

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет геоінформаційних систем і управління територіями

Кафедра геоінформатики і фотограмметрії

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**

на тему:

Розроблення засобів формування унікальних ідентифікаторів
будівель на основі відкритого коду місцеположення OLC в
середовищі СКБД PostgreSQL

Шкута Дмитро Валерійович

Рівень плагіату – _____

Член експертної комісії з виявлення та
запобігання академічного плагіату

(ПІБ експерта _____) _____

(підпис)

Київ – 2020 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет геоінформаційних систем і управління територіями

Кафедра геоінформатики і фотограмметрії

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ проф., д.т.н. Карпінський Ю.О.

“ _____ ” _____ 2020 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**

Розроблення засобів формування унікальних ідентифікаторів будівель на основі відкритого коду місцеположення OLC в середовищі СКБД PostgreSQL

Виконала студент групи ГСТ-51
193 «Геодезія та землеустрій»
Геоінформаційні системи і технології

Шкута Д.В.

Керівник: Лященко А.А, проф., д.т.н.

Ідентичність підтверджую

Київ 2020 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: Геоінформаційних систем та управління територіями

Кафедра: Геоінформатики і фотограмметрії

Освітній рівень: «магістр за ОПШ»

Спеціальність: 193 «Геодезія та землеустрій»

Спеціалізація: Геоінформаційні системи і технології

ЗАТВЕРДЖУЮ

Декан факультету

_____доцент., к.т.н. Нестерненко О. В.

“ _____ ” _____ 2020 року

**З А В Д А Н Н Я
ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**

Шкута Дмитро Валерійович

(прізвище, ім'я, по батькові)

1. Тема роботи: Розроблення засобів формування унікальних ідентифікаторів будівель на основі відкритого коду місцеположення OLC в середовищі СКБД PostgreSQL

затверджена наказом ректора КНУБА № від « _____ » _____ 2020 року

2. Керівник роботи проф., д.т.н. Лященко Анатолій Антонович

(прізвище, ім'я та по батькові, науковий ступінь, вчене звання)

3. Строк подання студентом роботи до захисту: 04 грудня 2020 р.

4. Зміст пояснювальної записки за розділами:

Р. 1. Обґрунтування, структура, завдання та обмеження проекту

Р. 2. Методичні засади формування унікальних ідентифікаторів будівель (UBID) на основі відкритого коду місцеположення OLC.

Р. 3. Розроблення прикладної функції для формування UBID на основі відкритого коду місцеположення OLC в середовищі СКБД PostgreSQL

Р. 4. Результати обчислювального експерименту формування UBID на основі відкритого коду місцеположення OLC

5. Графічний матеріал за розділами

Р. 1. Завдання, структура та обмеження проекту; Таблиця результатів порівняльного аналізу методів унікальної ідентифікації геопросторових об'єктів.

Р. 2. Структура та схема формування відкритого коду місцеположення OLC. Схема формування UBID на основі відкритого коду місцеположення OLC. Технологічна схема формування UBID в середовищі бази геопросторових даних.

Р. 3. Схема алгоритму прикладної функції формування UBID на основі відкритого коду місцеположення OLC. Реалізація прикладної функції формування UBID на мові PL/pgSQL в середовищі СКБД PostgreSQL/PostGIS.

Р. 4. Функціональна модель дослідної бази геопросторових даних для формування UBID. Схема обчислювального експерименту дослідження результатів формування UBID в середовищі СКБД PostgreSQL/PostGIS. Приклади електронних планів міст з відображенням

UBIV. Результати геоінформаційного аналізу використання OLC коду для формування UBIV для дослідних територій.

7. Календарний план виконання роботи:

б) практична частина

Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)
Розділ 1. Обґрунтування, структура, завдання та обмеження проекту. Призначення та завдання унікальної ідентифікація будівель в інформаційних системах управління будівництвом. Огляд та порівняльний аналіз методів унікальної ідентифікації об'єктів будівництва. Обґрунтування вибору методу формування UBID на основі відкритого коду місцеположення OLC. Структура, завдання та обмеження проекту.	30.09.2020
Розділ 2. Методичні засади формування унікальних ідентифікаторів будівель (UBID) на основі відкритого коду місцеположення OLC Структура та схема формування відкритого коду місцеположення OLC. Схема формування UBID на основі відкритого коду місцеположення OLC. Аналіз реалізацій засобів формування OLC коду.	15.10.2020
Розділ 3. Розроблення алгоритму прикладної функції формування UBID на основі відкритого коду місцеположення OLC. Модель геопросторових даних. Узагальнена схема алгоритму формування UBID на основі OLC. Реалізація прикладної функції формування UBID на мові PL/pgSQL в середовищі СКБД PostgreSQL/PostGIS.	15.11.2020
Розділ 4. Результати обчислювального експерименту формування UBID на основі відкритого коду місцеположення OLC. Функціональна модель дослідної БГД для формування UBID. Результати формування UBID в середовищі СКБД PostgreSQL/PostGIS. Результати геоінформаційного аналізу UBIV для дослідних територій.	23.11.2020
Остаточне оформлення роботи	23.11.2020
Направлення роботи на рецензування, перевірку на плагіат	30.11.2020
Попередній захист роботи на кафедрі	07.12.2020

8. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірів	
		дата	підпис
Розділ 1.			
Розділ 2.			
Розділ 3.			

9. Дата видачі завдання 25 червня 2020 р.

Зав. кафедри _____ Карпінський Ю.О.
(підпис) (прізвище та ініціали)

Керівник _____ Лященко А.А.
(підпис) (прізвище та ініціали)

Студент _____ Шкута Д.В.
(підпис) (прізвище та ініціали)

РЕЗЮМЕ (summary)		Шкута Дмитро Валерійович	
<i>До атестаційної випускної роботи студента:</i>			
<i>Назва ВНЗ</i>	Київський національний університет будівництва і архітектури		
<i>Тема</i>	Розроблення засобів формування унікальних ідентифікаторів будівель на основі відкритого коду місцеположення OLC в середовищі СКБД PostgreSQL		
<i>Освітній ступінь</i>	Магістр за освітньо-професійною програмою навчання		
<i>Факультет</i>	Геоінформаційних систем та управління територіями		
<i>Кафедри</i>	Геоінформатики та фотограмметрії		
<i>Спеціальність</i>	193 Геодезія та землеустрій		
<i>Спеціалізація</i>	Геоінформаційні системи і технології		
<i>Керівник</i>	Лященко Анатолій Антонович, д.т.н, професор		
<i>Обсяг роботи</i>	<i>пояснювальна записка, стор.</i>	<i>розділів</i>	<i>креслень формату А4</i>
	94	4	30
<i>Розділ 1</i>	Визначено роль унікальних ідентифікаторів об'єктів для інтеграції даних із різних джерел, здійснено огляд методів унікальної ідентифікації геопросторових об'єктів. Обґрунтовано вибір відкритого коду місцеположення OLC для створення унікальних ідентифікаторів будівель UBID.		
<i>Розділ 2</i>	Докладно розглянуто компоненти технології відкритого коду місцеположення OLC: структура, просторове розрізнення, загальна схема та існуючі засоби формування OLC.		
<i>Розділ 3</i>	Розроблено базову SQL функцію для формування OLC коду в середовищі СКБД PostgreSQL з просторовим розширенням PostGIS. Приведено концептуальну модель бази геопросторових даних, що використовувалася для проведення експерименту, та схему алгоритму прикладної SQL функції, що використовує базову функцію для формування OLC коду для побудови унікального ідентифікатора будівель UBID		
<i>Розділ 4</i>	Викладено результати обчислювального експерименту формування UBID на основі OLC коду в середовищі СКБД PostgreSQL/PostGIS з використанням розроблених прикладних SQL функцій. В експерименті використано реальні набори геопросторових даних про будівель міст Києва та Біла Церква. Проведено порівняльний аналіз використання для формування UBID координат центроїдів контурів будівель та центроїдів охоплюючих прямокутників; виконано статистичний аналіз часу виконання розроблених функцій на різних наборах даних..		
<i>Висновки по роботі:</i>	Використання відкритого коду місцеположення об'єктів OLC для створення унікального ідентифікатора будівель UBID забезпечує єдину унікальну ідентифікацію будівель незалежно від системи та середовища виробництва різних наборів даних. Розроблені прикладні SQL функції формування UBID на основі OLC незалежать від середовища конкретних інструментальних ГІС та можуть бути використані за призначенням для формування UBID об'єктів баз геопросторових даних в середовищі універсальних СКБД.		
Ключові слова: відкритий код місцеположення, універсальний ідентифікатор будівлі, UBID, бази геопросторових даних, СКБД PostgreSQL/PostGIS			
Keywords: open location code, OLC, universal building identifier, UBID, geospatial databases, PostgreSQL / PostGIS			

Укладач: _____ / Шкута Д.В./

Керівник: _____ /Лященко А.А./

« ____ » _____ 2020

Зміст	
ВСТУП.....	8
1 ОБҐРУНТУВАННЯ, СТРУКТУРА, ЗАВДАННЯ ТА ОБМЕЖЕННЯ ПРОЕКТУ	12
1.1 Призначення та завдання унікальної ідентифікація будівель в інформаційних системах управління будівництвом	12
1.2 Огляд та порівняльний аналіз методів унікальної ідентифікації об’єктів будівництва.....	19
1.3 Обґрунтування вибору методу формування UBID на основі відкритого коду місцеположення OLC.....	27
1.4 Структура, завдання та обмеження проекту.....	30
2 МЕТОДИЧНІ ЗАСАДИ ФОРМУВАННЯ УНІКАЛЬНИХ ІДЕНТИФІКАТОРІВ БУДІВЕЛЬ (UBID) НА ОСНОВІ ВІДКРИТОГО КОДУ МІСЦЕПОЛОЖЕННЯ OLC	33
2.1 Загальна структура та просторові властивості відкритого коду місцеположення OLC	33
2.2 Схема формування OLC коду	38
2.3 Аналіз реалізацій засобів формування OLC коду.....	42
3 РОЗРОБЛЕННЯ ПРИКЛАДНОЇ ФУНКЦІЇ ДЛЯ ФОРМУВАННЯ UBID НА ОСНОВІ ВІДКРИТОГО КОДУ МІСЦЕПОЛОЖЕННЯ OLC В СЕРЕДОВИЩІ СКБД POSTGRESQL	48
3.1 Модель бази геопросторових даних для формування UBID на основі OLC.....	48
3.2 Реалізація базової SQL функції для формування OLC коду в середовищі СКБД PostgreSQL.....	50
3.3 Реалізація прикладної функції формування UBID середовищі СКБД PostgreSQL/PostGIS	59
4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ ФОРМУВАННЯ UBID НА ОСНОВІ OLC КОДУ	62

4.1 Структурно-функціональна модель дослідної БГД для формування UBID	62
4.2 Результати обчислювального експерименту формування UBID для дослідних територій.....	69
4.2.1 Інформаційні ресурси та загальна схема обчислювального експерименту	69
4.2.2 Результати формування UBID на основі OLC коду для будівель дослідних територій	72
4.3 Геоінформаційний аналіз використання OLC коду для формування UBID для будівель на дослідних територіях	78
ВИСНОВКИ.....	92
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	93

ВСТУП

В зв'язку з широким використанням ГІС постають завдання інтеграції даних з різних джерел. Одним із засобів інтеграції даних є застосування єдиних уніфікованих ідентифікаторів об'єктів в моделях геопросторових даних.

Згідно постанови Кабінету Міністрів України від 1 липня 2020 р. № 559 Про реалізацію експериментального проекту щодо запровадження першої черги Єдиної державної електронної системи у сфері будівництва [1] кожній будівлі повинен присвоюватися унікальний ідентифікатор, який призначений для ідентифікації об'єкта будівництва або закінченому будівництвом об'єкту автоматично програмними засобами Реєстру будівельної діяльності.

Для унікальної ідентифікації є кілька систем такі як унікальний універсальний ідентифікатор UUID (від англ. Universally Unique Identifier)[6], адресний реєстр, GUID (від англ. Globally Unique Identifier)[5] та низка прийомів ідентифікації на основі їх місцеположення (геолокація).

Кожна система має переваги та недоліки. Зокрема використання GUID і UUID потребує організаційних та технологічних заходів для забезпечення однакових ідентифікацій для одного і того ж об'єкту, коли ця ідентифікація здійснюється в різних системах.

Коди на геолокації не мають недоліків, вони засновані на об'єктивних властивостях координат його місцеположення.

Оскільки в наборах даних про об'єкти будуть все частіше застосовуватись геопросторові дані які містять координатні описи об'єктів то природньо за основу ідентифікації будівель вибрати один із методів формування унікального ідентифікатора на основі геолокацій.

Для реалізації методів геолокації важливо мати відповідно програмний інструментарій який підтримував унікальну ідентифікацію в середовищі баз геопросторових даних(БГД). Враховуючи що для формування БГД за кордоном так і в нашій країні широко використовуються об'єктно-реаліційна система керування базами даних(ОРСКБД) PostgreSQL з функціонально просторовим

розширенням для опрацювання даних PostGis, актуально є завдання розробки моделей засобів підтримки унікальної ідентифікації будівель з використанням методу геолокації.

Найефективнішим є метод відкритого коду місцеположення OLC (від англ. Open Location Code)[15]. Його код заснований на простому кодуванні широти та довготи, створений на підрозділеній прямокутній сітковій референційній системі.

Метою роботи є розроблення моделей та програмних засобів формування унікальних ідентифікаторів будівель на основі відкритого коду місцеположення OLC в середовищі СКБД PostgreSQL.

Об'єкт дослідження: будівлі та набори геопросторових даних з їх інформаційними моделями.

Предмет дослідження: моделі і програмні засоби формування унікальних ідентифікаторів будівель на основі відкритого коду місцеположення OLC.

Для досягнення поставленої мети в роботі вирішуються такі завдання:

- аналіз призначення та завдання унікальної ідентифікації будівель в інформаційних системах управління будівництва
- порівняльний аналіз методів унікальної ідентифікації об'єктів будівництва
- обґрунтування вибору методу унікальної ідентифікації на основі відкритого коду місцеположення
- розробка схеми формування відкритого коду місцеположення та схеми формування UBID (від англ. Unique Building Identifier)
- аналіз реалізацій засобів формування на основі OLC
- розроблення прикладної функції та моделі БГД на основі відкритого коду місцеположення
- побудова алгоритму прикладної функції формування UBID на основі OLC
- аналіз реалізацій прикладної функції

- розроблення функціональної моделі дослідної БГД та проведення обчислювального експерименту дослідження результатів використання OLC коду для формування UBID в середовищі СКБД PostgreSQL/PostGIS/.

Методи дослідження. Методологічну основу роботи складають:

монографічний метод опрацювання наукових публікацій, нормативних документів, стандартів та інших матеріалів в сфері формування універсальних унікальних ідентифікаторів для геопросторових об'єктів в базах даних інформаційних систем;

методи формалізації для розроблення і подання технологічних моделей та моделей геопросторових даних з використанням IDF моделей та UML діаграм;

метод формування відкритого коду місцезоташування об'єктів OLC;

методи геоінформаційного аналізу геопросторових даних в ГІС;

методи програмування прикладних SQL функцій в середовищі об'єктно-орієнтованої СКБД PostgreSQL/PostGIS.

Новизна роботи: полягає в реалізації моделі і програмних засобів для формування універсального унікального ідентифікатора будівель в середовищі об'єктно-орієнтованої СКБД PostgreSQL/PostGIS.

Розділ 1
ОБГРУНТУВАННЯ, СТРУКТУРА, ОБМЕЖЕННЯ ТА ЗАВДАННЯ
ПРОЄКТУ

					ДИПЛОМНИЙ ПРОЕКТ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Виконав		Шкута Д.В.			Розроблення засобів формування унікальних ідентифікаторів будівель на основі відкритого коду місцеположення OLC в середовищі СКБД PostgreSQL	Літ.	Арк.	Аркушів
.		Лященко А.А.					1	20
Керівник		Лященко А.А.				ІПО КНУБА, група ГСТ-61		
Зав. каф.		Карпінський						

1 ОБҐРУНТУВАННЯ, СТРУКТУРА, ЗАВДАННЯ ТА ОБМЕЖЕННЯ ПРОЕКТУ

1.1 Призначення та завдання унікальної ідентифікація будівель в інформаційних системах управління будівництвом

Згідно законам та постанови Кабінету Міністрів України від 1 липня 2020 р. № 559 Про реалізацію експериментального проекту щодо запровадження першої черги Єдиної державної електронної системи у сфері будівництва кожен об'єкт нерухомості повинен мати унікальний ідентифікатор [1]. З появою нових кадастрових законів змінилися як вимогу до присвоєння ідентифікатора, так і якісне позначення символів. З огляду на те, що одна і та ж нерухомість може мати різні правові характеристики, передбачено присвоєння окремих ідентифікаторів.

У відповідності до закону України про регулювання містобудівної діяльності статті 22 «Єдина державна електронна система у сфері будівництва» створюється система яка призначена для ідентифікації будівель.[2]

Ідентифікатором об'єкта будівництва (закінченого будівництвом об'єкта) є унікальний набір даних (послідовність символів), що присвоюється об'єкту будівництва або закінченому будівництвом об'єкту та використовується для ідентифікації такого об'єкта. Ідентифікатор об'єкта будівництва (закінченого будівництвом об'єкта) присвоюється:

- 1) будинку, будівлі, споруди (у тому числі в комплексі);
- 2) складовим частинам будинку, будівлі, споруди (квартирам, гаражним боксам, машиномісцям, іншим житловим та нежитловим приміщенням, які після прийняття об'єкта в експлуатацію є самостійними об'єктами нерухомого майна);
- 3) лінійному об'єкту інженерно-транспортної інфраструктури.

Ідентифікатор об'єкта будівництва (закінченого будівництвом об'єкта) не присвоюється комплексам (будовам), чергам будівництва, пусковим комплексам, відокремленим частинам будинку, будівлі, споруди (блок-секції,

дільниці тощо), а також місцям загального користування та допоміжним приміщенням.

Ідентифікатор об'єкта будівництва присвоюється об'єкту під час:

1) внесення до Реєстру будівельної діяльності інформації про документ, що дає право на виконання будівельних робіт, — для об'єкта нового будівництва та його складових частин (квартир, вбудованих чи вбудовано-прибудованих житлових та нежитлових приміщень у будинку, будівлі, споруді, гаражних боксів, машиномісць, інших житлових та нежитлових приміщень, які після прийняття об'єкта в експлуатацію є самостійними об'єктами нерухомого майна). Ідентифікатор, який присвоюється складовій частині об'єкта, повинен включати ідентифікатор об'єкта, частиною якого він є;

2) внесення інформації до Реєстру будівельної діяльності вперше для об'єкта, щодо якого набуто право на виконання будівельних робіт або який прийнято в експлуатацію до запровадження Електронної системи, або для самочинно збудованого об'єкта, про:

- містобудівні умови та обмеження;
- погодження обґрунтованих відхилень від будівельних норм згідно із частиною п'ятою статті 10 Закону України “Про будівельні норми”;^[3]
- результати експертизи проектної документації на будівництво об'єктів;
- документи, що дають право на виконання підготовчих та будівельних робіт;
- документи, що засвідчують прийняття в експлуатацію закінчених будівництвом об'єктів;
- результати технічної інвентаризації об'єктів незавершеного будівництва, закінчених будівництвом об'єктів та їх частин (квартир, вбудованих чи вбудовано-прибудованих житлових та нежитлових приміщень у будинку, будівлі, споруді, гаражних боксів, машиномісць, інших житлових та нежитлових приміщень, які після прийняття об'єкта в експлуатацію є самостійними об'єктами нерухомого майна).

Ідентифікатор об'єкта будівництва (закінченого будівництвом об'єкта) є незмінним, крім випадків, визначених цим Порядком.

1) У разі поділу або об'єднання закінченого будівництвом об'єкта ідентифікатор закінченого будівництвом об'єкта анулюється. Закінченим будівництвом об'єктам, що утворилися внаслідок їх поділу або об'єднання, присвоюється новий ідентифікатор закінченого будівництвом об'єкта, який повинен містити посилання на ідентифікатор (ідентифікатори) об'єкта (об'єктів), з якого такі об'єкти утворилися.

2) У разі виділу частки із закінченого будівництвом об'єкта ідентифікатор об'єкта, з якого виділено частку, залишається незмінним. Закінченому будівництвом об'єкту, що утворився внаслідок виділу частки, присвоюється новий ідентифікатор закінченого будівництвом об'єкта. У відомостях про об'єкт, який утворився, зазначається посилання на ідентифікатор об'єкта, з якого такий об'єкт утворився.

3) У випадках, визначених цим Порядком, новий ідентифікатор закінченого будівництвом об'єкта присвоюється автоматично програмними засобами Реєстру будівельної діяльності під час внесення до Реєстру будівельної діяльності відомостей про технічну інвентаризацію відповідного об'єкта.

Ідентифікатор об'єкта будівництва або закінченого будівництвом об'єкта зазначається у таких документах:

1) технічних умовах (для закінчених будівництвом об'єктів та об'єктів нового будівництва, яким присвоєно ідентифікатор об'єкта будівництва до видачі технічних умов);

2) документі, що дає право на виконання підготовчих та будівельних робіт (крім повідомлення про початок виконання підготовчих та будівельних робіт щодо об'єкта нового будівництва);

3) рішеннях про присвоєння, зміну, коригування, анулювання адреси об'єкта будівництва та об'єкта нерухомого майна;

4) погодженні обґрунтованих відхилень від будівельних норм (для закінчених будівництвом об'єктів та об'єктів нового будівництва, яким присвоєно ідентифікатор об'єкта будівництва до отримання погодження);

5) енергетичному сертифікаті об'єкта будівництва та будівлі (для закінчених будівництвом об'єктів та об'єктів нового будівництва, яким присвоєно ідентифікатор об'єкта будівництва до видачі енергетичного сертифіката);

6) звіті про результати обстеження інженерних систем;

7) проектній документації на будівництво об'єкта та результатах її експертизи (для об'єктів, яким присвоєно ідентифікатор об'єкта будівництва до розроблення проектної документації);

8) документі, що засвідчує прийняття в експлуатацію закінченого будівництвом об'єкта;

9) інвентаризаційній справі, матеріалах технічної інвентаризації та технічному паспорті об'єкта незавершеного будівництва, закінченого будівництвом об'єкта та його частин (квартир, вбудованих чи вбудовано-прибудованих житлових та нежитлових приміщень у будинку, будівлі, споруді, гаражних боксів, інших житлових та нежитлових приміщень, які після прийняття об'єкта в експлуатацію є самостійними об'єктами нерухомого майна);

10) результатах контрольного геодезичного знімання закінченого будівництвом об'єкта.

