НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
Кафедра геоінформатики і аерокосмічних досліджень Землі

ГІС І БАЗИ ДАНИХ В ОХОРОНІ ГРУНТІВ

КУРС ЛЕКЦІЙ

для студентів спеціальності H1. «Агрономія»

(Освітня програма «Охорона та технології відновлення ґрунтів»)

Київ - 2026

УДК 004.65:631.4

Курс лекцій розроблений для підготовки здобувачів вищої освіти за спеціальністю H1. «Агрономія» (освітня програма «Охорона та технології відновлення ґрунтів»). Навчальний матеріал присвячений опануванню теоретичних основ і практичних інструментів застосування сучасних геоінформаційних систем (ГІС), баз даних та мови SQL-запитів у сфері моніторингу, раціонального використання та охорони ґрунтового покриву.

Структура конспекту містить 7 тем, які висвітлюють еволюцію та моделі просторово-атрибутивних даних, методи концептуального й логічного проектування баз даних, алгоритми виконання просторових операцій та аналізу. Окрему увагу приділено практичному застосуванню ГІС-технологій для ґрунтозахисних досліджень, .

Рекомендовано вченою радою агробіологічного факультету Національного університету біоресурсів і природокористування України (протокол №1 від 28 серпня 2026 р.).

Укладачі: А. Москаленко, Б. Заячківська, В. Козак

Рецензенти: А.О. Кошель, Р.П. Богданович

Навчальне видання

Курс лекцій з дисципліни

ГІС І БАЗИ ДАНИХ В ОХОРОНІ ГРУНТІВ

для студентів спеціальності H1. «Агрономія»

(Освітня програма «Охорона та технології відновлення ґрунтів»)

Укладачі: Москаленко Антоніна Анатоліївна

Заячківська Богдання Богданівна

Козак Володимир Миколайович

ВСТУП

Курс «Геоінформаційні системи і бази даних в охороні ґрунтів» присвячений вивченню теоретичних і практичних аспектів сучасних методів організації, зберігання та аналізу просторово-координованої інформації, що використовується у сфері охорони ґрунтів. Дисципліна орієнтована на формування у студентів знань і навичок роботи з базами даних, інтеграції атрибутивних та геопросторових даних, виконання запитів, вибірки даних, проведення інших просторових операцій, а також застосування ГІС-рішень у практиці охорони ґрунтів.

У результаті опанування курсу студенти здобувають компетенції щодо розробки та впровадження геоінформаційних систем у сфері охорони ґрунтів, аналізу процесів деградації та відновлення, а також використання міжнародних стандартів і технологій у природоохоронних та аграрних проєктах.

Мета дисципліни полягає у формуванні в студентів знань та практичних навичок використання сучасних геоінформаційних систем і баз даних для вирішення завдань моніторингу, аналізу та охорони ґрунтових ресурсів

ТЕМА 1. ОСНОВИ СУЧАСНИХ БАЗ ДАНИХ ТА ЇХ РОЛЬ У ГІС

План

1. Поняття та еволюція баз даних
2. Основні моделі та технології баз даних
3. Інтеграція баз даних у геоінформаційні системи
4.Роль баз даних у моніторингу та охороні ґрунтів

1. Поняття та еволюція баз даних

База даних (БД) є впорядкованою сукупністю логічно взаємопов’язаних даних, що відображають стан об'єктів реального світу та їхні взаємозв'язки в певній предметній області [8]. У геоінформаційних системах (ГІС) база даних виконує роль інформаційного ядра, яке інтегрує як просторовий позиціонувальний складник (координати, топологія), так і атрибутивний (агрохімічні, фізико-механічні характеристики ґрунтів, кадастрові номери, показники деградації) [3].

Управління цими масивами здійснюється за допомогою Систем Управління Базами Даних (СУБД) - комплексу програмних і лінгвістичних засобів загального або спеціального призначення, що забезпечують створення, ведення та спільне використання БД багаторазовими користувачами [30].

Еволюційні етапи розвитку систем управління даними:

Файлові системи (1950–1960-ті рр.): Початковий етап, що характеризувався жорсткою прив'язкою даних до конкретних прикладних програм. Головними недоліками були дублювання інформації (надмірність), відсутність цілісності даних та складність складних вибірок. [14].

