – Результат виконання скрипта

13. Відображення ділянок без комунікацій

```

create view landplot_without_communications as
select land_plot.Id from land_plot
join communications c 1<->1: on land_plot.Id = c.LandPlotId
where c.Electricity = 0 and c.Water = 0 and c.Sewerage = 0

```

Рис 5.37 – Скрипт для відображення ділянок без комунікацій

	Id
1	21
2	26
3	27
4	29
5	30

Рис 5.38 – Результат виконання скрипта

14. Відображення регіону, де проживають особи

```

create view person_by_region as
select pp.Surname, pp.Name as Name, pp.Fathername, r.Name as Region
from physical_person pp
    join land_plot lp 1<->1..n: on pp.Id = lp.OwnerId
    join region r 1..n<->1: on lp.RegionId = r.Id

```

Рис 5.39 – Скрипт для відображення регіону, де проживають особи

	Surname	Name	Fathername	Region
1	Порошенко	Денис	Маркович	Вайтран
2	Гончар	Олексій	Олегович	Вайтран
3	Гончар	Олексій	Олегович	Вінтерхолд
4	Коші	Христина	Максимівна	Вінтерхолд
5	Чичка	Максим	Святославович	Данстар
6	Порошенко	Дарина	Святославівна	Данстар
7	Коші	Ангеліна	Артемівна	Істмарк
8	Коші	Христина	Максимівна	Маркарт
9	Дійкстра	Дарина	Артемівна	Маркарт
10	Коші	Ангеліна	Артемівна	Рифт
11	Глуценко	Злата	Святославівна	Фолкріт
12	Порошенко	Дарина	Святославівна	Фолкріт
13	Німкевич	Соломія	Давидівна	Фолкріт
14	Німкевич	Соломія	Давидівна	Х'ялмарк
15	Немченко	Святослав	Владиславович	Хаафінгар

Рис 5.40 – Результат виконання скрипта

15. Відображення актів, що відбулись з ділянкою з заданим id

```

drop procedure if exists SellingHistoryById;
create procedure SellingHistoryById(SearchId int)
begin
    select (CONCAT(seller.Name, ' ', seller.Surname)) as Seller,
           (CONCAT(customer.Name, ' ', customer.Surname)) as Customer,
           a.Date
    from action a
        join physical_person customer 1..n<->1: on customer.Id = a.CustomerId
        join physical_person seller 1..n<->1: on seller.Id = a.SellerId
    where LandPlotId = SearchId
    order by Date;
end;
call SellingHistoryById( SearchId: 31)

```

Рис 5.41 – Скрипт для відображення актів, що відбулись з ділянкою з заданим id




	 Seller	 Customer	 Date
1	Соломія Німкевич	Роман Левенко	1996-01-02
2	Святослав Немченко	Соломія Німкевич	1998-01-23
3	Роман Левенко	Святослав Немченко	2001-02-24

Рис 5.42 – Результат виконання скрипта

16. Відображення актів здійснених заданим реєстратором

```

drop procedure if exists ActionsByRegistrator;
create procedure ActionsByRegistrator(SearchId int)
begin
    select (CONCAT(seller.Name, ' ', seller.Surname)) as Seller,
           (CONCAT(customer.Name, ' ', customer.Surname)) as Customer,
           Date,
           r.Name
    from action a
        join physical_person customer 1..n<->1: on customer.Id = a.CustomerId
        join physical_person seller 1..n<->1: on seller.Id = a.SellerId
        join registrator r 1..n<->1: on a.RegistratorId = r.Id
    where RegistratorId = SearchId
    order by Date;
end;
call ActionsByRegistrator( SearchId: 2)

```

Рис 5.43 – Скрипт для відображення актів здійснених заданим реєстратором

	Seller	Customer	Date	Name
1	Соломія Німкевич	Роман Левенко	1996-01-02	Маркартська Реєстраційна Комісія
2	Роман Левенко	Святослав Немченко	2001-02-24	Маркартська Реєстраційна Комісія
3	Соломія Мирончук	Соломія Німкевич	2012-02-22	Маркартська Реєстраційна Комісія