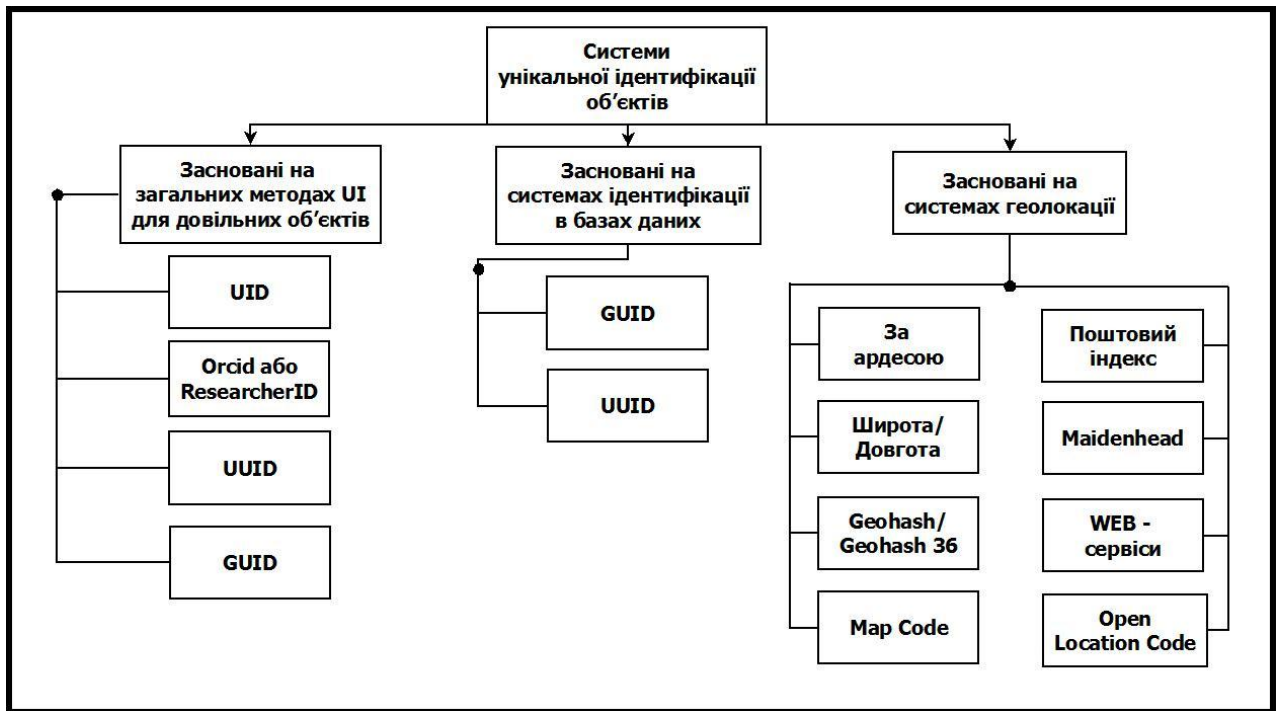


Рис. 1.1 Схема класифікації унікальних ідентифікаторів об'єктів

Розглянемо докладніше властивості кожної із систем формування універсальних ідентифікаторів GUID та UUID.

Глобальний унікальний ідентифікатор *GUID* (Globally Unique Identifier) та універсальний унікальний ідентифікатор *UUID* (Universally Unique Identifier) – це статистично унікальні 128-бітні ідентифікатори, які використовуються для ідентифікації інформації в комп'ютерних системах (далі позначаються загальним терміном *UUID*). Термін *GUID* використовується, як правило, в програмному забезпеченні, створеному Microsoft для ідентифікації програмних COM/OLE об'єктів, класів тощо.[5]

Головна особливість *UUID* – унікальність, яка дозволяє створювати розширювані сервіси, застосунки та записи в базах даних без побоювання конфліктів, викликаних збігом ідентифікаторів. 128-бітове значення ідентифікатора генерується за спеціальним алгоритмом та практично гарантується, що за цим же алгоритмом воно не буде отримано більше ніде в світі. Таким чином, ці ідентифікатори будуть унікальними і в розподілених системах, а не тільки в єдиній базі даних, як значення генераторів послідовностей.[6]

Унікальність UUID не залежить від центрального органу реєстрації або координації між сторонами, що їх створюють, на відміну від більшості інших схем нумерації.

Хоча ймовірність дублювання UUID не дорівнює нулю, вона досить близька до нуля, і на неї можна не звертати увагу з точки зору практичного застосування UUID для забезпечення унікальної ідентифікації об'єктів в комп'ютерних системах.

128 бітний (тобто 16 байтовий) ідентифікатор UUID в друкованому стандартному виді подається як 32 – знакова послідовність 16-річних чисел (на нижньому регістрі клавіатури) для 16 байт, що розбиті на групи дефісами й оточені фігурними дужками:

```
{d12beb59-6259-4fa1-a733-adcd523d72dc}
```

16-значне число відповідає позначенню 4 бітового двійкового числа (0 – 15) в символному виді (0, 1,2,3,4,5,6,7,8,9, a,b,c,d,e,f).

PostgreSQL підтримує стандартний та альтернативні варіанти подання: цифри у верхньому регістрі, стандартну запис, укладену в фігурні дужки, запис без мінусів або з мінусами, які поділяють будь-які групи з чотирьох цифр. наприклад:

```
A0EEBC99-9C0B-4EF8-BB6D-6BB9BD380A11
```

```
{A0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11}
```

```
a0eebc999c0b4ef8bb6d6bb9bd380a11
```

```
a0ee-bc99-9c0b-4ef8-bb6d-6bb9-bd38-0a11
```

```
{A0eebc99-9c0b4ef8-bb6d6bb9-bd380a11}
```

В PostgreSQL тип даних uuid призначений для зберігання ідентифікаторів UUID, визначених в стандартах RFC 4122, ISO / IEC 9834-8: 2005 та пов'язаних стандартах.

У PostgreSQL вбудовані функції зберігання і порівняння ідентифікаторів UUID, але немає внутрішньої функції генерування UUID, тому що не існує якогось єдиного алгоритму, придатного для всіх додатків. Згенерувати UUID можна за допомогою додаткового модуля *uuid-oss*, в якому реалізовані кілька

стандартних алгоритмів. Також можна скористатися модулем *pgcrypto*, де теж є функція генерування випадкових UUID. Крім того, можна зробити це в клієнтському додатку або в іншій бібліотеці, що підключена на стороні сервера.

Недоліком GUID є те, що він залежить від середовища застосування. Ідентифікатор GUID може бути застосований як засіб формування UBID.

Унікальний ідентифікатор будівель на основі координат контурів будинків (Unique Building Identifier, UBID). Заснований на референційній системі відліку, що використовується для визначення двомірних (x "та" y "розмірів" - просторових характеристик.

1.2 Огляд та порівняльний аналіз методів унікальної ідентифікації об'єктів будівництва

Дані про будівлі стають дедалі доступнішими, однозначне визначення будівельних записів у великих базах даних стає все більш важливим.

Вуличних адрес достатньо для таких випадків використання, як доставка пошти, але вони мають деякі помітні недоліки. По-перше, вони не стабільні, схильні до широкого діапазону варіацій формату та орфографії, що ускладнює точне узгодження. Що більш важливо, вони не мають геопросторової основи. Єдиний спосіб отримати адресу будівлі – це буквально десь її шукати, і такі пошуки піддаються як вищезазначеним варіаціям, так і природним неоднозначностям для будівель, які мають кілька адрес (наприклад, будинки з рядами) або не мають окремих вулиць взагалі (наприклад, будівлі, що є частиною кампусу). Ці проблеми роблять адреси менш ідеальними ідентифікаторами будівель у великих базах даних.

У багатьох частинах світу у більш половини міського населення відсутні адреси вулиць. Падіння цін на смартфони та їх все частіше впровадження можуть надати рішення, яке дозволило б зробити сигнали місцеположення GPS видимими та використаними як адреси. Для цього потрібно мати спосіб кодування інформації про місцеположення, щоб люди могли передавати місця розташування один одному.

Важливими атрибутами які повинен мати кожний код місцеположення для такої системи кодування є перелік наведений нижче [10]:

- коди повинні бути досить короткими, щоб їх можна було запам'ятати;
- код повинен бути достатнім сам по собі, не вимагаючи додаткової інформації, наприклад, назви країни;
- щоб уникнути плутанини, місце повинно мати лише один код;
- коди не повинні містити легко плутані символи (наприклад, 0 та O, 8 та B тощо);
- коди не повинні містити нецензурної лайки або, бажано, будь-яких слів будь-якою мовою;

- має бути можливим розглянути два коди і визначити, чи вони близькі один до одного, і навіть напрямок. Це допоможе людям знайти шлях і зробити коди більш корисними;

- коди повинні представляти область, а не точку, де розмір області мінливий;

- скорочення коду повинно надавати більшу область, що містить оригінальне розташування. Побічним ефектом цього є те, що коди поблизу матимуть загальний префікс;

- код місця повинен бути детермінованим, не вимагаючи налаштування чи застосування;

- коди повинні мати можливість створювати та декодувати в автономному режимі незалежно від середовища формування та зберігання даних;

- коди не повинні залежати від одного постачальника, щоб не було ризику, що вони перестануть працювати, якщо одна компанія припинить свою діяльність;

- алгоритм коду повинен бути відкритим та бути вільним для використання.

Щоб правильно приєднати різноманітні дані до будівлі, будівлю (або її частину) потрібно ідентифікувати за допомогою зовнішнього ключа - поля в реляційній базі даних, яке однозначно ідентифікує записи бази даних. Вуличні адреси в основному використовуються для ідентифікації місцеположення за допомогою людини, наприклад, доставки пошти або глобальної системи позиціонування.

В даний час щоб визначити місцеположення будівлі їх ідентифікують з адресою вулиці або з єдиною географічною координатною точкою (широтою та довготою).

Широта і довгота - це цифри із знаками з різними діапазонами. Порядок, в якому вони використовуються, має значення - якщо його змінити, вони представлятимуть інше місце.

Щоб виразити місцеположення з точністю до 10 метрів (п'ять знаків після коми), широта та довгота вимагають від 15 до 20 текстових символів ("0,39122,9,45225" до "-43,95134, -176,55053"). Це приблизно вдвічі довше типового телефонного номера.

Широта і довгота виражають точкове розташування, і не існує загальновизнаного способу визначити місце розташування чогось, що не є точкою, наприклад, футбольного поля, парку чи озера, крім надання кількох точок для огороження території. Зрізання широти та довготи не має жодного реального сенсу, оскільки воно просто переміщує місце розташування.

Дешеві GPS-пристрої існують щонайменше 14 років, і тим не менше координати широти та довготи все ще не широко використовуються людьми для вказівки місцезнаходжень. Це показує, що широта та довгота мають занадто багато недоліків, щоб бути прийнятими для рішення вуличної адреси.

Однією з найбільших проблем в гіперлокальній рекламі в Інтернеті є формування масштабованої техніки геокодування для створення ефективних пулів користувачів з різними налаштованими клієнтом радіусами навколо точки інтересу (POI). Об'єкти POI або точки інтересу - це об'єкти інфраструктури, гідності, природні об'єкти та важливі точки на дорогах, координати та інформація про те, які нанесені на карту GPS. Розмір POI або цільовий фізичний магазин надзвичайно різняться, і для досягнення найкращих результатів потрібно застосувати високо гнучку, але масштабовану техніку.[10]

На даний час існує багато систем та програм для ідентифікації місцезнаходження. Зручніше всього підходить Geohash, Geohash- 36, MapCode, Open Post Code, Nature Area Code, Maidenhead Locator System (MLS), Web-based services та Open Locaton Code. На прикладі було вибрано декілька з них для порівняння та аналізу.

Коди *Geohash* були розроблені для використання в коротких URL-адресах для ідентифікації місцезнаходжень. Вони використовують 32 символи у своєму наборі символів, що складається з 0-9A-Z, виключаючи "A", "I", "L" та "O". Це означає, що коди Geohash можуть містити голосні (і цифри "0" та "1", схожі на

"0" та "I"). Geohash може генерувати коди, що включають слова ("НАРКОТИЧНІ"), майже слова ("ZURICH") та нецензурну лексику. Він також включає деякі легко плутані символи, такі як "8" і "B".

Коди геохешу представляють області. Обрізання коду зменшує точність, розширюючи область за допомогою розширеної області, що містить вихідну точку.

Довжина коду Geohash залежить від кількості десяткових знаків координат. Це може призвести до того, що два коди, розташовані один біля одного, мають різну довжину, і це може заплутати у контексті адресації вулиць.

Geohash використовує двійковий алгоритм, який є досить елегантним, але в результаті виникають п'ять розривів (місця, де поблизу розташовані різні коди Geohash) на довготі 180, довготі 0, екваторі, а також на північному та південному полюсах. Довгота 180 та полюси є прийнятними через низьку кількість населення, але екватор та довгота 0 мають безліч центрів заселення вздовж них.[10]

Одне місце може мати більше одного коду, залежно від вхідних значень, і кілька різних кодів можуть декодуватися до одного і того ж значення. Наприклад, "c216ne4" та "c216new" (та інші) декодуються до однакових координат (45,37 -121,7).

Геохаш-36. Коди Geohash-36 призначені для URL-адрес та електронного зберігання та зв'язку, а не для людської пам'яті та розмов.[17]

Geohash-36 не включає голосні, але включає як версії нижнього, так і верхнього регістру деяких букв ("b" і "B"). Він також включає символи, які можна легко сплутати ("6" і "b", "8" і "B").

Geohash-36 базується на сітці 6x6, де кожна клітинка ідентифікується символом. Це призводить до того, що коди над межею відрізняються, хоча вони можуть бути сусідами:

2	3	4	5	6	7
8	9	b	B	C	d
D	F	g	G	h	H
j	J	K	I	L	M
n	N	P	q	Q	r
R	t	T	V	W	X

Рис. 1.2. Розташування кодів 9x, g2, g7 та G2

Маючи лише два рівні, ми можемо побачити, що клітинка "g2" (червона, лівий верхній кут комірки, позначений g) знаходиться поруч із коміркою 9X, але далі від g7 (яка знаходиться поруч з G2). Використовуючи справжні коди Geohash-36, "bdg345476Q" знаходиться поруч із "bdbtTVTXWB", але в декількох кілометрах від "bdg3Hhg4Xd".

Коди Geohash-36 можуть бути на один символ коротші, ніж повні коди відкритого розташування для подібної точності.

Визначення Geohash-36 включає необов'язкову специфікацію висоти та необов'язкову контрольну суму, жодне з яких не передбачено відкритим кодом місцеположення.

Система локації Maidenhead (MLS). Системні коди Maidenhead Locator явно представляють області і можуть бути усічені подібно до відкритих кодів розташування. Точність і довжина кодів є подібними, але коди системи Maidenhead Locator включають голосні, тому сформовані коди включають слова.

Системні коди Maidenhead Locator базуються на чергуванні широти та довготи, тому їх можна обрізати, а сусідні місця мають подібні коди. Формально він визначений лише довжиною до 8 символів.

Засновані на веб-сервісах. Нещодавно було створено різноманітні веб-сервіси, такі як MyDoorHandle , What3Words та Zippr .

Ці веб-сайти надають код, який при введенні на їх веб-сайті або використанні в URL-адресі відкриває веб-сторінку, що відображає місцеположення. Це схоже на, наприклад, Geohash, але на відміну від Geohash, коди місця можуть не існувати, поки хтось не подасть заявку на них. Коди зазвичай представляють точку.[12]

Коди можуть генеруватися псевдовипадковим чином, тому місця поблизу можуть мати зовсім інші коди. Можливо, кілька людей можуть подати заявку на отримання кодів для одного місця та для створення різних кодів.

Коди What3words використовують три слова для позначення квадратів 3x3 метри. Коди були призначені для всіх місць на Землі за допомогою патентованого алгоритму. Можна придбати окремі слова ("OneWord").

Помилка з кодом може просто відобразитися деінде - наприклад, на What3Words, "бананова кроляча мавпа" - це місце в Аргентині, "бананова кроляча мавпа" - у Росії.

Деякі служби беруть гроші або за надання коду, за розпізнавання кодів або за те, що дозволяють користувачам вибрати власний короткий код.

Ці системи не працюють в автономному режимі і мають єдиного постачальника. Здається, вони більше націлені на те, щоб бути бізнес-довідниками, розміщуючи додаткову інформацію, таку як контактні дані, фотографії тощо, крім місця розташування.

Відкритий код місцеположення (від англ. Open Location Code).

Атрибути вищезазначених систем недостатньо відповідають нашим вимогам. В результаті було створено нову систему кодування Open Location Code; коди, створені за допомогою цієї системи, називаються плюсовими кодами.

Коди плюс мають довжину від 10 до 11 символів. Їх також можна використовувати в короткій формі з чотирьох до семи символів, подібно до телефонних номерів та поштових індексів, приблизно за 50 км від початкового місця розташування. Приблизно на відстані 2,5 км від початкового місця їх можна ще більше скоротити - лише до чотирьох-п'яти символів.

У своїй короткій формі плюсові коди мають від чотирьох до семи символів. Вони можуть використовуватися самостійно в межах 50 км від місця або в усьому світі, надаючи місто чи населений пункт на відстані. Повний плюс-код не вимагає жодної іншої інформації, щоб знайти їх.

Існує лише один плюс-код для даного місця та розміру площі. Можна створювати різні коди для різних областей, але вони матимуть спільних провідних символів.

Символи коду відкритого коду виключають легко сплутані пари символів. Існує ризик того, що "VV" буде змінено з "W" у рукописних повідомленнях, але це малоймовірно, оскільки це змінить довжину коду, і це має виявити користувач або одержувач.

Набір символів для Open Location Code був обраний із понад восьми мільярдів можливостей, використовуючи список слів із 10 000 слів із 30 мов. Усі можливі набори оцінювались на предмет того, чи вміють вони писати тестові слова, а найбільш перспективні набори оцінювались вручну.

Набір символів, що використовується для формування плюсових кодів, не суміжний. Це результат видалення легко плутаних символів, голосних та деяких інших символів. Це ускладнює ручне порівняння кодів, оскільки потрібно пам'ятати, чи є символи від 9 до C, щоб визначити, чи 8FV9 поруч із 8FVC. Однак ми вважаємо, що це виправдано покращеною зручністю використання.

Місця поблизу мають подібні плюсові коди. Є три розриви, на довготі 180, а також на північному та південному полюсах, де сусідні місця можуть мати дуже різні коди.

За певної практики можна оцінити напрямок і навіть дуже великі відстані між двома кодами. Завдяки способу генерації кодів, широти відсікаються так, щоб вони були більшими або дорівнювали -90 і менше 90 градусів, що робить подання точного розташування Північного полюса неможливим, хоча його можна дуже точно наблизити.

Плюс коди представляють області, а розмір області залежить від довжини коду. Чим довший код, тим менша і точніша область. Обрізання плюсового коду збільшує площу та містить оригінальне розташування.

Коди засновані на простому кодуванні широти та довготи. Код місця може переглянути кожен і не вимагає налаштування. Плюс-коди можна кодувати та декодувати в автономному режимі.

Коди плюс не залежать від будь-якої інфраструктури, а тому не залежать від жодної організації чи компанії для їх подальшого існування чи використання.

В ході аналізу всіх систем для ідентифікації місцеположення в нашій роботі було обрано Open Location Code(OLC), оскільки він кращим варіантом для визначення вуличної адреси.

1.3 Обґрунтування вибору методу формування UBID на основі відкритого коду місцеположення OLC

Структура UBID сформована на основі референцної системи відліку, враховуючи її перевагу у визначенні двомірних (х "та" у "розмірів") будівельних характеристик в певній структурі. У цьому розділі представлено методологію перетворення місця розташування та контуру забудови на код референцної сітки. [22]

OLC заснований на простому кодуванні широти та довготи. Алгоритм є відкритим кодом. OLC може бути закодований, і розшифровується в режимі офлайн, не залежно від будь-якої інфраструктури або одного постачальника. OLC є простим і не вимагає жодної інфраструктури або налаштування. Більше того, OLC заснований на рекурсивно підрозділеній прямокутній сітковій референцній системі, де комірки у послідовних шарах повністю покривають їх попередників (тобто комірок у попередніх шарах) без накладання. Накладаючи сітку OLC на цифрову карту, можна шукати будівлю на сітці OLC, використовуючи її вуличну адресу на веб-сайті OLC.

Було розроблено методологію одного розміру для вирішення майже необмежених варіацій конструкцій будівель. Основне питання - як наблизити будь-яку форму контуру до прямокутника, який вписується в лінії сітки OLC. Пропонується використовувати найменший прямокутник, який може покрити будівельний контур. На рис. 1.3 показано чотири кроки формування ідентифікатора будівлі на основі OLC та меж охоплюючого прямокутникаЮ а саме:

- 1) отримання двовимірної геометрії будинків та визначення ширини /довжини кутів та двовимірний центр маси;
- 2) перетворення центр маси в контрольну клітинку OLC;
- 3) створення двовимірного обмежувального вікна (а саме прямокутник), що покриває контур вздовж сітки OLC;

4) обчислення кількості комірок OLC від опорної комірки до краю обмежувального поля в північному, східному, південному та західному напрямках;

5) Створення UBID, використовуючи інформацію з кроків 2 та 4.

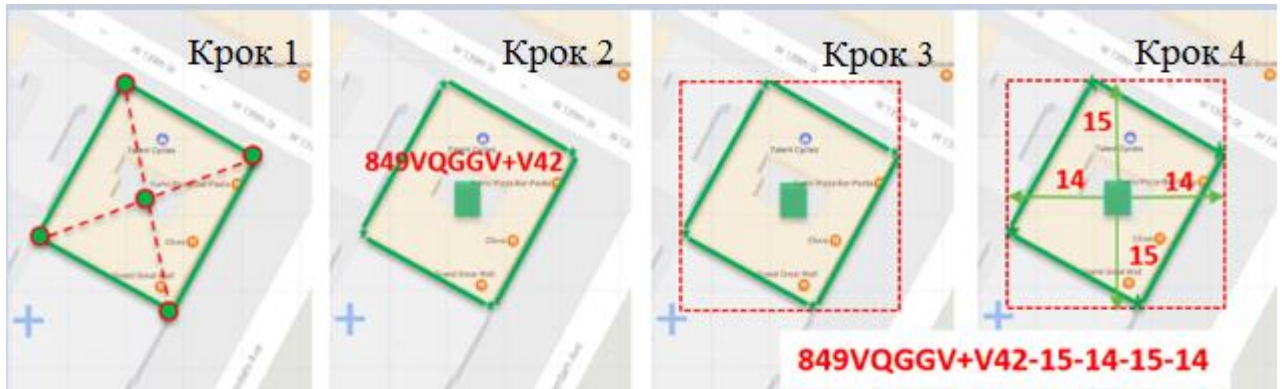


Рис.1.3. Етапи створення найменших прямокутників

UBID для прикладу будівлі, показаного на малюнку 1.3, містить основний корпус (центральный OLC з геометрії та числа осередків OLC на північний, східний, південний та західні напрямки). Визначаємо двовимірний центроїд, обчислений з геометрії площі будівлі, як центроїд UBID. Він попадає в контур у більшості випадків, крім двору будівлі. Обмежувальний прямокутник може перетинати контур сусідньої будівлі (та її обмежувальних прямокутників), оскільки обмежувальний прямокутник може бути набагато більшим, ніж будівельний контур залежно від форми контуру та орієнтації будівлі. Однак це не впливає на унікальність та ефективність UBID. За допомогою UBID можна відтворити вікно обмеження OLC на карті та визначити або перевірити будівлю, яку він містить. Для полегшення співставлення будівель можна розробити додаткові показники.

Інформаційний зміст унікального ідентифікатора будівлі (UBID) - це дві геометрії:[23]

- геометрія комірки сітки Open Location Code (OLC), що містить центроїд контуру будівлі;
- геометрія обмежувальної рамки, вирівняної за сіткою OLC, яка визначається північним, східним, південним та західним діапазонами.

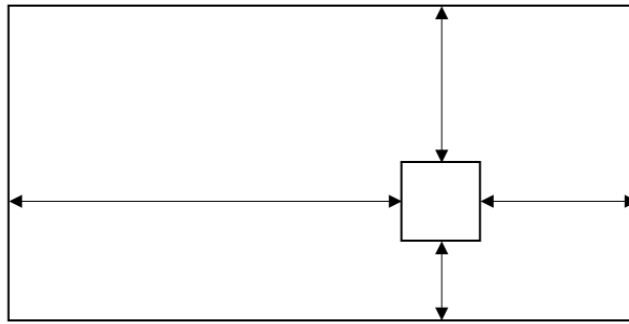


Рис.1.4. Приклад типового UBID без геометрії джерела для межі будівлі

Зображення типового UBID наведено на рис 1.4. Візуалізації UBID - це композиції візуалізацій двох геометрій, разом з візуалізацією півночі, сходу, півдня та заходу. Геометрія та протяжність зображені відповідно у вигляді прямокутників та двосторонніх стрілок.