Ієрархічна та мережева моделі (1960–1970-ті рр.): Поява перших СУБД (наприклад, IMS від IBM). Дані подавалися у вигляді дерев (ієрархічна) або графів (мережева). Зв'язки між записами кодувалися фізичними вказівниками, що ускладнювало модифікацію структури БД при зміні вимог завдання [14].

Реляційна модель (з 1970-х рр.): Запропонована Едгаром Коддом модель, базована на математичній теорії множин та предикатів першого порядку. Дані подаються у вигляді двовимірних таблиць (відношень). Для маніпулювання даними розроблено декларативну мову SQL (Structured Query Language) [14].

Об’єктно-орієнтовані та об’єктно-реляційні СУБД (1990-ті рр.): Поява можливості зберігати складні типи даних та методи їх обробки безпосередньо в базі. Саме Об'єктно-Реляційні СУБД (ОРСУБД), такі як PostgreSQL або Oracle Spatial, стали базисом для сучасних просторових баз даних за рахунок підтримки розширюваних типів даних (Abstract Data Types, ADT) [14].

NoSQL та розподілені просторові бази даних (з 2010-х рр.): Орієнтовані на обробку "великих даних" (Big Data) та хмарні обчислення. Включають документо-орієнтовані (MongoDB), ключові (Redis), графічні (Neo4j) та просторово-розподілені бази даних (Apache Sedonas, Rasdaman для растрових кубів даних) [24].

З погляду геоінформаційного моніторингу ґрунтів, еволюція баз даних дозволила перейти від простих таблиць агрохімічного обстеження окремих полів

до складних розподілених баз геоданих національного та регіонального рівнів. Сучасна просторова БД зберігає не просто векторну графіку, а просторові об'єкти (Spatial Features), що володіють геометрією, атрибутами, топологічними зв'язками та координатною системою відліку (CRS) [27, 7].

Таблиця 1.1

Порівняльна характеристика етапів еволюції СУБД

Етап / Модель	Основні носії / Формати	Переваги	Основні обмеження для ГІС
Файлова система	Текстові файли, CSV, Binary	Простота реалізації для одиночних задач	Відсутність топології, відсутність просторових індексів, надмірність
Реляційна	Таблиці (Relations), SQL	Строга математична цілісність, нормалізація	Неефективність обробки 2D/3D геометричних об'єктів без розширень
Об'єктно-реляційна	PostGIS (PostgreSQL), Oracle Spatial	Підтримка OGC-стандартів, R-Tree індексація, геодезичні функції	Високі вимоги до обчислювальних ресурсів сервера
NoSQL / Cloud-Native	GeoJSON, Cloud Optimized GeoTIFF (COG), FlatGeobuf	Масштабованість, висока швидкість обробки растрів та Big Data	Ослаблена ACID-транзакційність (BASE-концепція)

2. Основні моделі та технології баз даних

Сучасні геоінформаційні системи спираються на різноманітні концептуальні та технологічні моделі баз даних, кожна з яких визначає способи структуризації, зберігання, індексування та маніпулювання просторово-координованою інформацією [12]. Вибір конкретної моделі даних та архітектури СУБД визначає продуктивність всієї геоінформаційної системи, особливо під час обробки масштабних масивів агрохімічного моніторингу та кадастрового обліку ґрунтів [7].

Спосіб обробки цифрових даних за допомогою комп’ютерів називається системою управління базами даних (СУБД). Ці системи з часом еволюціонували. Найперші СУБД мали навігаційний характер — від систем із плоскими файлами до ієрархічних систем. Це були відносно прості системи з обмеженою гнучкістю. Однак саме ці початкові системи заклали основу для розробки реляційних систем

управління базами даних (РСУБД), а згодом — і більш складних та ефективних систем.

Ієрархічна модель. Ієрархічна модель бази даних — це модель даних, у якій дані організовані у вигляді деревоподібної структури (рис. 2). Така структура дозволяє подавати інформацію за допомогою зв'язків типу «батьківський елемент — дочірній елемент»: кожен батьківський елемент може мати багато дочірніх, тоді як кожен дочірній елемент має лише один батьківський (такий зв'язок також називають зв'язком «один до багатьох»). Усі атрибути конкретного запису перелічуються в межах певного типу сутності.