Рис 5.44 – Результат виконання скрипта

17. Відображення сумарної площі натуральних ресурсів на всіх ділянках

```
create view areas_of_natural_resources as
select nr.Name, sum((RightDownLatitude - LeftUpLatitude) *
(RightDownMagnitude - LeftUpMagnitude)) as TotalArea
from land_plot_chunks lpc
join natural_resource nr 1..n<->1: on lpc.NaturalResourcesId = nr.id
group by nr.id
```

Рис 5.45 – Скрипт для відображення сумарної площі натуральних ресурсів на всіх ділянках

	Name	TotalArea
1	Ліс	0.012383932888042182
2	Геотермальні джерела	0.0053450329578481615
3	Корисні копалини	0.002450138854328543
4	Водойма	0.0006159526674309745

Рис 5.46 – Результат виконання скрипта

18. Відображення реєстраторів, що співпрацювали з юридичною особою

```
create view registrator_by_juridical as
select r.Name as Registrator, jp.Name as Company from action
join physical_person pp 1..n<->1: on pp.Id = action.CustomerId
join juridical_person jp 1<->1: on jp.Id = pp.JuridicalPersonId
join registrator r 1..n<->1: on action.RegistratorId = r.Id
```

Рис 5.47 – Скрипт для відображення реєстраторів, що співпрацювали з юридичною особою

	Registrar	Company
1	Виндхельмська Реєстраційна Комісія	ПРАТ "Темне Братство"
2	Маркартська Реєстраційна Комісія	ПРАТ "Темне Братство"
3	Соліт'юдська Центральна Реєстраційна Комісія	ПРАТ "Темне Братство"

Рис 5.48 – Результат виконання скрипта

19. Відображення кількості ділянок з проведеною каналізацією в регіоні

```

create view regions_by_sewerage as
select Name, count(lp.Id) as Sewerage from region
join land_plot lp 1<->1..n: on region.Id = lp.RegionId
join communications c 1<->1: on lp.Id = c.LandPlotId
where Sewerage = 1
group by region.Id

```

Рис 5.49 – Скрипт для відображення кількості ділянок з проведеною каналізацією в регіоні

	Name	Sewerage
1	Х'ялмарк	1
2	Вайтран	2
3	Вінтерхолд	1
4	Хаафінгар	1
5	Фолкріт	1
6	Данстар	1

Рис 5.50 – Результат виконання скрипта

20. Відображення кількості ділянок з проведеним водопостачанням в регіоні

```

create view regions_by_water as
select Name, count(lp.Id) as Water from region
join land_plot lp 1<->1..n: on region.Id = lp.RegionId
join communications c 1<->1: on lp.Id = c.LandPlotId
where Water = 1
group by region.Id

```

Рис 5.51 – Скрипт для відображення кількості ділянок з проведеним водопостачанням в регіоні

	Name	Water
1	Рифт	1
2	Вайтран	2
3	Вінтерхолд	1
4	Хаафінгар	1
5	Фолкріт	1

Рис 5.52 – Результат виконання скрипта

5.5 Створення процедур

Для створення деяких запитів необхідні процедури. Було спроектовано такі процедури:

1. Пошук актів, проведених реєстратором

```

create
    definer = root@`%` procedure ActionsByRegistrator(IN SearchId int)
begin
    select (CONCAT(seller.Name, ' ', seller.Surname)) as Seller,
           (CONCAT(customer.Name, ' ', customer.Surname)) as Customer,
           Date,
           r.Name
    from action a
           join physical_person customer 1..n<->1: on customer.Id = a.CustomerId
           join physical_person seller 1..n<->1: on seller.Id = a.SellerId
           join registrator r 1..n<->1: on a.RegistratorId = r.Id
    where RegistratorId = SearchId
    order by Date;
end;