Якщо північний, східний, південний та західний напрямки протяжні рівні дорівнюють нулю, то, за визначенням, дві геометрії ідентичні, і зображена лише комірка сітки OLC, що містить центроїд межі будівлі, тобто OLC-сітка - вирівняний обмежувальний прямокутник і розміри не зображені.

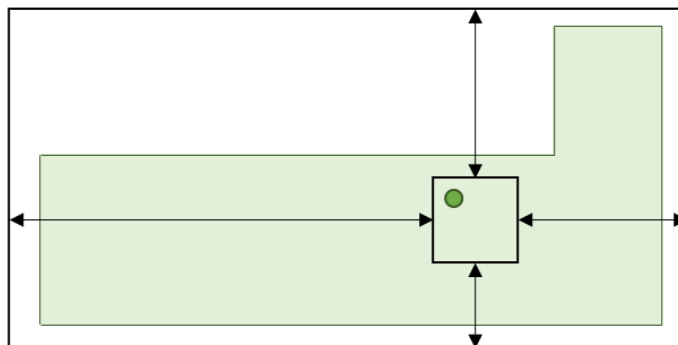


Рис.1.5. Приклад типового UBID з геометрією джерела для межі будівлі

Геометрія джерела для межі будівлі зображена у вигляді обмежувального прямокутника.

У деяких випадках використання для візуалізації вигідно зобразити центроїд геометрії джерела для межі будівлі. На рис.1.5 центроїд зображений у вигляді точки.

1.4 Структура, завдання та обмеження проекту

В цій роботі запропоновано метод ідентифікації будівель на основі відкритого коду місцеположення OLC. Проаналізовано системи та методи формування ідентифікаторів, розглянуто призначення та завдання ідентифікаторів. Обґрунтовано вибір метода унікальної ідентифікації будівель UBID. Для того, щоб представити структуру коду було наведено схему формування відкритого коду місцеположення та технологічну схему UBID, описано якими інструментами та засобами формується код.

За допомогою засобів та визначеної технології розроблено алгоритм прикладної функції формування UBID.

Для розкриття питань щодо ідентифікації об'єктів розглядається література та різні організації. В останніх розділах наведено приклади формування засобів унікальних ідентифікаторів будівель та надано результати обчислювального експерименту формування UBID основі відкритого коду місцеположення OLC.

Обмеження проекту полягають у використанні лише одного програмного продукту для практичної реалізації засобів формування унікальних ідентифікаторів будівель.

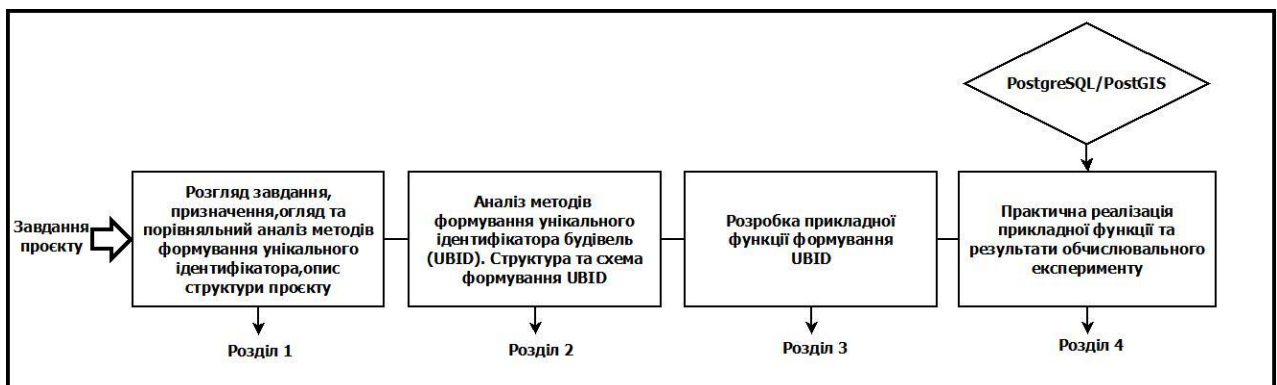


Рис. 1.6. Завдання, структура та обмеження проекту

Цим продуктом є об'єктно-реляційна система керування базами даних (СКБД) PostgreSQL, яка є програмою з відкритим кодом, тобто даний програмний продукт не контролюється якоюсь однією компанією, її розробка є можливою завдяки співпраці багатьох людей та компаній, які прагнуть

використовувати, розповсюджувати та розвивати цю СКБД впроваджуючи у неї найновіші досягнення. Окремо слід виділити розширення PostGIS, яке додає підтримку географічних об'єктів в реляційну базу даних PostgreSQL, дозволяючи виконувати операції над координатно-просторовими характеристиками об'єктів БД.

Розділ 2

Методичні засади формування унікальних ідентифікаторів будівель (UBID) на основі відкритого коду місцеположення oIs

					ДИПЛОМНИЙ ПРОЕКТ					
Змн.	Арк.	№ докум.	Підпис	Дата	Розроблення засобів формування унікальних ідентифікаторів будівель на основі відкритого коду місцеположення OLC в середовищі СКБД PostgreSQL			Літ.	Арк.	Аркушів
Виконав	Шкута Д.В.							1	13	
.	Лященко А.А.									
Керівник	Лященко А.А.							ІПО КНУБА, група ГСТ-61		
Зав. каф.	Карпінський									

2 МЕТОДИЧНІ ЗАСАДИ ФОРМУВАННЯ УНІКАЛЬНИХ ІДЕНТИФІКАТОРІВ БУДІВЕЛЬ (UBID) НА ОСНОВІ ВІДКРИТОГО КОДУ МІСЦЕПОЛОЖЕННЯ OLC

2.1 Загальна структура та просторові властивості відкритого коду місцеположення OLC

Згідно специфікації [11] подано загальну структуру та сформовані просторові властивості OLC коду.

Відкритий код місцеположення - це технологія, яка надає спосіб кодування місцеположення у форму, простішу у використанні, чем широта та довгота. Згенеровані коди називаються плюс-кодами, оскільки їх відмінним атрибутом є те, що вони містять символ "+". Відкритий код місцеположення кодує два числа в один короткий рядок.

Технологія розроблена для створення кодів, які можна використовувати як унікальні ідентифікатори, особливо в місцях, де будівлі не нумеровані або вулиці не названі.

Вхідними значеннями широти та довготи повинні бути значеннями WGS84. Якщо використовуються інші дані, це повинно бути зазначено та чітко вказано, що вони будуть містити різні місця.

Далі визначаються допустимі символи у коді відкритого місцеположення. Послідовності, що містять інші символи, за визначенням не є дійсними кодами відкритого місцеположення.

У загальному випадку код OLC може містити до 15 символів за такою структурою див. рис. 2.1.

Символи коду для перших 4-х пар значень широти/довготи на ієрархічних сітках								Роздільник	Символи коду 5-ї пари широти/довготи		11-15 Символи коду місцеположення точки на сітках 4x5			
Сш	Сд	Сш	Сд	Сш	Сд	Сш	Сд	+	Сш	Сд	Ск	Ск	Ск	Ск

Рис 2.1. Загальна структура OLC коду, де:

(Сш – символ широти; Сд – символ довготи; Ск – символ клітини)

Символи OLC коду місцеположення були обрані для зменшення помилок під час написання та запобігання випадковому написанню слів. На рис.2.2. приведено відповідність числового номеру координатної сітки кодування символу в 10-значному кодї OLC.

	Відповідність коду для індексів координатної сітки																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Символ коду	2	3	4	5	6	7	8	9	С	F	G	H	J	M	P	Q	R	V	W	X

Рис. 2.2. Відповідність перших 10-ти символів OLC коду значенням широти/довготи на координатних сітках

Цифри коду та впорядкування не змінюються в мовах RTL. Номер широти повинен бути відсічений, щоб знаходитись в діапазоні від -90 до 90. Число довготи потрібно нормувати, щоб воно знаходилося в діапазоні від -180 до 180.

Код починається з п'яти пар цифр, кожна з яких складається з однієї цифри, що представляє широту та одну, що представляє довготу. Найбільші блоки мають лише дві цифри. Після 8 цифр знак коду "+" вставляється в код як роздільник для полегшення візуального аналізу. Після 10 цифр у кожному підрозділі субблоки кодуються однією кодовою цифрою наступним чином:

	довгота (longitude) →			
широта (latitude) →	R	V	W	X
	J	M	P	Q
	C.	F	G	H
	6	7	8	9
	2	3	4	5

Рис. 2.3. Розділ підблоків у відкритих кодах місцеположення для символів коду після 10 символів

Області, більші за 8-значний блок, можна вказати, замінивши парну кількість кінцевих цифр перед знаком + на цифру 0, нічим після знаку +.[12]

OLC код складається із 10 цифр. Щоб їх вивести потрібно зробити деякі обчислення: додати 90 до широти та 180 до довготи, щоб привести їх у позитивні діапазони; кодувати як широту, так і довготу в основу 20,

використовуючи наведені вище символи, для п'яти цифр кожна, тобто до значення місця 0,000125. Починаючи з широти, чергувати цифри.

Далі подано алгоритм кодування значень від першої до останньої цифри:

1) додати 90 до широти та 180 до довготи, помноживши обидва на 8000 і прийняти цілі частини відповідно як широту та довготу;

2) префіксувати існуючий код символом, що має цілу частину модуля довготи 20;

3) префіксувати існуючий код символом, який має цілу частину модуля широти 20;

4) поділити довготу та ширину на 20;

5) повторити крок 2 ще чотири рази.

Інший метод відрізняється від наведеного вище методу тим, що на кожному кроці створюється один символ. Це кодує широту в основу п'ять і довготу в основу чотири, а потім поєднує цифри для кожної позиції разом.

Алгоритм для кодування значень:

1) додати до ширини 90, помножити дробову частину на $2,5e7$ і прийняти цілу частину як широту;

2) додати до довготи 180, помноживши дробову частину на $8,192e6$ і прийняти цілу частину як довготу;

3) взяти цілу частину модуля широти 5. Помножити це на 4 і додати цілу частину модуля довготи 4;

4) префіксувати існуючий код символом із зазначеним вище значенням;

5) розділити довготу на чотири, а широту на п'ять;

6) повторити крок 2 ще чотири рази.

Мінімальна допустима довжина відкритого коду розташування - дві цифри. Максимальна довжина коду відкритого місця - 15 цифр. Довжина за замовчуванням для більшості цілей - 10 цифр.

Формування здійснюється завдяки тому, що розділювач формату потрібно вставити після восьми цифр. Якщо довжина запитуваного коду менше восьми

цифр, решта цифр перед роздільником формату повинні складатися з символу заповнення.

У наступній таблиці наведено точність дійсних довжин коду в градусах та метрах. Там, де точність різниться між широтою і довготою, показано обидва:

Довжина коду	Просторове розрізнення коду	
	Градуси	Метри
2	20	2226 км
4	1	111.321 км
6	1/20	5566 м
8	1/400	278 м
10	1/8000	13.9 м
11	1/40000 x 1/32000	2.8 x 3.5 м
12	1/200000 x 1/128000	56 x 87 см
13	1/1e6 x 1/512000	11 x 22 см
14	1/5e6 x 1/2.048e6	2 x 5 см
15	1/2.5e7 x 1/8.192e6	4 x 14 мм

Рис.2.4. Просторове розрізнення OLC кодів різної довжини
(розрізнення в метрах відповідає значенню 1° на екваторі 111 321 метрів)

Координати, отримані при декодуванні, - це південно-західний кут. Координати північно-західного кута та центру можна отримати, додавши значення просторового розрізнення. Це означає, що північно-західні координати не входять в область коду, за винятком кодів, північна широта яких становить 90 градусів.

Крім довгих кодів існують і короткі коди, що використовуються щодо еталонного місця. Вони дозволяють частині коду бути коротшими, простішими у використанні та легшими для запам'ятовування.

Короткі коди мають щонайменше дві та максимум шість цифр, видалених з початку коду. Отриманий код повинен містити символ "+" (роздільник формату). Коди, що містять заповнені символи, не можна скорочувати.

Цифри можна видалити з коду, доки точність позиції не буде менше подвоєного максимуму зміщення широти чи довготи між центром коду та місцем відліку. Відновлення вихідного коду повинно відповідати тим самим критеріям.

Основні вимоги до впровадження бібліотеки Open Location Code повинні забезпечувати:

- метод перетворення широти та довготи у 10-значний код відкритого місцеположення;

- метод декодування 10-значного коду відкритого місцеположення як мінімум на широту та довготу південно-західного кута та висоту та ширину;

- метод визначення, чи є рядок допустимою послідовністю символів Open Location Code;

- метод визначення, чи рядок є дійсним повним кодом Open Location;

- методи декодування та перевірки не повинні враховувати регістр.

Реалізації бібліотеки Open Location Code можуть забезпечити:

- метод перетворення широти та довготи у будь-яку допустиму довжину відкритого коду місцеположення;

- метод декодування будь-якої допустимої довжини відкритого коду місцеположення, надаючи додаткову інформацію, таку як координати центру;

- спосіб перетворення дійсного відкритого коду місцеположення у короткий код із заданим місцем посилання;

- метод відновлення повного коду відкритого місця з короткого коду та контрольного місця;

- метод визначення, чи є рядок дійсним коротким кодом Open Location Location.

2.2 Схеми формування OLC коду

Докладніше розглянемо схему формування коду OLC на прикладі кодування Головного корпусу КНУБА.

OLC код має блоки 20 на 20 градусів (9 рядків та 18 стовпців) і ділиться на 20 на 20 підблоків до чотирьох разів. Початкова пара кодів ідентифікує клітину із сітки 18 x 9, що покриває Землю, де кожна клітина в сітці дорівнює 20 градусів на 20 градусів.[15]

Коди складаються з послідовності цифр, вибраних із набору 20. Цифри в кодї чергуються між широтою та довготою.

На прикладі формування UBID на основі відкритого коду місцеположення було взято головна корпусу КНУБА з координатами **50,427553; 30,465786**.

Код складається з п'яти пар - 10 цифр. Щоб їх вивести потрібно зробити деякі обчислення. А саме наступні 6 кроків:

Крок 1: Визначення першої пари символі

Додати 90 до широти та 180 до довготи, щоб привести їх у позитивні діапазони; кодувати як широту, так і довготу в основу 20.

$$(50,427553+90)/20=7,02137765$$

$$(30,465786+180)/20=10,5232893$$

Отримані числа прийняти цілі частини відповідно як широту та довготу, округливши їх до меншої частини, тобто:

$$7,02137765 = 7$$

$$10,5232893 = 10$$

Підставляючи отримані значення у таблицю 2.1 отримаємо два перші числа коду

7 – **9**

10 – **G**

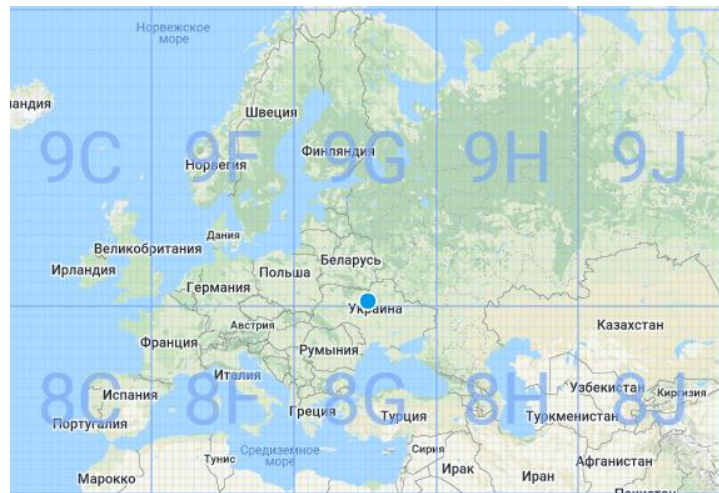


Рис. 2.5. Область перших двох символів коду

Крок 2: Визначення другої пари символів

Обчислення залишку координат виконується на кожному кроці.

Кількість кроків множимо на 20.

$$140,427553 - 140 = 0,427553 / 1 = 0,427553 = 0$$

$$210,465783 - 200 = 10,465783 / 1 = 10,465783 = 10$$

0 – 2

10 – G

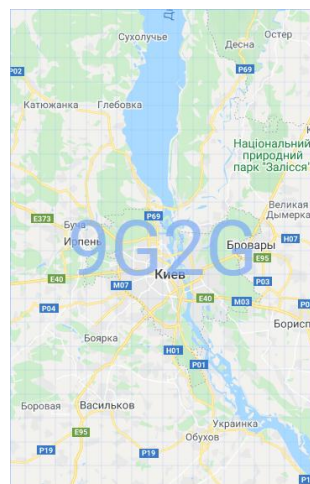


Рис.2.6. Область перших чотирьох символів

Перші чотири цифри коду описують один градус довготи та один градус широти, вирівняний за градусами. **9G2G** це прямокутник в область якого входять координати головної будівлі КНУБА.

Крок 3: Визначення третьої пари символів

Додаючи до коду ще дві цифри, площа зменшується до $1/20$ градуса на $1/20$ градуса в межах попередньої області.

$$0,427553/0,05=8,55106$$

$$0,465783/0,05=9,31572$$

8 – С

9 – F

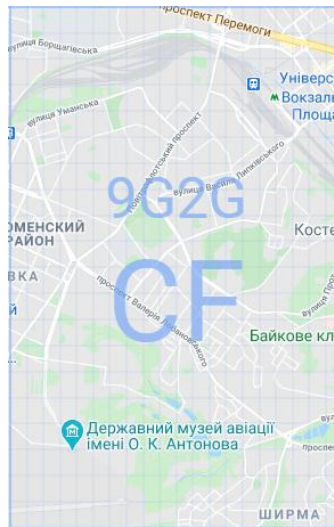


Рис. 2.7. Область перших шести символів

Крок 4: Визначення четвертої пари символів

Далі - кожна пара цифр зменшує площу до $1/400$ від попередньої площі.

$$0,027553/0,0025=11,0212-0,42=10,6212$$

$$0,015786/0,0025=6,3144$$

10 – G

6 – 8

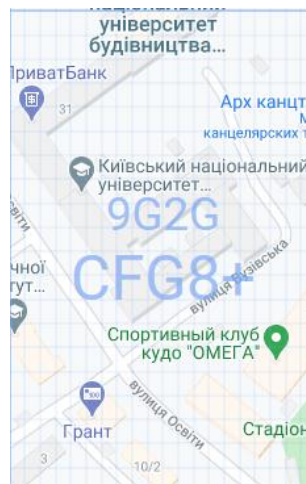


Рис. 2.8. Область перших восьми символів

Крок 5: Визначення п'ятої пари символів

$$0,002553/0,000125=20,424-0,427553=19,996447$$

$$0,00786/0,000125=9,35523$$

19 – **X**

9 - **F**

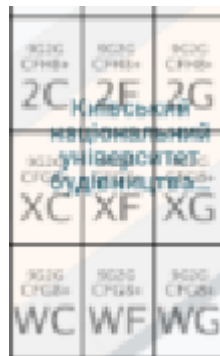


Рис. 2.9. Область знаків після роздільника « + »

Крок 6: Визначення 11 символу коду

Виконується з використанням сітки 5x4, в якій результати цілочисленого ділення дають номер рядка та номер стовпчика клітини, з якої береться 11-й символ коду.

В результаті визначення всіх символів складається загальний OLC код головного корпусу КНУБА.

Open Location Code(КНУБА) = 9G2GCFG8+XF

2.3 Аналіз реалізацій засобів формування OLC коду

Відкритий код місцеположення (OLC) є системою геокодування для ідентифікації області в будь-якому місці на Землі. Він був розроблений в інженерному офісі Google у Цюріху і випущений наприкінці жовтня 2014 року. Коди місцезнаходжень, створені системою OLC, називаються " плюсовими кодами " [15].

З серпня 2015 року Карти Google підтримують плюсові коди у своїй пошуковій системі. Алгоритм ліцензований за ліцензією Apache 2.0.[14] та доступний на GitHub. Google заявляє, що плюсові коди приймаються як поштові адреси в Кабо-Верд частинах Калькутті та в країні Навахо.

На прикладі формування OLC коду місцеположення розглянемо два способи, інтерактивний та на пакеті JavaScript.

Перший спосіб – інтерактивний, оснований на сервісах Google Maps на сайті Google Plus [13].

Для того, щоб знайти код будівлі потрібно зайти на сайт Google Plus та натиснути кнопку виклику пошуку коду.

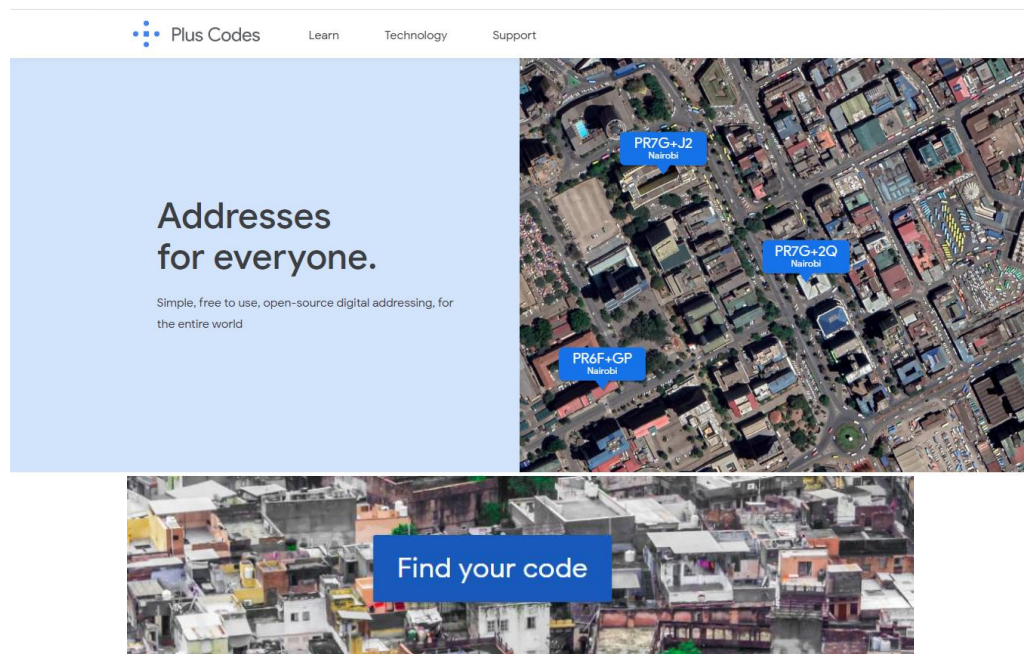


Рис .2.10. Кнопка пошуку кода

Додаток Google Sheets поєднує в собі плюсові коди із силою обробки даних. Це дозволяє перетворити координати широти та довготи в плюсові коди та створити адреси на основі плюсового коду.

Додаток Grid Service дозволяє додавати сітку Plus Codes на онлайн-карти та ГІС-програми, що полегшує їх пошук та використання.