У базі даних тип сутності є еквівалентом таблиці. Кожен окремий запис представлено у вигляді рядка, а кожен атрибут — у вигляді стовпця. Типи сутностей пов'язані між собою за допомогою відображень 1:N, які також називають зв'язками «один до багатьох». Ця модель вважається першою моделлю баз даних; її розробила компанія IBM у 1960-х роках [14].

Мережева модель. Мережева модель — це модель бази даних, створена як гнучкий спосіб представлення об'єктів та зв'язків між ними (рис. 3). Її відмінною рисою є те, що схема — якщо розглядати її як граф, де типи об'єктів є вузлами, а типи зв'язків — дугами, — не обмежується структурою ієрархії чи ґратки [14].

Реляційна система управління базами даних (РСУБД) дозволяє користувачам зберігати дані в кількох таблицях, де пов'язані елементи даних можуть взаємодіяти між собою через певні зв'язки. Кожен зв'язок має унікальну назву, за якою він ідентифікується в базі даних. Кожна таблиця складається з рядків, що називаються «записами», і стовпців, що називаються «полями». Кожне поле може містити дані лише певного типу, наприклад: текст, ціле число, десятковий дріб, дата тощо. Тип даних поля визначається під час його створення. Кожен запис у таблиці має унікальний ідентифікатор, який може вказувати на зв'язок між таблицями. Таблиці можна об'єднувати, якщо вони мають спільне поле. РСУБД забезпечує гнучкість у створенні численних зв'язків між даними, що підвищує ефективність їх пошуку та отримання. Система гарантує атомарність, узгодженість, ізольованість і довговічність (ACID) під час зберігання та отримання даних. Вона здатна зберігати великі обсяги інформації. Якщо застарілі системи управління базами даних часто дозволяли працювати з ними лише одному користувачеві одночасно, то РСУБД підтримують одночасну роботу багатьох користувачів. Проте можуть виникати конфлікти, коли кілька користувачів намагаються змінити одні й ті самі дані одночасно. Цьому можна запобігти за допомогою механізмів блокування та управління паралельним доступом. Блокування унеможливлює доступ інших користувачів до даних під час їх оновлення, тоді як управління паралельним доступом координує дії багатьох користувачів, що надсилають запити одночасно. Реляційні бази даних з'явилися в 1970-х роках і досі залишаються найпоширенішими у світі. Попри те, що вони стають складнішими, вони водночас стають кращими, швидшими та потужнішими [28].

Об'єктно-орієнтована система управління базами даних (ООСУБД)

Об'єктно-орієнтована система управління базами даних (ООСУБД) — це система управління базами даних, у якій дані представлено у вигляді об'єктів. Ці

об’єкти успадковують властивості класів та підкласів. Об'єктно-орієнтовані мови програмування, такі як Python і Java, можуть використовуватися для створення об'єктів баз даних та/або зберігання даних у вигляді об'єктів. Наразі ще не існує загальноприйнятих стандартів розробки об'єктно-орієнтованих систем управління базами даних (OODBMS). Однак такі системи повинні бути сумісними з сучасними об'єктно-орієнтованими мовами програмування. OODBMS краще підходять для зберігання даних зі складними взаємозв'язками та роботи з ними, забезпечуючи прості рішення для вилучення й обробки даних, що охоплюють кілька таблиць [28].

Хмарна система управління базами даних (CDBMS)/ Обсяги цифрових даних у всьому світі зростають експоненціально, і їх інтеграція та управління стають дедалі складнішими, якщо вони розміщуються на локальних серверах у різних місцях. Крім того, обробка даних у реальному часі стає новим стандартом для багатьох застосунків, оскільки затримки в передачі даних можуть призвести до катастрофічних наслідків. Кібербезпека — ще одне критично важливе питання, з яким стикаються компанії при розміщенні даних на власних серверах. Використання хмарних технологій для розміщення даних може допомогти вирішити деякі з цих проблем. Хмарні бази даних забезпечують гнучкість, надійність, безпеку, економічну ефективність та низку інших переваг [28].

Хмарна система управління базами даних розміщується на віртуальній машині та пропонує всі переваги хмарних обчислень, такі як швидкість, масштабованість, гнучкість і зниження витрат, завдяки розгортанню, наданню та доступу через хмару. Такі бази даних подібні до традиційних локальних систем і можуть використовуватися для зберігання структурованих, неструктурованих і напівструктурованих даних. Хмарна база даних може розміщуватися в публічному або гібридному хмарному середовищі. Вона може керуватися безпосередньо компанією-власником або ж зовнішнім постачальником у рамках моделі «база даних як послуга» (DBaaS). Обидва підходи мають свої переваги та недоліки. На хмарних платформах можуть розміщуватися як реляційні, так і нереляційні бази даних [28].