```

Рис 5.53 – Процедура для пошуку актів, проведених реєстратором

	Seller	Customer	Date	Name
1	Соломія Німкевич	Роман Левенко	1996-01-02	Маркартська Реєстраційна Комісія
2	Роман Левенко	Святослав Немченко	2001-02-24	Маркартська Реєстраційна Комісія
3	Соломія Мирончук	Соломія Німкевич	2012-02-22	Маркартська Реєстраційна Комісія

Рис 5.54 – Результат виконання процедури для SearchId = 2

2. Пошук актів, що відбулись між заданими датами

```

create
    definer = root@`%` procedure GetActionsBetween(IN start date, IN end date)
begin
    select (CONCAT(seller.Name, ' ', seller.Surname)) as Seller,
           (CONCAT(customer.Name, ' ', customer.Surname)) as Customer,
           a.Date
    from action a
        join physical_person customer 1..n<->1: on customer.Id = a.CustomerId
        join physical_person seller 1..n<->1: on seller.Id = a.SellerId
    where a.Date > start and a.Date < end;

end;

```

Рис 5.55 – Процедура для пошуку актів, проведених реєстратором

	Seller	Customer	Date
1	Дмитро Дірихле	Христина Коші	2001-07-19
2	Соломія Капшук	Ангеліна Коші	2005-04-27
3	Олексій Гончар	Дарина Порошенко	2009-08-18
4	Денис Порошенко	Злата Глущенко	2003-07-05
5	Соломія Мирончук	Соломія Німкевич	2012-02-22
6	Роман Левенко	Святослав Немченко	2001-02-24
7	Яна Мирончук	Соломія Німкевич	2004-05-23

Рис 5.56 – Результат виконання процедури для start = '2001-1-1',
end = '2020-1-1'

3. Пошук ділянок, що належать людям старше деякого віку

```

create
    definer = root@`%` procedure GetLandplotWithOwnerOlder(IN age int)
begin
    select land_plot.Id, pp.Surname, pp.Name, pp.DateOfBirth from land_plot
        join physical_person pp 1..n<->1: on pp.Id = land_plot.OwnerId
    where DATEDIFF(CURDATE(), pp.DateOfBirth)/365 > age;

end;

```

Рис 5.57 – Процедура для пошуку ділянок, що належать людям старше деякого віку

	Id	Surname	Name	DateOfBirth
1	23	Німкевич	Соломія	1949-08-08
2	33	Німкевич	Соломія	1949-08-08
3	30	Дійкстра	Дарина	1948-03-21
4	29	Глуценко	Злата	1951-02-23

Рис 5.58 – Результат виконання процедури для age = 70

4. Пошук історії актів даної ділянки

```

create
  definer = root@`%` procedure SellingHistoryById(IN SearchId int)
begin
  select (CONCAT(seller.Name, ' ', seller.Surname)) as Seller,
         (CONCAT(customer.Name, ' ', customer.Surname)) as Customer,
         a.Date
  from action a
    join physical_person customer 1..n<->1: on customer.Id = a.CustomerId
    join physical_person seller 1..n<->1: on seller.Id = a.SellerId
  where LandPlotId = SearchId
  order by Date;
end;

```

Рис 5.59 – Процедура для пошуку актів, проведених реєстратором

	Seller	Customer	Date
1	Соломія Німкевич	Роман Левенко	1996-01-02
2	Святослав Немченко	Соломія Німкевич	1998-01-23
3	Роман Левенко	Святослав Немченко	2001-02-24

Рис 5.60 – Результат виконання процедури для SearchId = 31

5.6 Оптимізація запитів за допомогою індексів

Було виконано спробу оптимізувати швидкість роботи бази даних за допомогою індексації поля DateOfBirth. Було отримано наступні результати:


 12_proc_get_landplots_with_owner_older.sql 236 ms

Рис 5.61 – Швидкість до оптимізації


 12_proc_get_landplots_with_owner_older.sql 85 ms

Рис 5.62 – Швидкість після оптимізації

5.7 Створення ролей користувачів бази даних

Для того, щоб грамотно використовувати базу даних, необхідно розподілити права між користувачами відповідно до їх позиції. Це необхідно для забезпечення безпеки збереження даних. Спершу було створено наступні скрипти:

```
drop role if exists administrator;
create role administrator;
grant all privileges on *.* TO administrator;

drop role if exists manager;
create role manager;
grant DELETE, INSERT, SELECT, UPDATE on cadastredb.* TO manager;

drop role if exists user;
create role user;
grant SELECT on cadastredb.* TO user;

flush privileges;
```

Рис 5.63 – Скрипт для створення ролей

Нижче наведено перелік логінів та паролів користувачів кожної ролі, а також їх права (таблиця 5.1)

Таблиця 5.1 – Логіни та паролі користувачів кожної ролі

Логін	Пароль	Права
admin1	apassword	Усі
manager1	mpassword	Додавання, видалення, змінення
user1	upassword	Перегляд

ВИСНОВКИ

У ході даної роботи було виконано аналіз предметної області, досліджено необхідні сутності, атрибути, бізнес правила, системи для ведення обліку земельних ділянок. Було сформовано всі вимоги, та розроблено 10 сутностей та їх атрибути, зв'язки, та побудовано ER-діаграму.

Створено логічну модель БД, створено таблиці та зовнішні ключі для їх зв'язків. Було спроектовано БД, дані були імпортовані з файлів .csv, розроблено 20 запитів для їх опрацювання.

Було використано СУБД - MySQL, а для побудови діаграм PlantUML. В результаті отримано БД для ведення обліку земельних ділянок.

ПЕРЕЛІК ПОСИЛАНЬ

1. MySQL Documentation [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://dev.mysql.com/doc/> (дата звернення 10.11.2021) – Назва з екрана.
2. DataGrip Documentation [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.jetbrains.com/help/datagrip/meet-the-product.html> (дата звернення 12.11.2021) – Назва з екрана.
3. Кадастр [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу:
<https://uk.wikipedia.org/wiki/%D0%9A%D0%B0%D0%B4%D0%B0%D1%81%D1%82%D1%80> (дата звернення 10.11.2021) – Назва з екрана.
4. Державний земельний кадастр [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу:
https://uk.wikipedia.org/wiki/%D0%94%D0%B5%D1%80%D0%B6%D0%B0%D0%B2%D0%BD%D0%B8%D0%B9_%D0%B7%D0%B5%D0%BC%D0%B5%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%BA%D0%B0%D0%B4%D0%B0%D1%81%D1%82%D1%80 (дата звернення 10.11.2021) – Назва з екрана.

Додаток А Тексти програмного коду

Тексти програмного коду програмного забезпечення

Реєстр земельних ділянок та їх власників

(найменування програми (документа))

SSD

(вид носія даних)

50,8 кб, 33 арк

(обсяг програми (документа), арк.)

студента групи ІП-02 ІІ курсу

Глуценка О.В.


```

create table if not exists juridical_person
(
    Id int auto_increment
        primary key,
    Name varchar(100) not null,
    constraint JuridicalPerson_Id_uindex
        unique (Id)
);

```

```

create table if not exists natural_resource
(
    id int auto_increment
        primary key,
    Name varchar(30) null,
    Tax int default 0 not null,
    constraint tax_check check (Tax>=0),
    constraint Natural_resources_id_uindex
        unique (id)
);

```

```

create table if not exists physical_person
(
    Id int auto_increment
        primary key,
    Name varchar(30) not null,
    Surname varchar(30) not null,
    Fathurname varchar(30) not null,
    DateOfBirth date not null,
    JuridicalPersonId int null,

```

```

constraint PhysicalPerson_Id_uindex
    unique (Id),
constraint physicalperson_JuridicalPersonId_uindex
    unique (JuridicalPersonId),
constraint physicalperson_juridicalperson_Id_fk
    foreign key (JuridicalPersonId) references juridical_person (Id)
    on update cascade on delete cascade
);