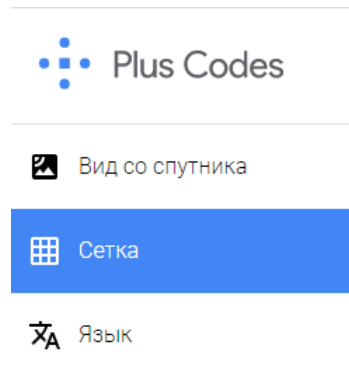


Рис. 2.11. Додавання сітки Plus Code на онлайн карту

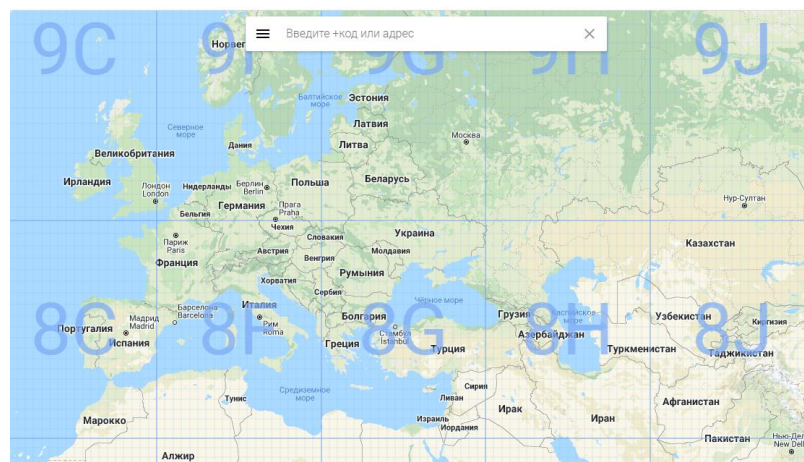


Рис. 2.12. Сітка з першими двома символами OLC коду

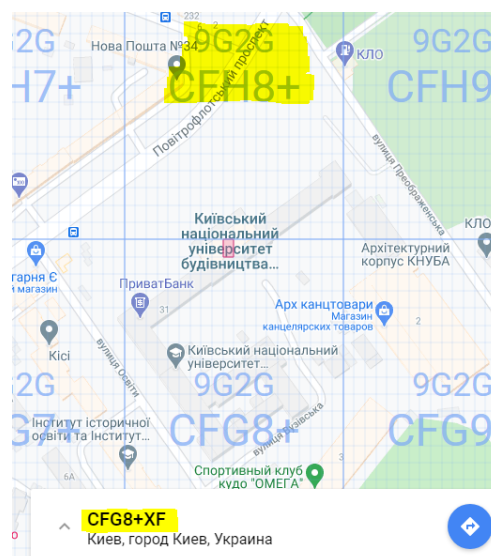


Рис. 2.13 Загальний вигляд OLC коду для головного корпусу КНУБА

Спосіб запропонований Google є простим у використанні та дає змогу швидко визначити код будівлі. Код надає тільки 10 символів. Зазвичай цього достатньо для точного визначення місцеположення.

Для більш точного визначення слід використовувати другий спосіб оснований на пакеті JavaScript. Перевага в цьому способі над інтерактивним полягає в тому, що пакет JavaScript забезпечує кодування, декодування та перевірку правильності коду аж до 11 символу.

Коди можна створювати та декодувати в автономному режимі та використовувати зменшений набір символів, що мінімізує можливість коду, включаючи слова.

Коди представляють прямокутні області, а не точки, і чим довше код, тим менша площа. Код із 10 символів представляє себе 13,5х13,5м площі метра на екваторі. Код із 11 символів приблизно представляє площу 2,8х3,5 метра.

У 11-символьному коді алгоритм змінюється таким чином, що кожен символ вибирає по одному положенню з сітки 4х5.

Опираючись на ліцензію,[25] нижче приведено алгоритм функції для формування OLC коду.

Функції

encode_olc

decode_olc

olctools

recover_olc

shorten_olc

validate_olc

encode_olc - створює відкриті коди місцеположення із значень широти та довготи заданої довжини.

decode_olc - симетрична операції та *shorten_olc* для перетворення "повних" відкритих кодів місцеположення на "короткі" коди.

Використання:

encode_olc(lats, longs, length)

Аргументи:

lats - числовий вектор широт;

longs - тягне числовий вектор довгот, еквівалентний за розміром латам довготи;

length - довжина OLC коду. Звичайні довжини – 10 або 11, причому будь-яке число вище 8 і будь-яке парне число нижче нього є прийнятним. Довжина повинна складатися з одного значення. Щоб обчислювалися всі коди до тієї ж довжини, або вектор значень того самого розміру, що *lats* та *longs* потрібно попередньо встановити значення.

decode_olc - приймає Open Location Codes і, якщо вони дійсні (див. *validate_full*), повертає мінімум по центру та максимальну широту і довготу для цих координат.

olctools - інструменти для роботи з відкритими кодами місцеположення

recover_olc - відновлення повних відкритих кодів місцеположення зі скорочених кодів

shorten_olc - скорочує код, зменшуючи простір, який він займає. Вони також обмежують його можливість перекладу назад у пари широти / довготи. *recovery_olc* відновлює повний код з скорочений, що дозволяє декодувати його за допомогою *decode_olc*. Деяка втрата точності є очікуваною, оскільки він знаходить найближчий збіг з координатами, а не з вихідним кодом, символи можуть бути різними.

Одна з речей, яка робить OLC корисними, полягає в тому, що їх можна скоротити - можна обрізати символи від них, заощаджуючи простір без істотної втрати точності. *shorten_olc* займає повну довжину OLC (генерується за допомогою *encode_olc* або будь-яким іншим способом) і скорочує їх.

Використання:

recovery_olc (olcs, lats, longs)

Аргументи:

olcs - вектор коротких відкритих кодів місцезнаходжень, згенерований за допомогою *shorten_olc* або через будь-які інші засоби;

lats - числовий вектор широт;

lons - числовий вектор довгот, еквівалентний за розміром латам.

validate_olc - ці функції дозволяють користувачеві перевіряти, чи надані ними OLC є дійсними чи ні. *valid_short* визначає, чи є вектор OLC дійсними "короткими" кодами; *valid_long* ідентифікує чи OLC є дійсними "довгими" кодами, і *valid_full* визначає, чи OLC є дійсними.

Використання[^]

validate_olc(codes)

validate_short(codes)

validate_full(codes)

Аргумент:

codes - вектор символів, що містить відкриті коди розташування.

Приклад

#Validate that a particular OLC is valid

validate_olc("WC2345+G6g")

#[1] TRUE

#It is! Is it a short?

validate_short("WC2345+G6g")

#[1] TRUE

#Yep!

#So it's not full?

validate_full("WC2345+G6g")

#[1] FALSE

#Nope!

Вектор значень TRUE та FALSE, де TRUE відповідає валідному коду, а FALSE не валідному.

Розділ 3

**РОЗРОБЛЕННЯ ПРИКЛАДНОЇ ФУНКЦІЇ ДЛЯ
ФОРМУВАННЯ UVID НА ОСНОВІ ВІДКРИТОГО КОДУ
МІСЦЕПОЛОЖЕННЯ OLC В СЕРЕДОВИЩІ СКБД
POSTGRESQL**

					ДИПЛОМНИЙ ПРОЕКТ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Виконав		Шкута Д.В.			Розроблення засобів формування унікальних ідентифікаторів будівель на основі відкритого коду місцеположення OLC в середовищі СКБД PostgreSQL	Літ.	Арк.	Аркушів
.		Лященко А.А.					1	12
Керівник		Лященко А.А.				ІПО КНУБА, група ГСТ-61		
Зав. каф.		Карпінський						

3 РОЗРОБЛЕННЯ ПРИКЛАДНОЇ ФУНКЦІЇ ДЛЯ ФОРМУВАННЯ UBID НА ОСНОВІ ВІДКРИТОГО КОДУ МІСЦЕПОЛОЖЕННЯ OLC В СЕРЕДОВИЩІ СКБД POSTGRESQL

3.1 Модель бази геопросторових даних для формування UBID на основі OLC

Концептуальна модель – це абстрактна модель, що виявляє причинно-наслідкові зв'язки, властиві об'єкту, що досліджується, і суттєві в рамках певного дослідження. Основне призначення концептуальної моделі - виявлення набору причинно-наслідкових зв'язків, облік яких необхідний для отримання необхідних результатів. Вона представляє загальний погляд на дані. Концептуальну модель бази геопросторових даних, що використовується в цій роботі для формування UBID подано у вигляді UML-діаграми (рис.3.1).

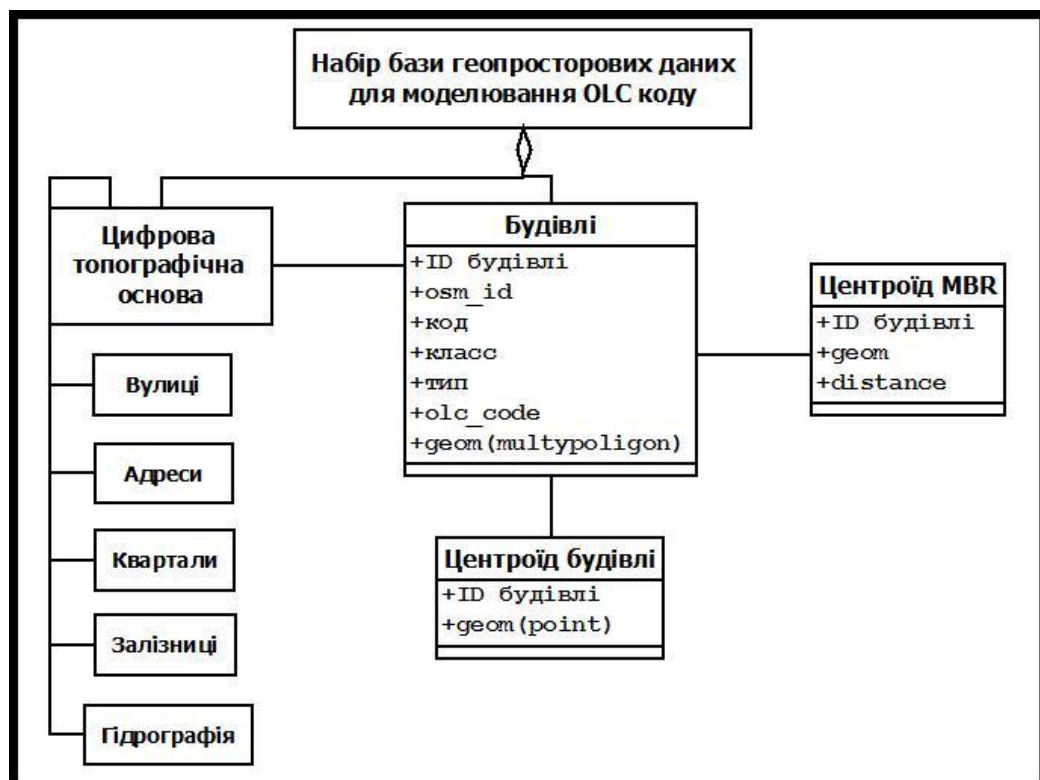


Рис 3.1. UML діаграма концептуальної моделі бази геопросторових даних для формування UBID на основі OLC коду

(MBR - мінімальний обмежувальний прямокутник (від англ. Minimum Bounding Rectangle))

В концептуальній моделі бази геопросторових даних для формування UBID на основі OLC визначено такі класи об'єктів цифрової топографічної основи тематичного картографування: вулиці; адреси; квартали; залізниці; гідрографія.

Крім вищезазначених об'єктів цифрової топографічної основи з узагальненими класами в концептуальній моделі БГД для формування UBID на основі коду OLC визначено моделі цільових об'єктів, що безпосередньо використовуються (створюються) при моделюванні, а саме:

- будівлі з атрибутами (ID будівлі, *osm_id*, код, клас, тип та *OLC_code*);
- центроїди будівель з атрибутом (ID будівлі + *geom* з типом *Point*);
- охоплюючі прямокутники будівель з атрибутом (ID будівлі + *geom* з типом *Polygon*);
- центроїди охоплюючого прямокутника з атрибутами (ID будівлі, відстань до центроїда будівлі (*distance*) + *geom* з типом *Point*).

3.2 Реалізація базової SQL функції для формування OLC коду в середовищі СКБД PostgreSQL

Аналіз засобів формування OLC коду (див. п. 1.2) показує, що відсутні реалізації функцій формування OLC коду, що забезпечують використання цієї технології безпосередньо в базах геопросторових даних разом з SQL як основною мовою сучасних СКБД. Далі стисло розглядаються призначення та засоби розроблення прикладних функцій постійного зберігання в сучасних СКБД, а також реалізація базової функції `_encode_olc(B,L)` на мові PG/PL SQL, яка забезпечує формування 10 значного OLC коду для заданої пари координат широти/довготи.

Основна мета функцій в програмуванні взагалі – спростити вирішення задачі за рахунок її декомпозиції на дрібніші підзадачі. Спрощення досягається за рахунок того, що про функції можна думати, абстрагувавшись від «великої» задачі. Для цього функція визначає чіткий інтерфейс із зовнішнім світом (параметри і значення результату). Її реалізація (тіло функції) може змінюватися; сторона, що викликає функцію, «не бачить» цих змін та не залежить від них. Цій ідеальній ситуації може заважати глобальний стан (глобальні змінні), і треба враховувати, що у випадку БД таким станом є вся база даних.[20]

У традиційних мовах функції часто об'єднуються в модулі, які мають власний інтерфейс і реалізацію. Межі модулів можуть проводитися більш-менш довільно. Для PostgreSQL є жорстка межа між клієнтською частиною і серверною: серверний код працює з базою, клієнтський - керує транзакціями.

Функції є об'єктами бази даних, разом з таблицями, видами та індексами. У PostgreSQL є велика кількість стандартних функцій, зокрема понад 1000 для опрацювання просторових даних. Крім цього, можна створювати і власні функції на різних мовах програмування, зокрема на SQL та PL/pgSQL. При необхідності змінити функцію файл редагується і виконується (за допомогою `psql` або засобами IDE). Якщо ж змінювати визначення функцій відразу в БД, зміни легко втратити.

Тіло функції містить команди на мові програмування, обраній у реченні LANGUAGE. Тіло задається у вигляді текстових рядків. Ці рядки зберігаються в системному каталозі та інтерпретуються кожен раз, коли функція викликається. Оскільки тіло функції з великою ймовірністю буде містити апострофи і лапки, зручно використовувати синтаксис укладення рядків в долари. При визначенні функції вказуються формальні параметри – вони доступні в тілі функції.

Для параметра можна задати значення за замовчуванням, яке буде використовуватися, якщо при виклику функції не вказати відповідний фактичний параметр.

Значення, що повертається можна визначити двома способами:

- використовувати окрему конструкцію RETURNS для вказівки типу,
- визначити вихідні параметри (з режимом OUT або INOUT).

Наприклад, функція із зазначенням RETURNS integer та функція з параметром OUT integer повертають ціле число. Можна використовувати і обидва способи одночасно. У цьому випадку функція також буде повертати одне ціле число. Але при цьому типи RETURNS і режими вихідних параметрів повинні бути узгоджені між собою.

Причиною існування вбудованих функцій є ефективність їх використання. Виклик звичайної функції виконається з деякою послідовністю настанов, пов'язаних з обробленням самого виклику, що містить занесення її аргументів у стік, чи поверненням їх з функції. У деяких випадках значна кількість циклів центрального процесора використовується саме для виконання цих дій.

Але, якщо функція вбудовується в рядок програми, то таких системних витрат просто немає, і загальна швидкість виконання програми зростає. Якщо ж вбудована функція виявляється великою, то загальний розмір програми може істотно збільшитися. Тому краще як вбудовані використовувати тільки маленькі функції, а ті, що є більшими, оформляти у вигляді звичайних функцій.

Функції дозволяють виконувати деякий код безпосередньо сервером бази даних. Ці функції можуть бути написані на SQL, який має деякі примітивні програмні оператори, такі як галуження та цикли. Але гнучкішою буде функція написана на одній із мов програмування, з якими PostgreSQL може працювати. До таких мов належить PL/SQL.

PL / SQL – це процедурна, блочно-структурована мова програмування, що є розширенням мови SQL СУБД Oracle. Вона надає ряд можливостей, які дозволяють створити великі, багатофункціональні програми для роботи з БД.

Мова програмування, що використовується для доступу до реляційних баз даних, називається мовою структурованих запитів (SQL, Structured Query Language). SQL – гнучка і ефективна мова, всі засоби якої служать для маніпулювання реляційними даними і для їх дослідження.[21]

PL / SQL означає Procedural Language / SQL (процедурна мова / SQL), розширює можливості SQL, додаючи в нього такі конструкції процедурних мов, як:

- змінні і типи даних (як попередньо визначені, так і обумовлені користувачами);
- керуючі структури, такі як умовні оператори та цикли;
- процедури і функції;
- об'єктні типи і методи.

Алфавіт мови включає такий набір символів:

- англійські букви верхнього і нижнього регістрів A..Z, a..z;
- арабські цифри 0..9;
- символи + - * / <> =; :. , ' ~! @ # \$% ^ & _ () {} [];
- символи табуляції, пробіли і символи повернення каретки.

Ідентифікатори мають довжину до тридцяти символів і складаються з великих і малих літер, цифр і знака підкреслення, причому перша повинна бути буква. Допускається, але не рекомендується використовувати спеціальні символи, такі як #, \$. Деякі з ідентифікаторів в мові PL / SQL мають спеціальне синтаксичне значення.

Структура програми представляє собою набір блоків PG/PL SQL, рекурсивно вкладених один в одного.

Структура блоку:

```
[<< мітка >>]
[DECLARE розділ оголошень]
[BEGIN виконуваний розділ]
[EXCEPTION розділ обробки виключень]
END [<< мітка >>];
```

Запити формуються за допомогою SQL, що створює в мережі велике число пересилань, по одному на кожен SQL-оператор (див. рис. 3.2). Однак кілька SQL-операторів можна об'єднати в один блок PG/PL SQL і послати їх серверу як єдине ціле. В результаті мережевий трафік знижується, і додаток функціонує набагато швидше.

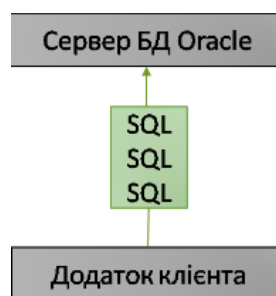


Рис. 3.2. Використання PG/PL SQL в середовищі клієнт-сервер

В роботі на мові PG/PL SQL реалізовано базову функцію *_ecode_olc(B, L)*, яка забезпечує формування 10 значного OLC коду для заданої пари координат широти/довготи. Ця функція застосовується в подальшому для реалізації загальної прикладної функції формування UBID будівель на основі координат

центроїдів їх охоплюючих прямокутників. Узагальнений алгоритм функції $_encode_olc(B, L)$ зображено на рис. 3.3.

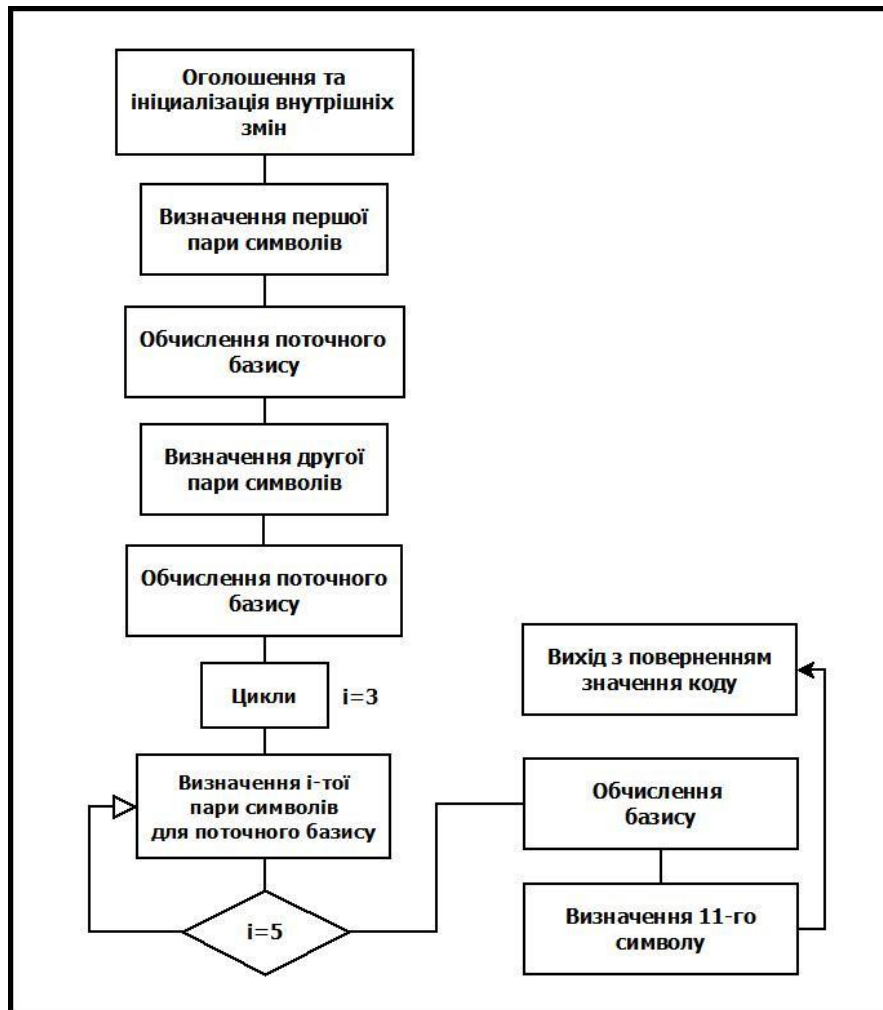


Рис.3.2. Схема алгоритму SQL функції формування OLC коду

Функція визначає 10-ти значний OLC код для точки із заданою широтою/довготою, де десятизначний OLC код з роздільником "+" обчислюється за п'ять кроків з використанням ієрархії координатних сіток з просторовим розрізненням: 20, 1, 0.05, 0.0025, 0.000125 градусів.

Коди для сіткових значень широти та довготи вибираються із наперед заданого упорядкованого набору символів:

{"2","3","4","5","6","7","8","9","C","F","G","H","J","M","P","Q","R","V","W","X"}

Повний текст функції приведено в лістингу нижче:

```

CREATE OR REPLACE FUNCTION _encode_olc(latitude double precision,longitude
double precision)
RETURNS text AS
$BODY$
DECLARE
-- оголошення результативного коду як масиву 11 символів
olc_code text[11];
-- оголошення масиву символів, що використовуються для кодування
s_code TEXT[:]='{"2","3","4","5","6","7","8","9","C","F","G","H","J","M","P","Q",
"R","V","W","X"}';
-- бази для обчислення індексів символів
base_lat double precision;
base_lon double precision;
-- індекси для вибору символів
i integer;
j integer;

BEGIN
-- визначення першої пари символів olc_code
base_lat=latitude + 90.0;
base_lon=longitude + 180.0;
i=floor(base_lat/20.0);
j=floor(base_lon/20.0);
olc_code [1]= s_code[i+1];
olc_code [2]= s_code[j+1];
-- визначення бази другої пари символів
base_lat=base_lat - 20.0*i;
base_lon=base_lon - 20.0*j;
i=floor(base_lat);
j=floor(base_lon);
записуємо другу пару символів в масив olc_code
olc_code [3]= s_code[i+1];
olc_code [4]= s_code[j+1];
-- визначення третьої пари символів
base_lat=base_lat - i;
base_lon=base_lon - j;
i=floor(base_lat/0.05);

```

```

j=floor(base_lon/0.05);
-- записуємо третю пару символів в масив olc_code
olc_code [5]= s_code[i+1];
olc_code [6]= s_code[j+1];
-- визначення четвертої пари символів
base_lat=base_lat - i*0.05;
base_lon=base_lon - j*0.05;
i=floor(base_lat/0.0025);
j=floor(base_lon/0.0025);
olc_code [7]= s_code[i+1];
olc_code [8]= s_code[j+1];
-- вставка роздільника "+" як дев'ятого символу в масив olc_code
olc_code[9]= '+';
-- визначення п'ятої пари символів
base_lat=base_lat - i*0.0025;
base_lon=base_lon - j*0.0025;
i=floor(base_lat/0.000125);
j=floor(base_lon/0.000125);
-- записуємо п'яту пару символів в масив olc_code
olc_code [10]= s_code[i+1];
olc_code [11]= s_code[j+1];
-- перетворення масиву символів OLC коду в рядок символів,
-- який повертається функцією як результат її роботи
return array_to_string (olc_code, "");
END;
$BODY$
LANGUAGE plpgsql VOLATILE STRICT
COST 100;
ALTER FUNCTION _encode_olc(double precision,double precision)
OWNER TO postgres;

```

Схему послідовності обчислювальних блоків зображено на рис. 3.3.