NoSQL та документоорієнтовані моделі: Використовуються для обробки неструктурованих або напівструктурованих геоданих великого обсягу (Big Spatial Data). Популярні сутності зберігають дані у формі JSON/GeoJSON документів (MongoDB, CouchDB) або графічних структур (Neo4j для топологічних мереж) [12].

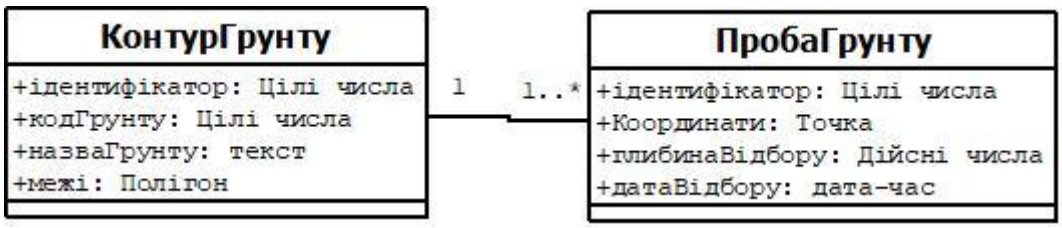
Сховища даних. Обсяги даних у всьому світі стрімко зростають, сягаючи петабайтів. Виконання аналітики таких великих масивів даних (Big Data) за допомогою стандартних методів роботи з базами даних є складним або й неможливим завданням. Так само отримання даних із різних джерел завжди було викликом для пересічних користувачів. Сховище даних пропонує ефективне вирішення цієї проблеми, дозволяючи користувачам збирати дані з різних джерел в єдине, централізоване та узгоджене сховище. Сховище даних підтримує аналіз даних, інтелектуальний аналіз даних (data mining), технології штучного інтелекту (ШІ) та машинного навчання. Воно також надає організації можливість проводити потужну аналітику великих даних [28].

Концепція сховищ даних не є новою: вона існує з 1980-х років як складова рішень для бізнес-аналітики (BI) і спочатку реалізовувалася на локальних серверах (on-premises). Такі сховища були призначені для вилучення даних із різних джерел, їх очищення та підготовки, а також для завантаження й підтримки в традиційних реляційних базах даних. Однак розвиток інтернет-технологій уможливив їх розміщення в хмарі. Популярність хмарних сховищ даних стрімко зростає, оскільки дедалі більше компаній використовують хмарні сервіси, щоб зменшити залежність від власних локальних центрів обробки даних. Сучасні сховища даних також оснащуються інструментами для аналітики, візуалізації та представлення даних. До основних категорій сховищ даних належать корпоративні сховища даних (EDW), сховища оперативних даних (ODS) та вітрини даних (DM).

Просторова база даних – це база даних, що містить просторовий компонент (часто у вигляді координат), пов’язаний з атрибутами даних, для визначення місцезнаходження та геометричних характеристик об’єкта. Вона дозволяє зберігати в системі як векторні (точки, лінії, полігони), так і растрові дані. Просторові бази даних спеціально розроблені для представлення просторових даних та їх обробки в межах просторової системи координат. Оскільки для визначення місцезнаходження об’єкта завжди потрібна система просторової прив’язки, така база даних повинна підтримувати перетворення систем координат. Просторові бази даних зазвичай складніші за традиційні, оскільки взаємозв’язки між просторовими об’єктами часто є неявними. Вони дозволяють моделювати такі відношення, як «позаду», «перетин», «накладання» тощо. Така база даних є фундаментальним елементом будь-якої геоінформаційної системи (ГІС), забезпечуючи виконання низки просторових операцій, недоступних у традиційних базах даних. Консорціум Open Geospatial Consortium (OGC) [27] розробив стандарти для додавання просторових функціональних можливостей до просторових баз даних. Розробка просторових баз даних уможливлює впровадження нового набору функцій в інструкції «SELECT» мов запитів до даних (наприклад, SQL). У просторових базах даних використовується система просторового індексування, спеціально розроблена для багатовимірного впорядкування просторових даних із одночасним встановленням геометричних взаємозв’язків між різними об’єктами та елементами [28].