```

```

create table if not exists region
(
    Id int auto_increment
        primary key,
    Name varchar(50) not null,
    Tax float default 0 not null,
    constraint Region_Id_uindex
        unique (Id),
    constraint Region_Name_uindex
        unique (Name),
    constraint tax_check check (Tax>=0)
);

```

```

create table if not exists registrator
(
    Id int auto_increment
        primary key,
    Name varchar(60) null,
    constraint Registrator_Id_uindex
        unique (Id)
);

```

);

create table if not exists usage_type

```
(
  Id int auto_increment
    primary key,
  Name varchar(50) not null,
  Tax int default 0 not null,
  constraint usage_type_Id_uindex
    unique (Id),
  constraint tax_check check (Tax>=0)
);
```

create table if not exists land_plot

```
(
  Id int auto_increment
    primary key,
  OwnerId int null,
  UsageTypeId int null,
  RegionId int null,
  constraint LandPlot_Id_uindex
    unique (Id),
  constraint land_plot_physical_person_Id_fk
    foreign key (OwnerId) references physical_person (Id)
    on update cascade,
  constraint land_plot_region_Id_fk
    foreign key (RegionId) references region (Id)
    on update cascade,
  constraint land_plot_usage_type_Id_fk
```

```

        foreign key (UsageId) references usage_type (Id)
            on update cascade on delete set null
    );

create table if not exists action
(
    Id          int auto_increment
        primary key,
    LandPlotId int not null,
    SellerId    int not null,
    CustomerId  int not null,
    RegistratorId int not null,
    Date        date not null,
    constraint Action_Id_uindex
        unique (Id),
    constraint Action_land_plot_Id_fk
        foreign key (LandPlotId) references land_plot (Id)
            on update cascade,
    constraint Action_physical_person_Id_fk
        foreign key (SellerId) references physical_person (Id)
            on update cascade,
    constraint Action_physical_person_Id_fk_2
        foreign key (CustomerId) references physical_person (Id)
            on update cascade,
    constraint action_registrator_Id_fk
        foreign key (RegistratorId) references registrator (Id)
            on update cascade
);

```

create table if not exists communications

```
(
  Electricity tinyint(1) default 0 not null,
  Water      tinyint(1) default 0 not null,
  Sewerage   tinyint(1) default 0 not null,
  LandPlotId int          null,
  constraint communications_LandPlotId_uindex
    unique (LandPlotId),
  constraint communications_land_plot_Id_fk
    foreign key (LandPlotId) references land_plot (Id)
);
```

create table if not exists land_plot_chunks

```
(
  Id          int auto_increment
    primary key,
  LeftUpLatitude float null,
  LeftUpMagnitude float null,
  RightDownLatitude float null,
  RightDownMagnitude float null,
  LandPlotId  int null,
  NaturalResourcesId int null,
  constraint land_plot_chunks_land_plot_Id_fk
    foreign key (LandPlotId) references land_plot (Id)
    on update cascade on delete cascade,
  constraint land_plot_chunks_natural_resource_id_fk
    foreign key (NaturalResourcesId) references natural_resource (id)
    on update cascade,
  constraint check_latitude check ( LeftUpLatitude < RightDownLatitude ),
```

```
constraint check_magnitude check ( LeftUpMagnitude <  
RightDownMagnitude)  
);
```

```
create trigger 'physical_person_before_insert' before insert on 'physical_person'  
for each row
```

```
begin  
  if datediff(curdate(), NEW.DateOfBirth)/365 <=18 then  
    signal sqlstate '45000'  
    set MESSAGE_TEXT = 'Client is under 18 years';  
  end if;  
end;
```

```
create trigger 'action_before_insert' before insert on 'action' for each row
```

```
begin  
  if New.Date > CURDATE() then  
    signal sqlstate '45000'  
    set MESSAGE_TEXT = 'Date is in future';  
  end if;  
end;
```