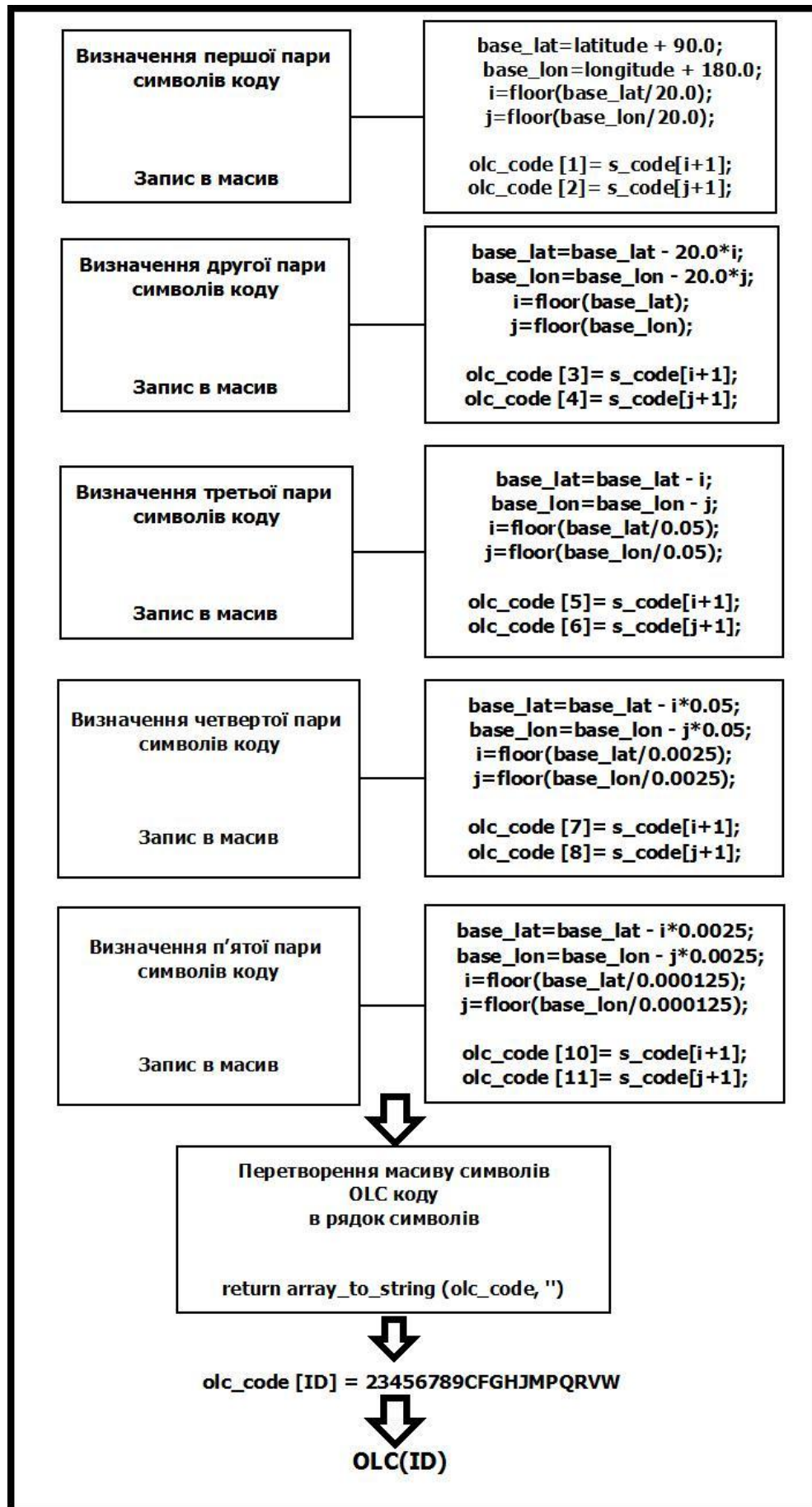


Рис.3.3 Схема етапів прикладної функції формування масиву OLC коду

Розглянемо докладніше зміст окремих блоків функції.

Блок оголошення типу функції та внутрішніх змінних:

Блок нормування вхідних координат для визначення OLC коду:

```
BEGIN
base_lat=latitude + 90.0;
base_lon=longitude + 180.0;
i=floor(base_lat/20.0);
j=floor(base_lon/20.0);
olc_code [1]= s_code[i+1];
olc_code [2]= s_code[j+1];
```

Далі визначаємо базу для обчислення індексів другої пари символів як залишок після першого кроку та індекси для другої пари символів.

```
base_lat=base_lat - 20.0*i;
base_lon=base_lon - 20.0*j;
i=floor(base_lat);
j=floor(base_lon);
```

Записуем другу пару символів в масив коду та аналогічно виконуємо обчислення для наступних пар символів.

По закінченню обчислення п'ятої пари символів та запису її в масив коду функція перетворення масиву символів OLC коду в рядок символів повертається як результат її роботи.

```
return array_to_string (olc_code, "");
END;
$BODY$
LANGUAGE plpgsql VOLATILE STRICT
COST 100;
ALTER FUNCTION _encode_olc(double precision,double precision)
OWNER TO postgres;
```

return array_to_string (olc_code) –перетворення масиву символів OLC коду в рядок символів як результат її роботи, що повертається в програму, що викликала функцію.

3.3 Реалізація прикладної функції формування UBID середовищі СКБД PostgreSQL/PostGIS

На мові PL/pgSQL реалізовано функцію `_olc_code_building()`, яка з використанням таблиці будівель `building` бази геопросторових даних визначає центроїди охоплюючих прямокутників та з використанням функції `_ecode_olc(B, L)` визначає OLC код для них, який і записується у відповідне поле таблиці `building` як `UBID` будівлі. Алгоритм функції приведено на рис. 3.4.

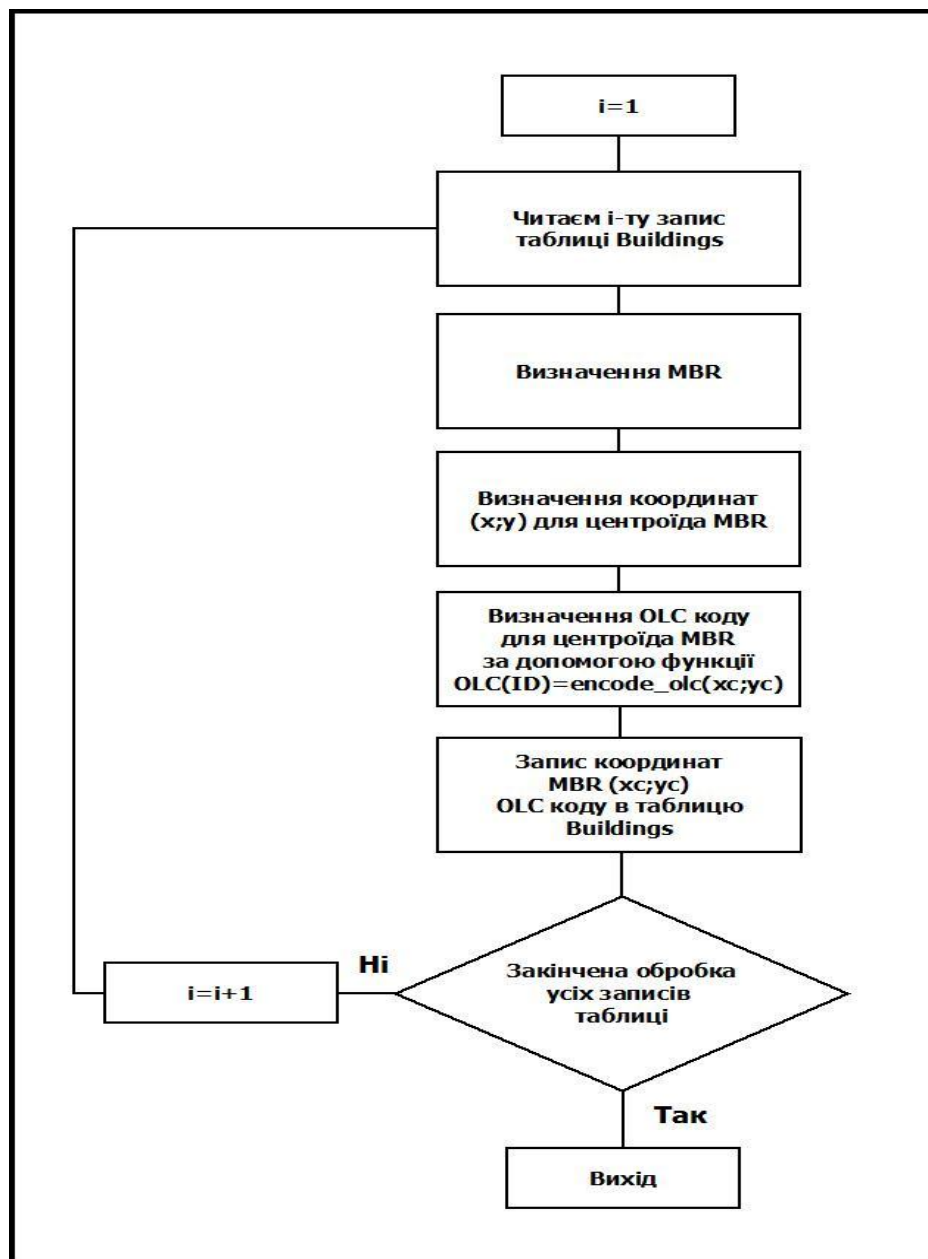


Рис. 3.4. Алгоритм реалізації прикладної функції формування UBID

Текст функції `_olc_code_building()`, що реалізує цей алгоритм приведено на листингу далі.

```

CREATE OR REPLACE FUNCTION _olc_code_building()
  RETURNS text AS
  $BODY$
  DECLARE
    i record;
  -- оголошення змінної для центроїду mbr будівлі
    mbrcntr geometry;
  BEGIN
  -- визначення центроїдів mbr будинків та оновлення поля olc_code будинків з
  --- використанням координат цих центроїдів та функції _ecode_olc()
    FOR i in (SELECT gid,osm_id FROM building)
      loop
        mbrcntr=(SELECT ST_Centroid(ST_Envelope(c.geom)) FROM (
          SELECT b.gid, b.geom
            FROM building as b
           WHERE b.gid=i.gid)as c);
        UPDATE building SET olc_code=_ecode_olc(ST_Y(mbrcntr),ST_X(mbrcntr))
          WHERE gid=i.gid;
      end loop;
    return 'Done';
  END;
  $BODY$
LANGUAGE plpgsql VOLATILE STRICT
COST 100;

```

Для просторових побудов у функції використано такі базові функції PostGIS:

PostGIS функції та типи даних параметрів	Стислий опис призначення функції
<code>ST_Envelope(geometry):: geometry</code>	Повертає геометрію охоплюючого прямокутника для заданої геометрії
<code>ST_Centroid(geometry):: geometry</code>	Повертає точку центроїду для заданої геометрії
<code>ST_Y(geometry):: double precision</code>	Повертають відповідні координати для точки, що подана як клас <code>geometry</code>
<code>ST_X(geometry):: double precision</code>	

Розділ 4

**РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ
ФОРМУВАННЯ UVID НА ОСНОВІ ВІДКРИТОГО КОДУ
МІСЦЕПОЛОЖЕННЯ OLC**

					ДИПЛОМНИЙ ПРОЕКТ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Виконав		Шкута Д.В.			Розроблення засобів формування унікальних ідентифікаторів будівель на основі відкритого коду місцеположення OLC в середовищі СКБД PostgreSQL	Літ.	Арк.	Аркушів
.		Лященко А.А.					62	29
Керівник		Лященко А.А.				ІПО КНУБА, група ГСТ-61		
Зав. каф.		Карпінський						

4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ ФОРМУВАННЯ UBID НА ОСНОВІ OLC КОДУ

4.1 Структурно-функціональна модель дослідної БГД для формування UBID

Для реалізації дослідної БГД (рис. 4.1) використовуються програмні продукти з відкритими кодами та безкоштовними ліцензіями, а саме:

інструментальна ГІС QGIS;

СКБД PostgreSQL з просторовим розширенням PostGIS;

pgShapeLoader - спеціальна утиліта для завантаження в базу геопросторових даних PostgreSQL/PostGIS векторних даних у форматі shp-файлів;

pgAdmin III – програма для адміністрування баз даних на основі СКБД PostgreSQL та середовище програмування і виконання SQL запитів.

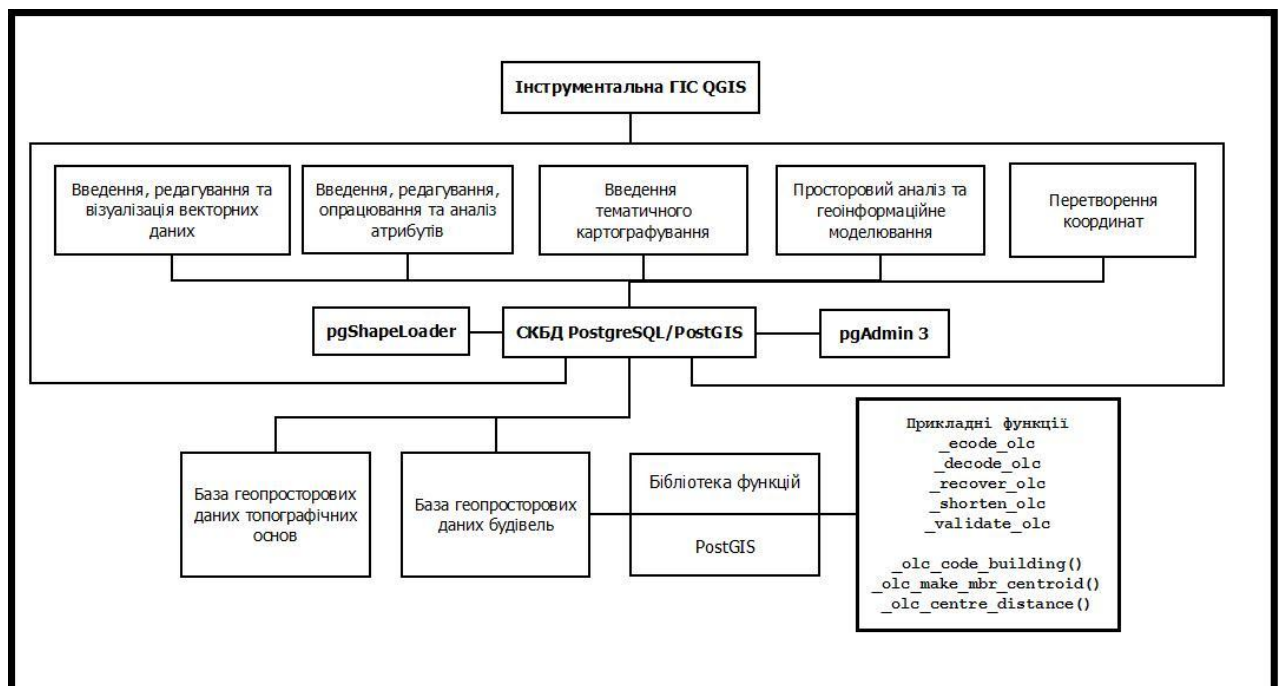


Рис. 4.1. Структурно-функціональна модель бази-геопросторових даних для формування UBID

Для швидкого та якісного вирішення будь-якої задачі, розробник повинен використовувати найкращий інструментарій. Такий набір

інструментів допоможе вирішити всі основні задачі, володіє зрозумілим інтерфейсом та широкими можливостями.

Сьогодні, в середовищі програмного забезпечення для геоінформаційних систем існує безліч різних рішень. Більшість програмного забезпечення (ПО) являє собою широкий набір інструментів для вирішення практично будь-якої задачі. Такі комплекси різнобічні, багаті функціями і надають розробнику різні механізми обробки даних. Такі системи можуть працювати з різними наборами даних в різних форматах.

В роботі для вирішення завдання буде використовуватись об'єктно-реляційна система керування базами даних (ОР СКБД) *PostgreSQL* з просторовим розширенням *PostGIS* та *pgShapeLoader* – завантажувача шейп-файлів в базу геопросторових даних PostgreSQL/PostGIS, а для візуалізації результатів аналізу – геоінформаційну систему *QGIS*. Усі програмні продукти належать до програм з відкритими кодами та розповсюджуються на умовах GNU General Public License.[24]

Вибір цих програмних продуктів для вирішення поставлених задач обґрунтовуємо такими чинниками:

- 1) вони належать до так званих відкритих вільних продуктів, а отже доступні для безоплатного використання будь-коли й будь-де;
- 2) PostgreSQL/PostGIS за повнотою реалізації стандартних геометричних типів та набором функцій займає друге місце (після Oracle) поміж чотирьох найпоширеніших СКБД (DB2, Oracle, MS SQL Server, PostgreSQL).

Розглянемо детальніше кожен з вище перерахованих програмних засобів, визначимо їх переваги та недоліки над іншими, обґрунтуємо вибір кожного з них.

Для візуалізації результатів просторового аналізу будемо використовувати систему QGIS, яка просто з'єднується з базою просторових даних на основі PostgreSQL/PostGIS як для візуалізації класів просторових об'єктів у векторних шарах електронної карти, так і для їх редагування.

QGIS (раніше відомий як «Quantum GIS») - повна настільна геоінформаційна система для роботи з геопросторовими даними і електронними картами, що підтримує багато форматів просторових даних та легко інтегрується з базами просторових даних в середовищі універсальних СКБД.

QGIS як і будь-яка повнофункціональна ГІС – це окремий світ інтерфейсів, функціональних можливостей з опрацювання й аналізу даних, редагування й створення нових векторних і растрових даних, роботи з шарами електронної карти, таблицями, друку та багато іншого.[9]

QGIS може оперувати даними в наступних форматах: GRASS GRID, ESRI Arc / Info Grid, ESRI E00 Files, USGS SRTM Grid, MOLA Grid, SRTM30 DEM, Images (GIF, JPEG, BMP, PNG, XPM, TIFF), GDAL raster formats, Gstat Shapes, ODBC tables. В цій системі дані можуть зберігатися як в текстовому файлі, так і в будь-якій базі даних яка підтримує формат SQL - наприклад, MySQL.

Quantum GIS дозволяє зберігати інформацію в більш широкому і популярному наборі форматів, при цьому використовується більш сучасна база даних для зберігання. Після того, як в систему заведений набір даних для аналізу, наступного найважливіший крок - коректна обробка. Система має можливості для побудови таких об'єктів, як точки, лінії, полігони. Всі ці об'єкти прив'язуються до певного інформаційного шару. З кожним об'єктом можна зв'язати дані, назву і положення (координати).

Простота роботи в QGIS пов'язана з її зручним графічним інтерфейсом. Інтерфейс QGIS (рис. 4.2) багато в чому стандартний і є кілька областей – робоча і найбільш інформативна область (карта об'єкта з полігонами) та службова на якій відображений провідник з усіма верствами і меню управління.

Для візуалізації та редагування класів просторових об'єктів у векторних шарах електронної карти QGIS пов'язують з базою геопросторових даних.

База геопросторових даних(БГД) - це база даних, яка оптимізована для зберігання та запиту даних, які являють собою об'єкти у геометричному просторі. Більшість просторових баз даних дозволяють відображати прості геометричні об'єкти, такі як: точки, лінії та багатокутники.

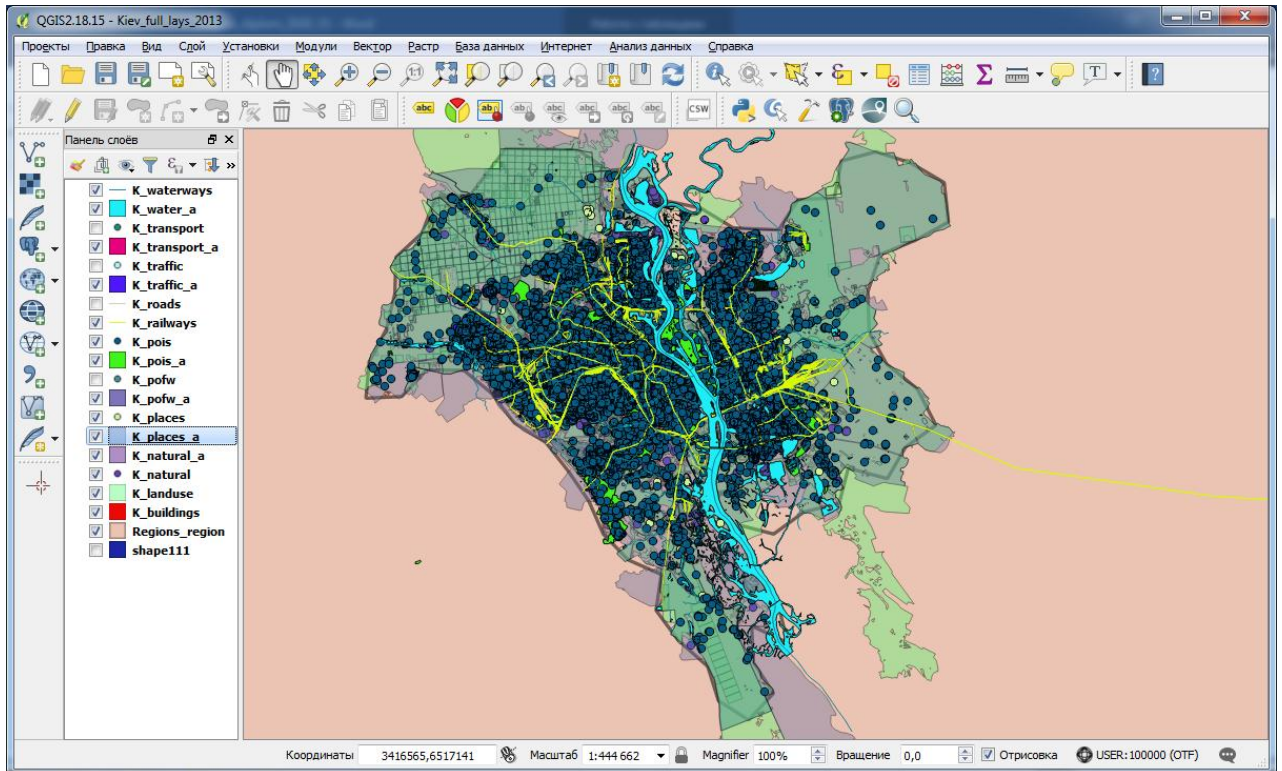


Рис. 4.2. Интерфейс QGIS

База даних містить абстрактні моделі реального світу і подає їх як визначену систему класифікації об'єктів та явищ. В базі забезпечено однозначну інтерпретацію абстрактних моделей комп'ютерними системами та їх користувачами, створено умови для розподіленого вироблення, поширення та використання геопросторових даних.

Для вирішення поставленої задачі обрано PostgreSQL з розширення PostGIS.

PostgreSQL - об'єктно-реляційна система керування базами даних (СКБД). Є альтернативою як комерційним СКБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СКБД з відкритим кодом (MySQL, Firebird, SQLite).[19]

Порівняно з іншими проектами з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СКБД та впроваджувати у неї найновіші досягнення.

Сервер PostgreSQL написаний на мові C. Зазвичай розповсюджується у вигляді набору текстових файлів із сирцевим кодом. Для інсталяції необхідно відкомпілювати файли на своєму комп'ютері і скопіювати в деякий каталог. Весь процес детально описаний та наведений нижче.

Створення бази геопросторових даних виконується з використанням додатку адміністрування баз даних *pgAdminIII* та завантажувача шейпфайлів *pgShapeLoader*.

pgAdminIII – це один із додатків-клієнтів для сервера PostgreSQL. PgAdmin є зручним засобом для адміністрування сервера PostgreSQL та створення баз даних, оскільки має всі необхідні для цього інструменти та інтуїтивно зрозумілий графічний інтерфейс.

В процесі роботи додаток pgAdminIII буде використовуватися як основний засіб доступу та роботи з базами даних на сервері СКБД PostgreSQL.

Інтерфейс або робоче вікно pgAdminIII зручний та простий у використанні, оскільки має всього три робочі зони.

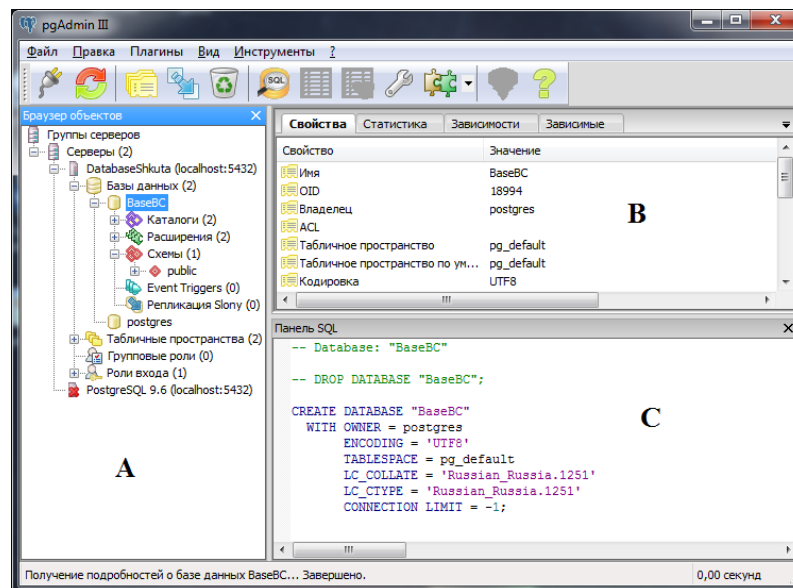


Рис. 4.3. Головне робоче вікно pgAdminIII

Зона А –навігатор об'єктів (Objects browser), в якому відображаються та вибираються різні об'єкти сервера і баз даних, наприклад: підключення до сервера, об'єкти сервера та баз даних – від схем баз даних і таблиць до окремих атрибутів і функцій;

Зона В – вікно властивостей поточного об'єкта, тобто вибраного в зоні А на дереві структури об'єктів сервера, баз даних, таблиць, окремих полів тощо;

Зона С – SQL-панель для відображення послідовності SQL операторів, за якими був створений поточний (вибраний в зоні А) об'єкт.

В процесі роботи будуть використовуватись такі основні операції pgAdminIII:



- Оновлення змісту навігатора об'єктів в зоні А. Застосовується після виконання операцій з базою даних (операторів SQL), якщо вміст навігатора не змінився автоматично.



- Перегляд властивостей поточного об'єкта бази даних.



- Створення нових об'єктів на сервері баз даних. Запускається процес в контексті поточного типу об'єкта, вибраного в зоні А основного вікна pgAdmin. В данній роботі будуть створюватись бази просторових даних і таблиці в них.



- Формування та виконання SQL операторів для маніпулювання об'єктами бази даних. Створюється окреме вікно з формами для інтерактивного введення й редагування послідовності SQL команд.



- Перегляд записів поточної таблиці зі значеннями атрибутів. Створюється окреме вікно із записами вибраної таблиці та набором інструментів для роботи з ними.

Додаток pgAdmin надає цілу низку інших корисних функцій з адміністрування даних, зокрема: реєстрація нових груп користувачів з їх ролями, резервне копіювання та відновлення баз даних, стиснення (VACUUM), аналіз (ANALYZE) і переіндексація (REINDEX) обраної бази даних або окремої таблиці.

Для завантаження шейп-файлів в просторову базу даних PostGIS в складі СКБД PostgreSQL/PostGIS постачається зручна утиліта *pgShapeLoader*.

Завантажувач *pgShapeLoader* для кожного вхідного шейп-файла автоматично виконує такі основні операції:

розпізнає тип класу просторових об'єктів у вихідному шейп-файлі;

створює таблицю бази даних з іменем та структурою атрибутів вихідного шейп-файла;

створює в цій таблиці спеціальний стовпчик *geom* з типом даних *geometry* або *geography* в залежності від вибраної користувачем системи координат;

реєструє в схемі бази даних параметри референцної системи координат із заданим користувачем ідентифікатором *SRID*;

завантажує атрибути та координатні описи просторових об'єктів вихідних шейп-файлів як записи створеної таблиці бази даних.

4.2 Результати обчислювального експерименту формування UBID для дослідних територій

4.2.1 Інформаційні ресурси та загальна схема обчислювального експерименту

Обчислювальний експеримент формування UBID виконано для будівель міст Києва та Біла Церква. Як вхідні дані для наповнення бази геопросторових даних використано набори векторних даних у форматі shp-файлів на територію зазначених міст, які завантажено із сайту відкритих геопросторових даних Open Street Map (OSM). Набори OSM даних містять шейп-файли таких класів об'єктів: будинки, використання земель, місцевості, ландшафти, вулиці та дороги, транспорт, залізниця та гідрографія (з відповідними назвами файлів :osm_buildings, osm_landuse, osm_places_a, osm_natural, osm_roads, osm_transport, osm_railways, osm_water, osm_waterways). Усі класи подано в координатах широта/довгота в системі геодезичних координат EPSG 4326 WGS-84 з індексом референцної системи координат SRID: EPSG 4326.

Для коректної картографічної візуалізації даних в QGIS передусім необхідно здійснити перетворення координат мінімум одного шару із ситеми координат SRID: EPSG 4326 в координати картографічної проекції, наприклад, проекцію Меркатора з SRID: EPSG 3395, WGS-84/World Mercator. Найкраще для цих цілей можна використати OSM шар *osm_places_a полігонального типу*, що містить, зокрема, межу міста та визначає просторове охоплення для усіх файлів OSM набору даних на території міста. Для такого перетворення завантажуюємо шар *osm_places_a* в QGIS та зберігаємо його копію на диску у файл *places_a_UTM* обравши для копії систему координат EPSG:3395, WGS-84/World Mercator. Створивши новий проект та долучивши першим до нього шар *places_a_UTM*, а потім всі інші шари з коректним налаштуванням атрибутів візуалізації шарів, ми отримаємо прийнятне зображення електронної карти набору даних міста (рис.4.4). Перепроціювання усіх шарів QGIS здійснює «на льоту», застосовуючи для цього проекцію першого шару, але без зміни системи координат власне в самих шейп-файлах.

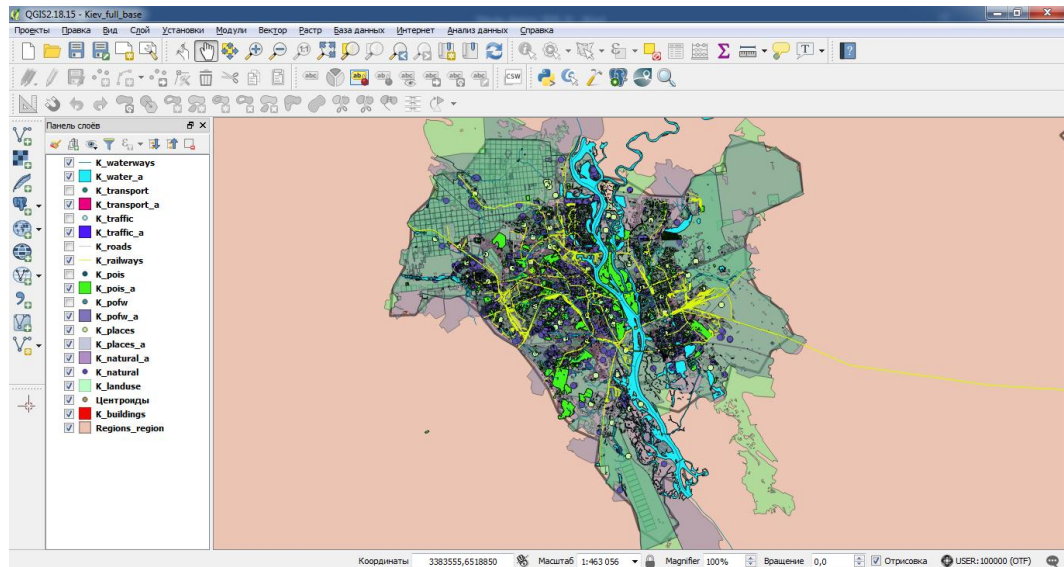


Рис. 4.4 Приклад карти Київa з доданими шарами в QGIS

Додавши всі шари до карти в шарі *building* відкриємо таблицю атрибутів та створимо нове поле *olc_code* з типом текст та довжиною 16 (у загальному випадку OLC коду складається з масиву 11-15 символів плюс роздільний знак).

Обчислювальний експеримент виконується за такими основними етапами:

1) ***підготовчий етап*** – попереднє опрацювання вихідних даних, доповнення їх потрібними атрибутами та створенням додаткових класів об'єктів, що необхідні для експерименту за такими кроками:

a) доповнення вхідного шару будинків додатковим атрибутом ***olc_code:string (16)*** для зберігання сформованого UBID на основі OLC коду;

b) створення шару ***build_centroid*** точкового типу, що містить центроїди для контурів будинків;

c) зберігання шару ***build_centroid*** у файл з іменем ***mbr_centroid*** як шаблону для побудови і зберігання координат центроїдів обмежувальних прямокутників будинків;

d) доповнення шару ***mbr_centroid*** додатковим атрибутом ***dist_ctr:real (9,5)*** для зберігання вимірів відстаней між центроїдами будинків та центроїдом MBR будинків;

е) зберігання шару *build_centroid* у файл з іменем *build_centroid_utm* в системі координат EPSG:3395, WGS-84/ World Mercator для коректного обчислення відстаней між центроїдами як значення атрибута *dist_centr*;

2) формування UBID на основі OLC коду для будівель дослідних територій – основний етап роботи на якому здійснюються:

а) створення в середовищі СКБД PostgreSQL/PostGIS бази геопросторових даних з цільовими шарами класів об'єктів, модифікованих і створених на підготовчому етапі;

б) завантаження та виконання SQL запитів для створення і зберігання в базі даних розроблених прикладних SQL функцій: *_ecode_olc (B,L)* як базової для формування OLC коду для заданої пари координат широта/довгота та спеціальної функції *_olc_code_building()* для формування UBID будівель з використанням базової функції *_ecode_olc (B,L)* за координатами центроїдів MBR будівель;

с) складання в QGIS електронних тематичних карт з візуалізацією будівель та сформованих UBID для них;

3) геоінформаційний аналіз використання OLC коду для формування UBID будівель – докладно розглядається в п.4.3 з такими етапами:

а) оновлення в таблиці бази даних *mbr_centroid* поля геометрії точкою центроїдів MBR будівель з використанням розробленої SQL-функції *_olc_make_mbr_centroid()*;

б) перепроєцювання шару *mbr_centroid* із системи координат WGS 84 в координати проекції Меркатора та зберігання результатів цієї операції в шарі з іменем *mbr_centroid_utm*;

с) визначення відстаней між центроїдами будівель (клас просторових об'єктів в таблиці *build_centroid_utm*) та центроїдами MBR будівель (клас просторових об'єктів в таблиці *mbr_centroid_utm*) з використанням розробленої SQL-функції *_olc_centr_distance()*;

д) побудова гістограм та статистичний аналіз збігу центроїдів будівель та центроїдів MBR будівель;

е) геоінформаційний аналіз просторового відношення належності центроїдів будівель та центроїдів MBR будівель контурам будівель та їх порівня;

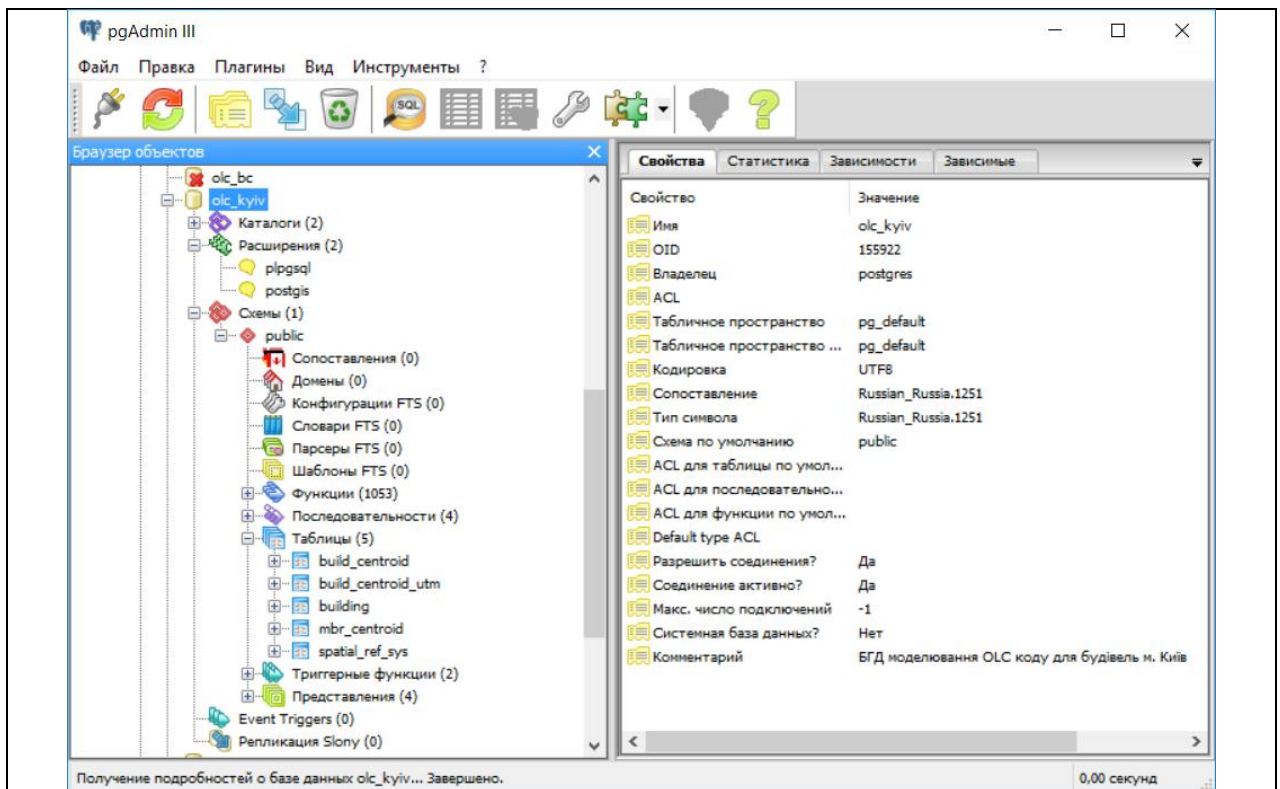
ф) статистичний аналіз показника часу виконання розроблених функцій на дослідних наборах геопросторових даних.

Зважаючи, що на підготовчому етапі виконуються практично елементарні операції з векторними даними з використанням базових функцій інструментальної ГІС QGIS, а зміст кроків очевидний з їх опису в пп. а) – е) переліку першого етапу, то далі докладно розглядаються лише кроки і результати основного етапу 2, а етап 3 докладніше розглянуто в п.4.3.

4.2.2 Результати формування UBID на основі OLC коду для будівель дослідних територій

Результати формування UBID на основі OLC коду для будівель дослідних територій докладно розглядаються за кожним кроком етапу формування UBID на прикладі міста Київ, а для міста Біла Церква подаються лише результати завершального кроку та картографічна візуалізація результату. Результати подаються у вигляді скрін-шотів екрану, які засвідчують виконання певного кроку та зміст його результату.

Крок 1: створення в середовищі СКБД PostgreSQL/PostGIS бази геопросторових даних з цільовими шарами класів об'єктів модифікованих і створених на підготовчому етапі:



Дерево об'єктів бази гепросторової даних `olc_kyiv` у екранній формі програми `pgAdmin III`, з таблицями цільових класів об'єктів, що використовуються для створення UBID на основі OLC коду.

	gid [PK] serial	osm_id character vary	code smallint	fclass character vary	name character vary	type character vary	olc_code character vary	geom geometry(Mult
1	1	4338650	1500	building	Льодовий ст	constructio		0106000020E
2	2	23118657	1500	building		apartments		0106000020E
3	3	23151565	1500	building		apartments		0106000020E
4	4	23151566	1500	building	ЖРЕО-417	apartments		0106000020E
5	5	23151569	1500	building		apartments		0106000020E
6	6	23151570	1500	building		apartments		0106000020E
7	7	23151577	1500	building		apartments		0106000020E
8	8	23151578	1500	building		apartments		0106000020E
9	9	23151579	1500	building		apartments		0106000020E
10	10	23151580	1500	building		apartments		0106000020E
11	11	23151581	1500	building		apartments		0106000020E
12	12	23151582	1500	building		apartments		0106000020E
13	13	23151584	1500	building		apartments		0106000020E
14	14	23151585	1500	building		apartments		0106000020E
15	15	23151586	1500	building		apartments		0106000020E
16	16	23152702	1500	building		apartments		0106000020E

Виведення вмісту таблиці `building` з незаповненими значеннями атрибута `olc_code`

Крок 2: Завантаження та реєстрація як об'єктів бази даних розроблених *SQL функцій* для формування UBID будівель з використанням OLC коду (зміст та алгоритми функцій розглянуто в п 3.2 та 3.3).

```

CREATE OR REPLACE FUNCTION _ecode_olc(latitude double precision,longitude double precision)
RETURNS text AS
$BODY$
DECLARE
-- оголошення результативного коду як масиву 11 символів
olc_code text[11];
-- оголошення з предустановкою масиву символів, що використовуються для кодування
a_code TEXT[] :=
{'2','3','4','5','6','7','8','9','C','E','G','H','J','M','P','Q','R','V','W','X'};
-- бази для обчислення індексів символів
base_lat double precision;
base_lon double precision;
-- індекси для вибору символів
i integer;
j integer;

BEGIN
-- визначення індексів для першої пари символів olc_code
base_lat=latitude + 90.0;
base_lon=longitude + 180.0;
i=floor(base_lat/20.0);

```

Запрос успешно выполнен без возвращаемых данных за 13 мс.

Форма програми pgAdmin III із SQL-кодом функції *_ecode_olc (B,L)* як базової для формування OLC коду для заданої пари координат широта/довгота

```

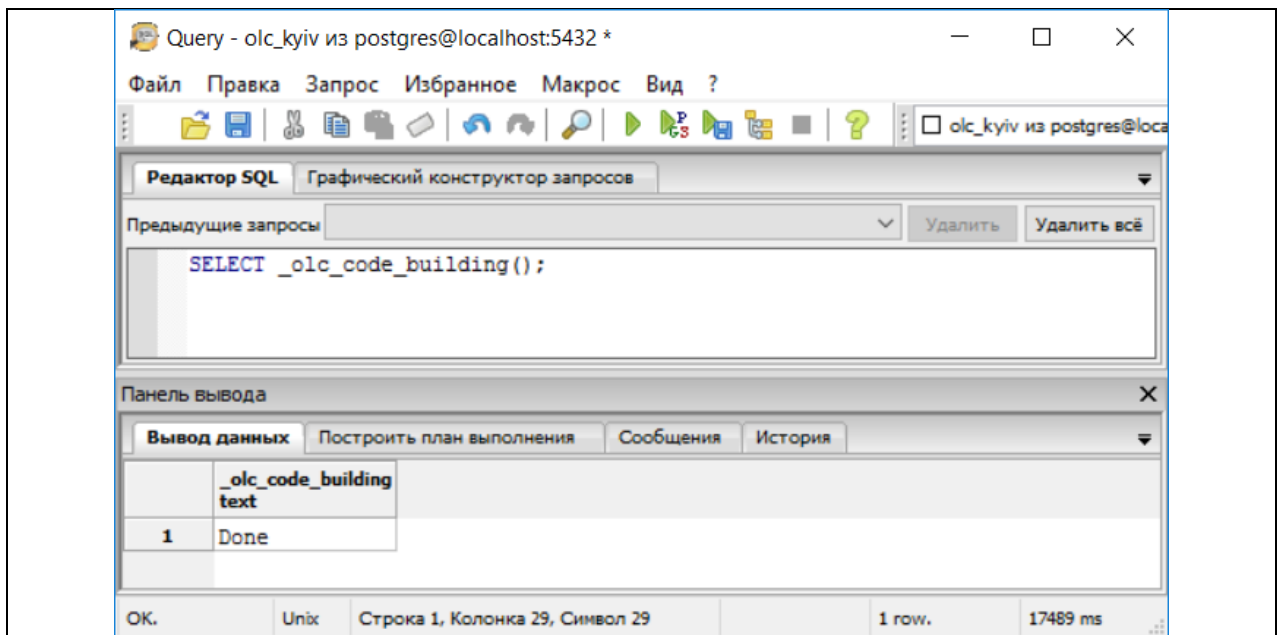
CREATE OR REPLACE FUNCTION _olc_code_building()
RETURNS text AS
$BODY$
DECLARE
i record;
mbrctr geometry;
BEGIN
--Вибірка будинків по черзі для визначення центроїдів mbr будинків
--та оновлення поля olc_code будинків з використанням координат mbr будинків
FOR i in (SELECT gid,osm_id FROM building)
loop
mbrctr=(SELECT ST_Centroid(ST_Envelope(c.geom)) FROM (
SELECT b.gid, b.geom
FROM building as b
WHERE b.gid=i.gid)as c);
UPDATE building SET olc_code=_ecode_olc(ST_Y(mbrctr),ST_X(mbrctr))
WHERE gid=i.gid;
end loop;
return 'Done';
END;

```

Запрос успешно выполнен без возвращаемых данных за 34 мс.

Форма програми pgAdmin III із SQL-кодом функції *_olc_code_building()* для формування UBID будівель з використанням базової функції *_ecode_olc (B,L)* за координатами центроїдів MBR будівель.

Виконання запиту *SELECT _olc_code_building()* для виклику функції формування OCL коду для координат центроїдів MBR будівель як їх ідентифікаторів UBID



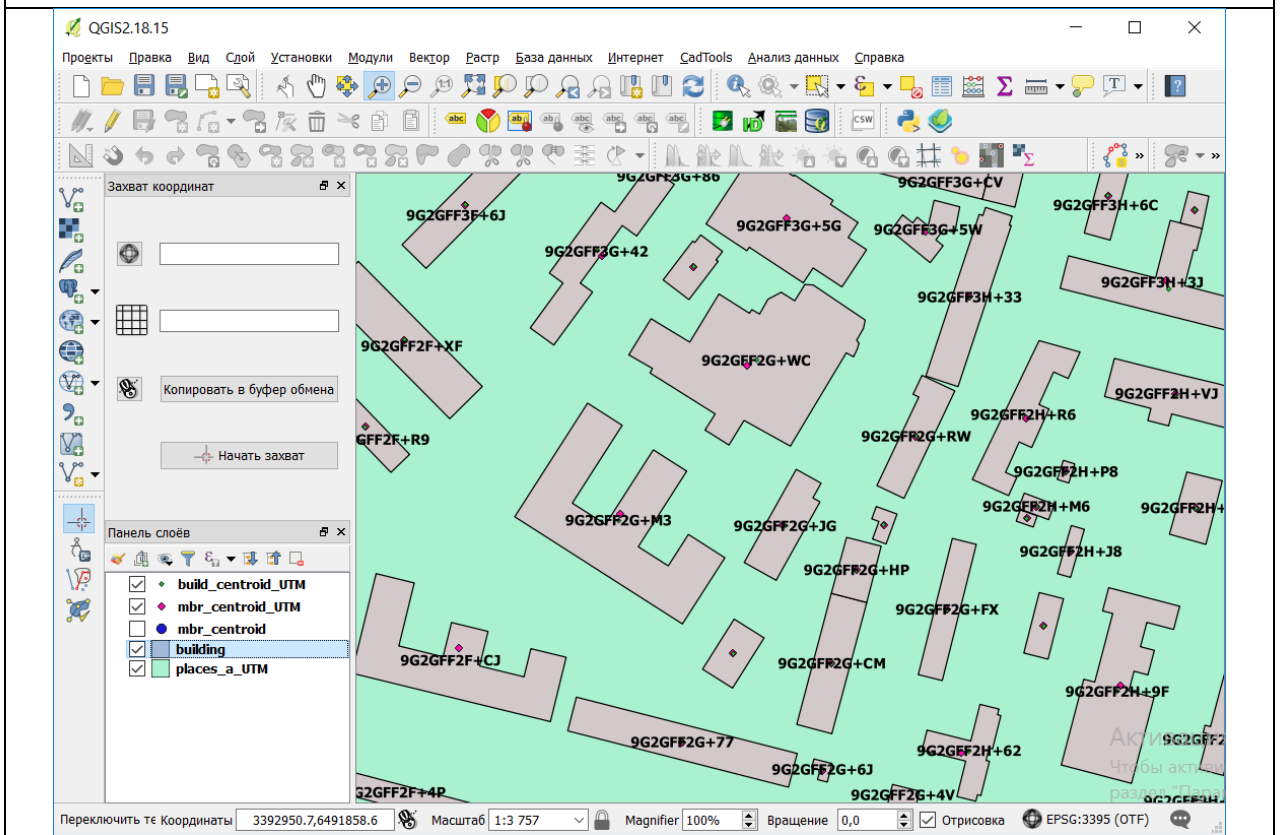
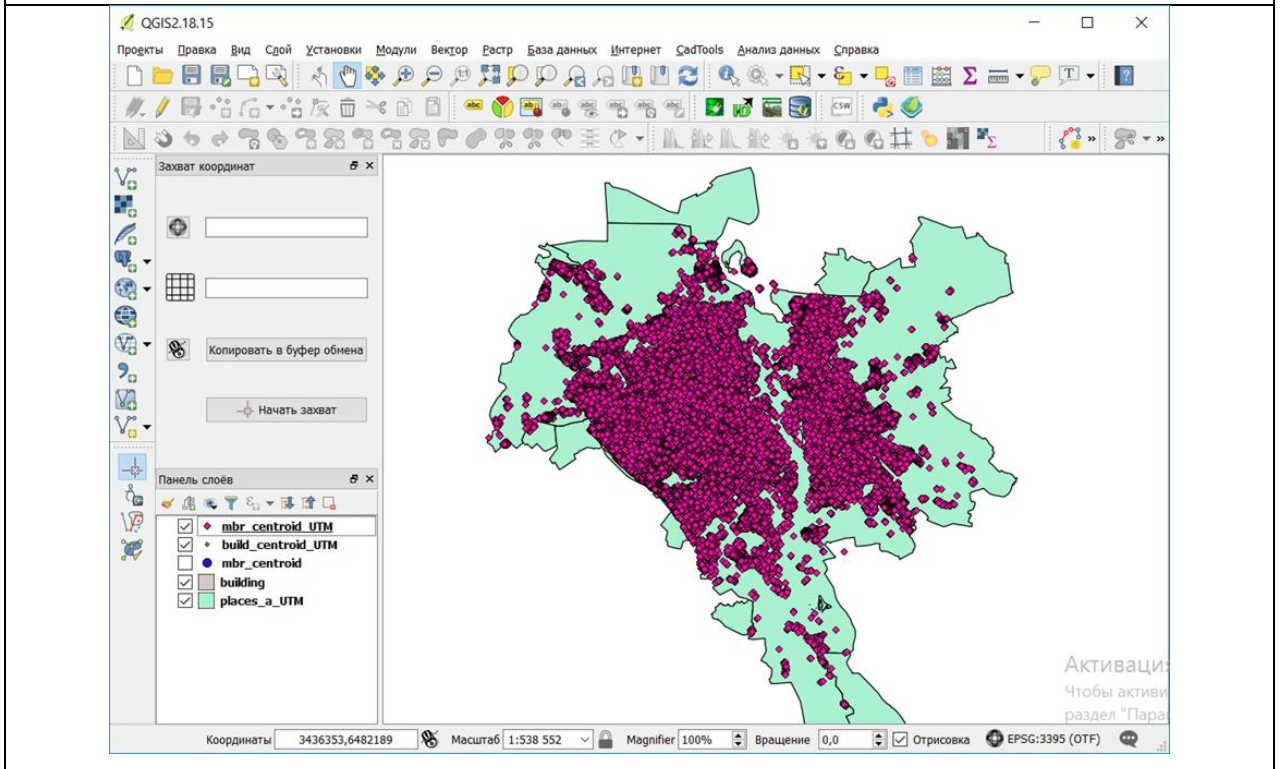
The screenshot shows a window titled "Редактирование данных - PostGIS (localhost:5432) - olc_kyiv - building". The menu bar includes "Файл", "Правка", "Вид", "Инструменты", and "?". The toolbar contains various icons for data manipulation. The main area displays a table with the following columns: `gid` (PK), `serial`, `osm_id`, `code`, `fclass`, `name`, `type`, `olc_code`, and `geom`. The table contains 16 rows of data. The `olc_code` column is highlighted in the first row.

gid	serial	osm_id	code	fclass	name	type	olc_code	geom
1	1	4338650	1500	buildi	Льодовий	construc	9G2G9FF8+JR	0106000020E61000000
2	2	23118657	1500	buildi		apartmen	9G2GCJPG+97	0106000020E61000000
3	3	23151565	1500	buildi		apartmen	9G2GCJPG+3C	0106000020E61000000
4	4	23151566	1500	buildi	ЖРЕО-417	apartmen	9G2GCJPG+HC	0106000020E61000000
5	5	23151569	1500	buildi		apartmen	9G2GCJPG+FP	0106000020E61000000
6	6	23151570	1500	buildi		apartmen	9G2GCJPG+5M	0106000020E61000000
7	7	23151577	1500	buildi		apartmen	9G2GCJPH+G5	0106000020E61000000
8	8	23151578	1500	buildi		apartmen	9G2GCJPH+RC	0106000020E61000000
9	9	23151579	1500	buildi		apartmen	9G2GCJPH+JW	0106000020E61000000
10	10	23151580	1500	buildi		apartmen	9G2GCJPG+X5	0106000020E61000000
11	11	23151581	1500	buildi		apartmen	9G2GCJQG+76	0106000020E61000000
12	12	23151582	1500	buildi		apartmen	9G2GCJQF+9W	0106000020E61000000
13	13	23151584	1500	buildi		apartmen	9G2GCJQF+5Q	0106000020E61000000
14	14	23151585	1500	buildi		apartmen	9G2GCJPF+VX	0106000020E61000000
15	15	23151586	1500	buildi		apartmen	9G2GCJQH+CC	0106000020E61000000
16	16	23152702	1500	buildi		apartmen	9G2GCJQG+8M	0106000020E61000000

The status bar at the bottom indicates "83210 строк." (83210 rows).

Форма програми pgAdmin III із фрагментом вмісту таблиці будівель із заповненим полем `olc_code` як результат роботи функції `_olc_code_building()`. **Всього для міста Києва опрацьовано 83 210 об'єктів за 17489 мілісекунд.**

Крок 3: складання в QGIS електронних тематичних карт з візуалізацією будівель та сформованих UBID для них для міста Києва



Фрагмент електронної карти з візуалізацією ідентифікатора UBID для будівель міста Києва

4.3 Геоінформаційний аналіз використання OLC коду для формування UBID для будівель на дослідних територіях

Основне завдання геоінформаційного аналізу використання OLC коду для формування UBID полягає в порівнянні рівня збігу координат центроїдів обмежувальних прямокутників будівель (MBR), які використовуються для визначення OLC кодів будівель, з центроїдами власне будівель, а також в аналізі належності центроїдів MBR та центроїдів будівель контурам відповідних будівель. Результати такого аналізу дозволять зробити висновок настільки OLC код або універсальний ідентифікатор UBID будівлі на його основі можна застосовувати не лише за призначенням (забезпечення універсальної унікальної ідентифікації будівель в наборах даних в різних джерелах даних та інформаційних системах), але й для визначення координати орієнтованого місцеположення будівлі. Вочевидь, якщо точки центроїду MBR належать області обмеженій контуром будівлі, то декодування OLC коду дозволить отримати з певною точністю координати чаруночки, що просторово належить множині точок об'єкта і її можна використовувати в деяких задачах просторового пошуку та геоінформаційного аналізу. Виконання сформульованого завдання геоінформаційний аналіз використання OLC коду для формування UBID будівель здійснюється в такій послідовності:

а) оновлення в таблиці бази даних *mbr_centroid* поля геометрії точкою центроїдів MBR будівель з використанням розробленої SQL-функції *_olc_make_mbr_centroid()*;

б) перепроєцювання шару *mbr_centroid* із системи координат WGS 84 в координати проекції Меркатора та зберігання результатів цієї операції в шарі з іменем *mbr_centroid_utm*;

с) визначення відстаней між центроїдами будівель (клас просторових об'єктів в таблиці *build_centroid_utm*) та центроїдами MBR будівель (клас

просторових об'єктів в таблиці `mbr_centroid_utm`) з використанням розробленої SQL-функції `_olc_centroid_distance()`;

d) побудова гістограм та статистичний аналіз збігу центрів будівель та центрів MBR будівель;

e) геоінформаційний аналіз просторового відношення належності центрів будівель та центрів MBR будівель контурам будівель та їх порівня;

f) статистичний аналіз показника часу виконання розроблених функцій на дослідних наборах геопросторових даних.

Розглянемо докладніше зміст перелічених етапів.

Необхідність оновлення в таблиці бази даних `mbr_centroid` поля геометрії точкою центрів MBR будівель з використанням розробленої SQL-функції `_olc_make_mbr_centroid()`, зумовлена тим, що інструментальні ГІС не мають окремого засобу побудови MBR об'єкта подібно інтерактивному засобу побудови центру як просторового класу точкового типу. Разом з цим, в PostGIS реалізовані відповідні SQL функції для виконання необхідних побудов. Для цього в рамках цієї роботи розроблено SQL функцію `_olc_make_mbr_centroid()`, текст якої наведено нижче:

```
CREATE OR REPLACE FUNCTION _olc_make_mbr_centroid()
  RETURNS text AS
  $BODY$
  DECLARE
  i record;
  mbrcntr geometry;
  BEGIN
  FOR i in (SELECT gid,osm_id FROM building)
    loop
    mbrcntr=(SELECT ST_Centroid(ST_Envelope(c.geom)) FROM (
    SELECT b.gid, b.osm_id, b.geom
    FROM building as b
    WHERE b.gid=i.gid)as c);
    UPDATE mbr_centroid SET geom=mbrcntr,dist_cntr=0.0 WHERE osm_id=i.osm_id;
    end loop;
  return 'Done';
  END;
  $BODY$
  LANGUAGE plpgsql VOLATILE STRICT COST 100;
```

Як вихідні дані у функції `_olc_make_mbr_centroid()` використовується таблиця `building` з просторовими даними контурів будівель, результат у вигляді центроїдів MBR контурів будівель зберігається в полі `geom` таблиці `mbr_centroid`, яка первинно на підготовчому етапі була сформована з використанням засобу створення центроїдів будівель. Таким чином у функції `_olc_make_mbr_centroid()` за допомогою функції PostGIS `ST_Envelope(geom)` будується MBR контуру будівлі, а функція `ST_Centroid(geom)` обчислює центроїд цього охоплюючого прямокутника. В SQL коді цій побудові відповідає один оператор:

```
mbrcntr=(SELECT ST_Centroid(ST_Envelope(c.geom)) FROM (.....))
```

Результатом побудови є точка центроїда MBR `mbrcntr`, яка й використовується для оновлення поля `geom` в таблиці `mbr_centroid`: з одночасною ініціалізацією нулем значення відстані між центроїдами `dist_cnr`:

```
UPDATE mbr_centroid SET geom=mbrcntr, dist_cnr=0.0.....
```

Для визначення відстаней між відповідними центроїдами будівель центроїдами MBR реалізована SQL-функція `_olc_centr_distance()`:

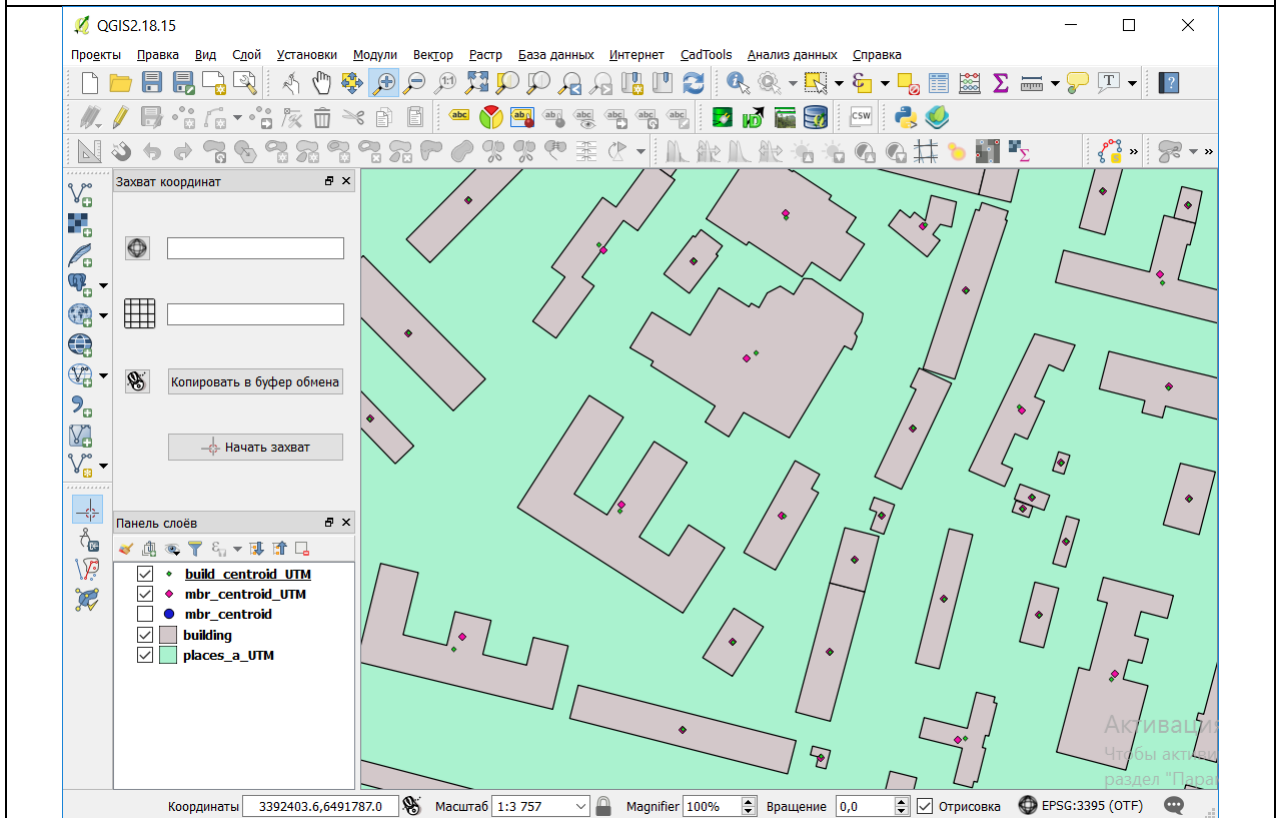
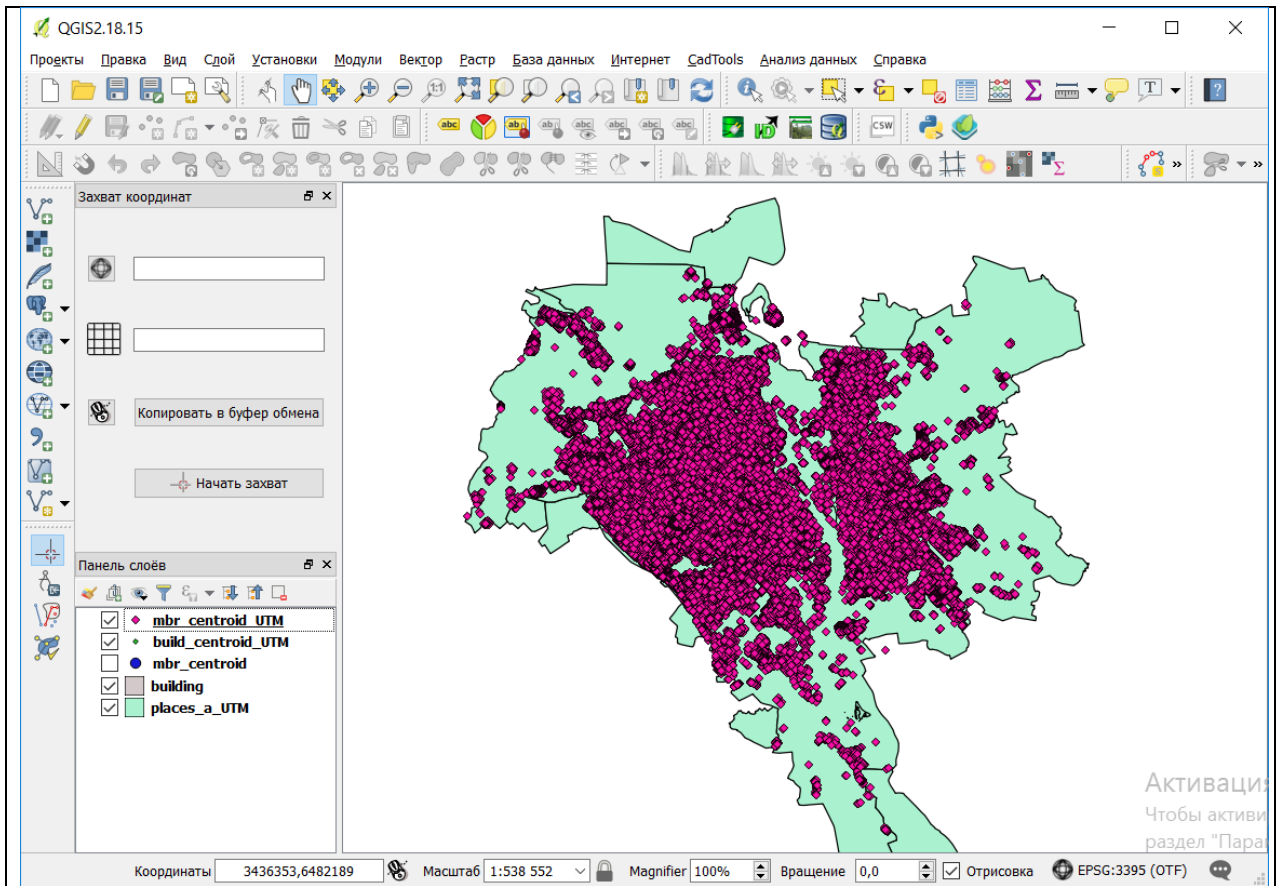
```
CREATE OR REPLACE FUNCTION _olc_centr_distance()
  RETURNS text AS
  $BODY$
  DECLARE
  i record;
  dist double precision;
  BEGIN
  FOR i in (SELECT gid,osm_id FROM mbr_centroid_utm)
  loop
  dist=(SELECT ST_Distance(b.geom,r.geom) FROM
  mbr_centroid_utm as b, build_centroid_utm as r
  WHERE b.gid=i.gid and r.osm_id=i.osm_id);
  -- оновлення дистанції в таблицях для mbr_centroid
  UPDATE mbr_centroid_utm SET dist_cnr=dist WHERE gid=i.gid;
  UPDATE mbr_centroid SET dist_cnr=dist WHERE osm_id=i.osm_id;
  end loop;
  return 'Done';
  END;
  $BODY$
  LANGUAGE plpgsql VOLATILE STRICT
  COST 100;
```

Для обчислення відстаней між центроїдами як вихідні дані у функції `_olc_centroid_distance()` використовуються таблиці `build_centroid_utm` та `mbr_centroid_utm`, що містять точки з координатами відповідних центроїдів в проекції Меркатора, оскільки пряме використання координат широти/довготи як вихідні дані для функції `ST_Distance(b.geom,r.geom)` поверне не коректний результат, тому відстань буде обчислено в градусній мірі, але без урахування кривизни земної поверхні. Результати обчислення відстаней наведено нижче.

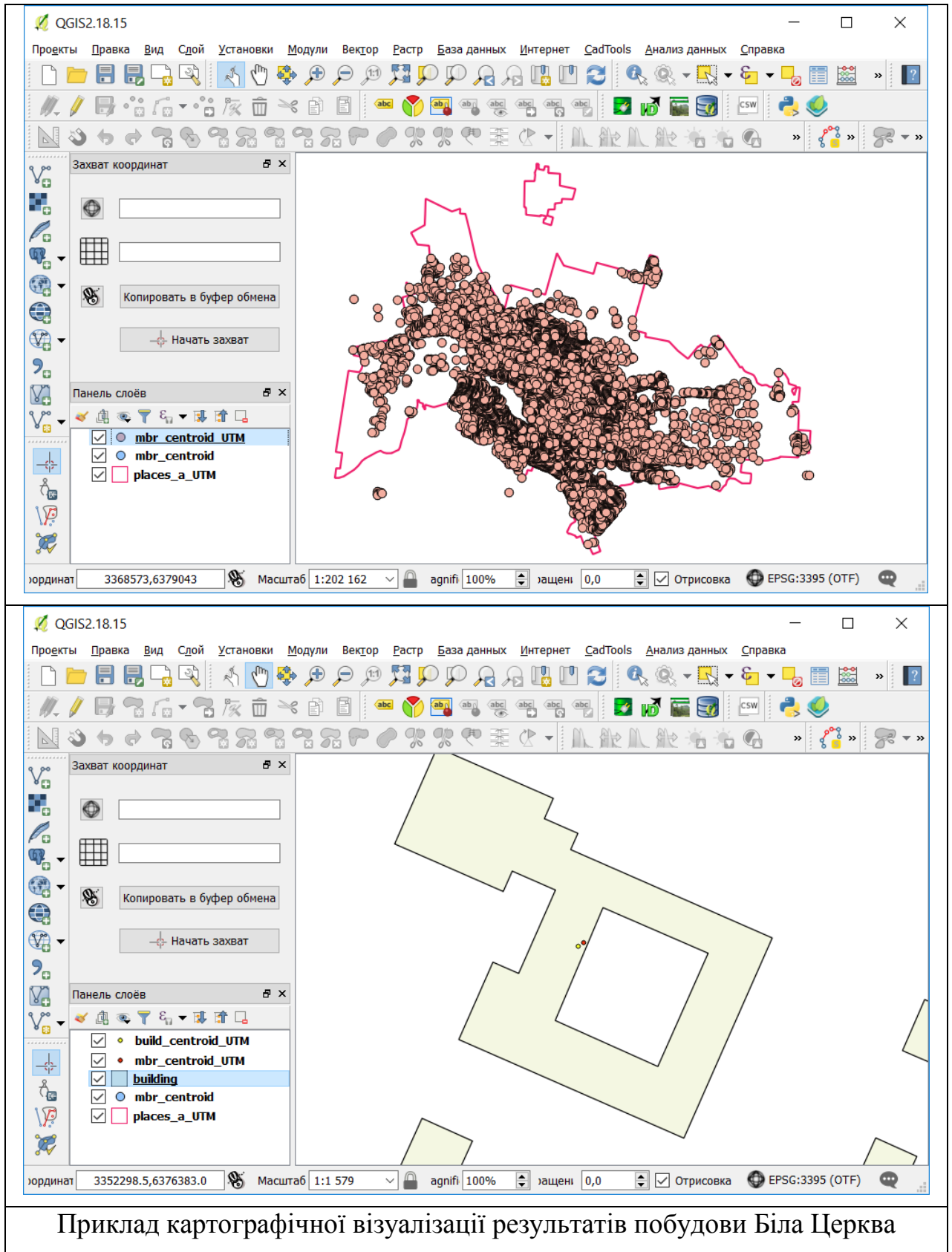
	gid	serial	_gid	osm_id	dist_cnr	geom
	[PK]		numeric(10,0)	character vary	numeric	geometry(Point,3395)
1	1	2	23118657	0	0101000020430D0000A82A22A5A7024A	
2	2	83	317987157	0.00436064359518592	0101000020430D00008C8324E776EC49	
3	3	252	23675151	0.00362597296978418	0101000020430D0000BEA52F0C4D004A	
4	4	317	23690948	0.104916842094779	0101000020430D0000D6EFD738DDE849	
5	5	378	23706766	1.43652302357875	0101000020430D000085AAEF7AC9D749	
6	6	379	23706770	1.66254804665041	0101000020430D000018E353A888D849	
7	7	380	23706776	23.0271917031579	0101000020430D0000AFA6A08BDD849	
8	8	381	23706780	3.32639684853807	0101000020430D0000354129BFA8D749	
9	9	382	23706792	7.18504265168556	0101000020430D000047EE2C20FCE749	
10	10	383	23706793	0.00528090836013119	0101000020430D00003C604D6390E749	
11	11	385	23707162	4.0398135527689	0101000020430D0000592B9E38ADEA49	
12	12	419	472344442	0.00000000104125029	0101000020430D0000AF491421DEE649	
13	13	450	675172217	0	0101000020430D00005420DE59CAD149	
14	14	692	24545724	0.00084986315241697	0101000020430D00009D3C5A56C3E449	
15	15	773	24696459	0	0101000020430D00009C0A67B52FEE49	
16	16	774	334867103	0.00343508934263023	0101000020430D0000E06F1AC385DF49	

	gid	serial	osm_id	dist_cnr	geom
	[PK]		character vary	numeric	geometr
1	1	88	820214586	0.16432947090582	010100
2	2	88	281664052	0.000000000931322574615479	010100
3	3	88	820214587	0.729164188257564	010100
4	4	88	281664054	0.00000000186264514923096	010100
5	5	88	820214588	0.233774558337335	010100
6	6	90	281664056	0	010100
7	7	91	820214589	0.232600024615142	010100
8	8	92	281664057	0.00218074626488977	010100
9	9	93	820214590	0.374677607755538	010100
10	10	94	281664059	0.00533958801136812	010100
11	11	95	820214591	0.589593628983003	010100
12	12	96	281664061	0.00315165675723588	010100
13	13	97	820214592	0.376590265175406	010100
14	14	98	281664062	0.00198150775751008	010100
15	15	99	820214593	0.746657913189234	010100
16	16	100	281664064	0.00417762489272498	010100
17	17	100	820214594	0.137193117084014	010100

Екранні форми таблиць `mbr_centroid` з результатами відстаней між центроїдами будівель та їх MBR для міст Київ і Біла Церква



Приклад картографічної візуалізації результатів побудови центрів для будівель та для MBR будівель міста Києва



Із картографічних зображень центрів можна зробити висновок про достатньо високий рівень збігу центрів будівель та центрів їх

охоплюючих прямокутників, але для кількісної оцінки цього збігу варто виконати статистичний аналіз відстаней між центроїдами. З цією метою для таблиць *mbr_centroid* наборів даних обох міст були виконані запити з обчисленням кількості об'єктів в різних інтервалах відстані між центроїдами.

Тексти запитів подано нижче:

Текст SQL запиту на отримання інтервального ряду розподілу кількості об'єктів за відстанню між їх центроїдами для міста Києва:

```
SELECT count (*), count (*)*100./83210. from mbr_centroid
where dist_cntr<0.25 and dist_cntr>=0.
UNION ALL
SELECT count (*), count (*)*100./83210. from mbr_centroid
where dist_cntr<0.5 and dist_cntr>=0.25
UNION ALL
SELECT count (*), count (*)*100./83210. from mbr_centroid
where dist_cntr<1.0 and dist_cntr>=0.5
UNION ALL
SELECT count (*), count (*)*100./83210. from mbr_centroid
where dist_cntr<2.0 and dist_cntr>=1.0
UNION ALL
SELECT count (*), count (*)*100./83210. from mbr_centroid
where dist_cntr<5.0 and dist_cntr>=2.0
UNION ALL
SELECT count (*), count (*)*100./83210. from mbr_centroid
where dist_cntr<10.0 and dist_cntr>=5.0
UNION ALL
SELECT count (*), count (*)*100./83210. from mbr_centroid
where dist_cntr<20.0 and dist_cntr>=10.0
UNION ALL
SELECT count (*), count (*)*100./83210. from mbr_centroid
where dist_cntr<50.0 and dist_cntr>=20.0
UNION ALL
SELECT count (*), count (*)*100./83210. from mbr_centroid
where dist_cntr<75.0 and dist_cntr>=50.0
UNION ALL
SELECT count (*), count (*)*100./83210. from mbr_centroid
where dist_cntr>=75.0;
```

**Текст SQL запиту на отримання інтервального ряду розподілу
кількості об'єктів за відстанню між їх центроїдами
для міста Біла Церква:**

```

SELECT count (*), count (*)*100./13610. from mbr_centroid
where dist_cntr<0.25 and dist_cntr>=0.
UNION ALL
SELECT count (*), count (*)*100./13610. from mbr_centroid
where dist_cntr<0.5 and dist_cntr>=0.25
UNION ALL
SELECT count (*), count (*)*100./13610. from mbr_centroid
where dist_cntr<1.0 and dist_cntr>=0.5
UNION ALL
SELECT count (*), count (*)*100./13610. from mbr_centroid
where dist_cntr<2.0 and dist_cntr>=1.0
UNION ALL
SELECT count (*), count (*)*100./13610. from mbr_centroid
where dist_cntr<5.0 and dist_cntr>=2.0
UNION ALL
SELECT count (*), count (*)*100./13610. from mbr_centroid
where dist_cntr<10.0 and dist_cntr>=5.0
UNION ALL
SELECT count (*), count (*)*100./13610. from mbr_centroid
where dist_cntr<20.0 and dist_cntr>=10.0
UNION ALL
SELECT count (*), count (*)*100./13610. from mbr_centroid
where dist_cntr<50.0 and dist_cntr>=20.0
UNION ALL
SELECT count (*), count (*)*100./13610. from mbr_centroid
where dist_cntr<75.0 and dist_cntr>=50.0
UNION ALL
SELECT count (*), count (*)*100./13610. from mbr_centroid
where dist_cntr>=75.0;

```

Результати запитів зведено в таблицях 4.1 та 4.2.

Таблиця 4.1

Статистика відстаней між центроїдами будинків та їх MBR для м. Києва

Номер інтервалу	Значення інтервалу		Число об'єктів	
	від	до	значення	%
1.	0	0.25	57678	69.37
2.	0.25	0.5	3325	4.00
3.	0.5	1.0	5953	7.15
4.	1.0	2.0	6455	7.76
5.	2.0	5.0	5764	6.93
6.	5.0	10.0	2336	2.81
7.	10.0	20.0	1184	1.42
8.	20.0	50.0	488	0.59
9.	50.0	75.0	24	0.029
10.	>75		3	0.004
Разом			83210	100

Таблиця 4.2

Статистика відстаней між центроїдами будинків та їх MBR для м. Біла Церква

Номер інтервалу	Значення інтервалу		Число об'єктів	
	від	до	значення	%
1.	0	0.25	11944	87.759
2.	0.25	0.5	272	1.998
3.	0.5	1.0	378	2.777
4.	1.0	2.0	379	2.785
5.	2.0	5.0	377	2.770
6.	5.0	10.0	141	1.036
7.	10.0	20.0	78	0.573
8.	20.0	50.0	38	0.279
9.	50.0	75.0	2	0.015
10.	>75		1	0.007
Разом			13610	100

Для геоінформаційного аналізу та співставлення належності центроїдів будівель та центроїдів MBR полігонам контурів будівель до таблиць будівель та центроїдів було виконано запити з просторовими предикатами на основі функції `st_within(geom1, geom2)` бібліотеки PostGIS, яка перевіряє чи розташовані об'єкти з геометрією `geom1` повністю всередині області з геометрією `geom2`. Тексти відповідних запитів для міст Київ та Біла Церква подано нижче.

Текст SQL запиту на отримання кількості центроїдів, що повністю належать /не належать контурам відповідних будівель для міста Києва:
SELECT count (*), count (*)*100./83210. from mbr_centroid as c, building as b where st_within(c.geom, b.geom)='t' and c.osm_id=b.osm_id UNION ALL SELECT count (*), count (*)*100./83210. from build_centroid as bc, building as b where st_within(bc.geom, b.geom)= 't' and bc.osm_id=b.osm_id UNION ALL SELECT count (*), count (*)*100./83210. from mbr_centroid as c, building as b where st_within(c.geom, b.geom)='f' and c.osm_id=b.osm_id UNION ALL SELECT count (*), count (*)*100./83210. from build_centroid as bc, building as b where st_within(bc.geom, b.geom)= 'f' and bc.osm_id=b.osm_id;
Текст SQL запиту на отримання кількості центроїдів, що повністю належать /не належать контурам відповідних будівель для міста Біла Церква:
SELECT count (*), count (*)*100./13610. from mbr_centroid as c, building as b where st_within(c.geom, b.geom)='t' and c.osm_id=b.osm_id UNION ALL SELECT count (*), count (*)*100./13610. from build_centroid as bc, building as b where st_within(bc.geom, b.geom)= 't' and bc.osm_id=b.osm_id UNION ALL SELECT count (*), count (*)*100./13610. from mbr_centroid as c, building as b where st_within(c.geom, b.geom)='f' and c.osm_id=b.osm_id UNION ALL SELECT count (*), count (*)*100./13610. from build_centroid as bc, building as b where st_within(bc.geom, b.geom)= 'f' and bc.osm_id=b.osm_id;

Результати виконання запитів щодо топологічних відношень цетроїдів будівель та центроїдів MBR будівель відносно відповідних полігонів контурів будівель подано у формах нижче та узагальнено в табл. 4.3.

Query - olc_kyiv из postgres@localhost:5432 -

Файл Правка Запрос Избранное Макрос Вид ?

Редактор SQL Графический конструктор запросов

Предыдущие запросы Удалить Удалить все

```

SELECT count (*), count (*)*100./83210. from mbr_centroid as c, building as b
where st_within(c.geom, b.geom)='t' and c.osm_id=b.osm_id
UNION ALL
SELECT count (*), count (*)*100./83210. from build_centroid as bc, building as b
where st_within(bc.geom, b.geom)= 't' and bc.osm_id=b.osm_id
UNION ALL
SELECT count (*), count (*)*100./83210. from mbr_centroid as c, building as b
where st_within(c.geom, b.geom)='f' and c.osm_id=b.osm_id
UNION ALL
SELECT count (*), count (*)*100./83210. from build_centroid as bc, building as b
where st_within(bc.geom, b.geom)= 'f' and bc.osm_id=b.osm_id;

```

Панель вывода

Вывод данных Построить план выполнения Сообщения История

	count bigint	?column? numeric
1	80138	96.308136041341
2	81144	97.517125345511
3	3072	3.6918639586588
4	2066	2.4828746544886

OK. Unix Строка 11, Колонка 62, Символ 595 4 rows. 1536 ms

а)

Query - olc_bc из postgres@localhost:5432 -

Файл Правка Запрос Избранное Макрос Вид ?

Редактор SQL Графический конструктор запросов

Предыдущие запросы Удалить Удалить все

```

SELECT count (*), count (*)*100./13610. from mbr_centroid as c, building as b
where st_within(c.geom, b.geom)='t' and c.osm_id=b.osm_id
UNION ALL
SELECT count (*), count (*)*100./13610. from build_centroid as bc, building as b
where st_within(bc.geom, b.geom)= 't' and bc.osm_id=b.osm_id
UNION ALL
SELECT count (*), count (*)*100./13610. from mbr_centroid as c, building as b
where st_within(c.geom, b.geom)='f' and c.osm_id=b.osm_id
UNION ALL
SELECT count (*), count (*)*100./13610. from build_centroid as bc, building as b
where st_within(bc.geom, b.geom)= 'f' and bc.osm_id=b.osm_id;

```

Панель вывода

Вывод данных Построить план выполнения Сообщения История

	count bigint	?column? numeric
1	13394	98.4129316678912564
2	13394	98.4129316678912564
3	216	1.5870683321087436
4	216	1.5870683321087436

OK. Unix Строка 6, Колонка 10, Символ 302 4 rows. 184 ms

б)

Результати запиту щодо топологічних відношень цетроїдів будівель та центроїдів MBR будівель відносно відповідних полігонів контурів будівель:

а) для міста Києва; б) для міста Біла Церква

Таблиця 4.3

Результати статистики належності центроїдів контурам будівлі

Місто	Кількість центроїдів, що належать контуру будівлі			
	Центроїд будівлі		Центроїд MBR	
	кількість	%	кількість	%
Київ	81144	97,52	80138	96,31
Біла Церква	13394	98,41	13394	98,41

Результати статистики належності та не належності центроїдів MBR будинків та центроїдів будинків відповідним контурам будинків дають підстави констатувати майже повне співпадіння показників для обох типів центроїдів.

Оскільки одним із предметів роботи була реалізація SQL функцій постійного зберігання для формування ідентифікаторів будівель UBID на основі OLC коду та їх геоінформаційного і статистичного аналізу, то в табл. 4.4 приведемо показники часу виконання розроблених функцій для дослідних наборів геопросторових даних.

Таблиця 4.4

Час виконання реалізованих SQL функцій для дослідних наборів геопросторових даних

Місто набору даних (число об'єктів в наборі)	Час виконання реалізованих SQL функцій		
	<i>_olc_code_building()</i> для формування OLC коду	<i>_olc_make_mbr_centroid()</i> для визначення MBR будівель та їх центроїдів	<i>_olc_centroid_distance()</i> для визначення відстані між центроїдами
1	2	3	4
Київ (83 210)	19.936 сек	10.811 сек (20.523 хв) *	2.563 хв (39.152 хв) *
Біла Церква (13 610)	2. 651 сек.	1.629 сек (32, 527 сек) *	2.891 сек (1.074 хв) *

Примітка до табл. 4.4: зірочкою (*) позначено час виконання функцій в базі даних без додаткових індексів для значень поля *osm_id* в таблицях *building*, *buil_centroid*, *buil_centroid_utm* та *mbr_centroid_utm*.

Дані в табл. 4.4 свідчать про прийнятну швидкодію реалізованих SQL функцій для формування ідентифікаторів будівель UBID на основі OLC коду та їх геоінформаційного і статистичного аналізу.

Значення у стовпчиках 3 та 4 табл. 4.4, що в дужках та помічені зірочкою, це час виконання функцій до створення в базі даних додаткових індексів для значень поля `osm_id` в таблицях `building`, `buil_centroid_utm` та `mbr_centroid_utm`.

При завантаженні шейп-файлів векторних даних в таблиці СКБД PostgreSQL автоматично за замовчуванням створюється поле системної ідентифікації записів *gid* та відповідний індекс для значень цього поля. Для просторових даних за замовчуванням створюється просторовий індекс типу R-дерево. У функції `_olc_code_building()` використовується лише одна таблиця `building` та значення поля системного ідентифікатора як параметру циклу та оновлення записів таблиці в процесі визначення OLC коду будівель.

У функціях `_olc_make_mbr_centroid()` та `_olc_centr_distance()` використовується одночасно кілька таблиць, а для оновлення записів обчисленими значеннями як умову необхідно використовувати поле `osm_id` для значень якого індекс за замовчуванням, не створюється. Цим пояснюється досить великі значення часу роботи цих функцій. Для набору даних для будівель м. Києва витрати часу складають понад 20 та 30 хв.

Для прискорення виконання функцій доцільно створити індекси для таблиць по стовпчику `osm_id`, який є у всіх таблицях та використовується у запитах як атрибут для встановлення відповідності між записами різних таблиць. В PostgreSQL для полів із числовими та символічними типами даних, за замовчуванням використовуються індекси типу В-дерево. Для цього в програмі pgAdminIII виконуємо таку послідовність запитів на створення індексів для таблиць нашої дослідної бази даних:

```
CREATE UNIQUE INDEX b_osm_idx ON building (osm_id);
CREATE UNIQUE INDEX bc_osm_idx ON buil_centroid (osm_id);
CREATE UNIQUE INDEX bcu_osm_idx ON buil_centroid_utm (osm_id);
```

```
CREATE UNIQUE INDEX m_osm_idx ON mbr_centroid (osm_id);  
CREATE UNIQUE INDEX mu_osm_idx ON mbr_centroid_utm (osm_id);
```

Як видно із табл. 4.4 після створення додаткових індексів в базі даних дослідних наборів за значенням поля *osm_id*, **час виконання функцій зменшився у 22 -150 разів**. Це ще раз підтверджує доцільність використовувати механізм індексування в базах даних для полів, які часто використовуються в умовах запитів.

ВИСНОВКИ

Відсутність стандартизованого способу ідентифікації будівель призводить до того, що важко точно пов'язати дані з конкретним об'єктом, створити бар'єр для ефективного управління активами, виконувати дослідження та аналізувати.

В даний час щоб визначити своє місцеположення або місцеположення якогось об'єкту використовують географічні координати, широту та довготу.

Відсутність стандартизованого способу ідентифікації будівель призводить до того, що важко інтегрувати дані з різних джерел та створює бар'єри для ефективного управління активами або виконання досліджень.

Унікальна система ідентифікації будівель на основі OLC коду забезпечує умови для інтегрування даних про будівлі з різних джерел

SQL функція формування UBID на основі OLC реалізована на мові PL/pg SQL, що забезпечує її незалежність від інструментальної ГІС та можливість використання для наборів геопросторових даних, що зберігаються в базі геопросторових даних ОР СКБД PostgreSQL/PostGIS.

Результати статистики належності та неналежності центроїдів MBR та центроїдів будинків відповідним контурам будинків дають підстави констатувати майже повне співпадіння показників для обох типів центроїдів.

Використання центроїдів MBR для формування унікальних ідентифікаторів просторових об'єктів на основі OLC коду створює передумови застосування цього методу також і для об'єктів з лінійним типом просторової локалізації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Постанова КМУ від 1 липня 2020 р. № 559 Про реалізацію експериментального проекту щодо запровадження першої черги Єдиної державної електронної системи у сфері будівництва. [Електронний ресурс] – Режим доступу: <https://zakon.rada.gov.ua/laws/show/559-2020-%D0%BF#Text>
2. Закон України Про регулювання містобудівної діяльності Стаття 22.Єдина державна електронна система у сфері будівництва. [Електронний ресурс] – Режим доступу: <https://zakon.rada.gov.ua/laws/show/3038-17#Text>
3. Закон України Про будівельні норми. [Електронний ресурс] – Режим доступу: <https://zakon.rada.gov.ua/laws/show/1704-17#Text>
4. Геоінформаційні технології та інфраструктура просторових даних: у шести томах. Том 2: Системи керування базами геоданих для інфраструктури просторових даних. Навчальний посібник. / Кейк Д., Лященко А.А., Путренко В.В., Хмелевський Ю., Дорошенко К.С., Говоров М. - К.: Планета-Прінт, 2017. 456 с..
5. GUID [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/GUID>
6. UUID [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/UUID>
7. Бази геопросторових даних. Частина 1: Основи роботи з геопросторовими даними в середовищі СКБД PostgreSQL/PostGIS: Методичні вказівки до виконання лабораторних робіт / уклад. : А. А. Лященко. – К. : КНУБА, 2019. – 98 с.
8. SOFTPRO: Містобудівний кадастр. Введення бази даних інформаційних ресурсів єдиної цифрової топографічної основи. [Електронний ресурс] – Режим доступу: <http://cadastre.com.ua/ua/vedennja-bazi-danih-informacinih-resursiv-edinoii-cifrovoi-topografichnoii-osnovi>

9. Основы QGIS. / Свидзинская Д. В., Бруй А. С.– К.: КНУ ім. Т.Г. Шевченка. – 2011. –8 6 с. [Електронний ресурс] – Режим доступу: <https://docplayer.ru/25808843-D-v-svidzinskaya-a-s-bruy-osnovy-qgis.html>
10. Evaluation of Location Encoding Systems [Електронний ресурс] – Режим доступу: <https://github.com/google/open-location-code/wiki/Evaluation-of-Location-Encoding-Systems>
11. Open Location Code Specification [Електронний ресурс] – Режим доступу: <https://github.com/google/open-location-code/blob/master/docs/specification.md>
12. Google Open Source Blog [Електронний ресурс] – Режим доступу: <https://opensource.googleblog.com/2015/04/open-location-code-addresses-for.html>
13. Plus code [Електронний ресурс] – Режим доступу: <https://plus.codes/benefits>
14. Community-led development «The Apache Way» [Електронний ресурс] – Режим доступу: <http://www.apache.org/licenses/LICENSE-2.0>
15. Open Location Code [Електронний ресурс] – Режим доступу: https://en.wikipedia.org/wiki/Open_Location_Code
16. Plus code [Електронний ресурс] – Режим доступу: <https://maps.google.com/pluscodes>
17. Geohash Vs OLC:Evaluating the geocoding techniques in adtech/Kartik Bhanot 10.07.2019 [Електронний ресурс] – Режим доступу: <https://medium.com/miq-tech-and-analytics/geohash-vs-open-location-codes-understanding-the-two-geocoding-techniques-in-advertising-space-8002452201c8>
18. Постанова КМУ від 4 вересня 2013р. № 661 Київ про затвердження порядку загальнодержавного топографічного і тематичного картографування. [Електронний ресурс] – Режим доступу: <https://zakon.rada.gov.ua/laws/show/661-2013-%D0%BF#Text>
19. PostgreSQL [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/PostgreSQL>

20. SQL функції / Postgres Professional, 2017 год / Егор Рогов, Павел Лузанов/ Переклад українською та адаптацією до баз геопросторових даних Лященко А.А., Київ, КНУБА, 2017.

21. Основи мови PL/SQL. [Електронний ресурс] – Режим доступу: https://elearning.sumdu.edu.ua/free_content/lectured:a8104441b8e00905159c1ff04257b014dd456247/20151109195846/164783/index.html

22. Unique Building Identifier: A natural key for building data matching and its energy applications [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/329660923_Unique_Building_Identifier_A_natural_key_for_building_data_matching_and_its_energy_applications

23. UBID Visualization [Електронний ресурс] – Режим доступу: https://github.com/pnml/buildingid/blob/master/VIZ_GUIDE.md

24. GNU General Public License [Електронний ресурс] – Режим доступу: https://ru.wikipedia.org/wiki/GNU_General_Public_License

25. Open Location Code Handling in R 0.3.0 [Електронний ресурс] – Режим доступу: <https://cran.r-project.org/web/packages/olctools/olctools.pdf>.

ДОДАТКИ

					ДИПЛОМНИЙ ПРОЕКТ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Виконав		Шкута Д.В.			Розроблення засобів формування унікальних ідентифікаторів будівель на основі відкритого коду місцеположення OLC в середовищі СКБД PostgreSQL	Літ.	Арк.	Аркушів
.		Лященко А.А.					1	1
Керівник		Лященко А.А.				ІПО КНУБА, група ГСТ-61		
Зав. каф.		Карпінський						

Результат перевірки на антиплагіат

Аналіз результатів на плагіат									
ID	Назва	Автори	Руководителі	Консультанти	Опоненти	Max Плагиат Си...	Ошибки, %	Кол. Лексем	
82336	Моделювання вл...	Серб Р. В.	Патракеєв І.М.			44.0	9	547	2020-
82186	Створення етал...	Троян Владисла...	Лазоренко-Геве...			12.0	19	442	2020-
82184	Моделювання вл...	Серб Р. В.	Патракеєв І.М.			58.0	10	511	2020-
82183	Геоінформаційн...	Лисенко І.П.	Патракеєв І.М.			2.0	8	749	2020-
82182	Геоінформаційн...	Василига Павло	Патракеєв І.М.			4.0	10	669	2020-
82180	Симуляція інтен...	Авраменко Кост...	Горковуч Денис...			8.0	10	443	2020-
81904	Розроблення за...	Шкута Дмитро В...	Лященко А.А.			2.0	15	790	2020-