Приклад структуризації просторово-атрибутивних зв'язків для моніторингу ґрунтів

У задачах агрохімічного моніторингу ґрунтів реалізується реляційний зв'язок типу «один-до-багатьох» (1:M) між просторовим контуром земельної ділянки та часовим рядом спостережень:



Така організація дозволяє зберігати просторові межі поля один раз, прив'язуючи до нього необмежену кількість повторних агрохімічних обстежень за різні роки без дублювання геометрії.

3. Інтеграція баз даних у геоінформаційні системи

Інтеграція баз даних (БД) та систем управління базами даних (СУБД) із геоінформаційними системами (ГІС) є фундаментальною передумовою для побудови сучасних просторово-аналітичних комплексів. Основна мета такої інтеграції полягає в забезпеченні єдиного, узгодженого та високопродуктивного середовища для збереження, маніпулювання, аналізу та візуалізації як просторової (геометричної та топологічної) складової геоданих, так і складних масивів атрибутивної (семантичної) інформації [12].

В історичному та технологічному аспекті формування архітектури зв'язку між ГІС та СУБД проходило декілька етапів — від розширення файлових систем до створення повнофункціональних об'єктно-реляційних баз геоданих та хмарних розподілених середовищ [8].

Концептуальні архітектури інтеграції СУБД та ГІС

Виділяють три базові архітектурні моделі, які визначають спосіб взаємодії між геоінформаційним ПЗ та базах даних:

Гібридна (подвійна) архітектура (Dual / Hybrid Architecture) У гібридній архітектурі, характерній для ранніх ГІС-пакетів (наприклад, ранні версії ArcView, PC ARC/INFO), просторова та атрибутивна інформація розділені на системному рівні: Просторові дані (координати точок, вершин полігонів, топологія) зберігаються у спеціалізованих бінарних файлах розробника ГІС (наприклад, ESRI Coverage або Shapefile). Атрибутивні дані зберігаються у зовнішній реляційній СУБД або таблицях dBase (.dbf). Зв'язок між геометрією та атрибутами здійснюється через унікальний системний ідентифікатор об'єкта (Feature ID / FID).

Повноінтегрована (об'єктно-реляційна) архітектура (Integrated Spatial DBMS Architecture) Сучасний галузевий стандарт, за якого і геометрія, і атрибути зберігаються в єдиному просторі реляційної або об'єктно-реляційної СУБД (PostgreSQL/PostGIS, Oracle Spatial, Microsoft SQL Server) [29]. Геометрія подається як спеціальний тип даних (ST_Geometry, GEOMETRY, GEOGRAPHY) відповідно до специфікацій OGC *Simple Feature Access* [27].

Розподілена та сервіс-орієнтована архітектура (Service-Oriented Architecture, SOA). Базується на використанні веб-протоколів та специфікацій Open Geospatial Consortium (OGC) для передачі просторових даних між СУБД та ГІС-клієнтами через проміжне програмне забезпечення (Middleware) — картографічні сервери (GeoServer, MapServer, ArcGIS Enterprise): WMS (Web Map

Service): Передача растрового зображення картографічного шару, згенерованого на сервері на основі даних із СУБД. WFS / WFS-T (Web Feature Service / Transactional): Передача безпосередньо векторних об'єктів (у форматі GML або GeoJSON) із можливістю їх редагування на клієнті та зворотної транзакції у СУБД. OGC API - Features: Сучасний RESTful-стандарт, що замінює класичний WFS для роботи з просторовими даними через JSON/GeoJSON.

Таблиця 1.2

Порівняльна характеристика моделей інтеграції

Критерій оцінки	Гібридна модель	Повноінтегрована ОРСУБД	Сервіс-орієнтована (SOA)
Місце збереження геометрії	Векторні файли (.shp, .gdb)	Таблиці СУБД (ST_Geometry)	Просторова СУБД через сервери
Підтримка ACID-транзакцій	Відсутня	Повна	Повна (через WFS-T / API)
Багатокористувацький доступ	Обмежений / Локальний	Високий (паралельні транзакції)	Необмежений (через веб-клієнти)
Просторові індекси	Індекси файлів (.sbn, .qix)	R-Tree / GiST / Quadtree	Індекси СУБД + Кеш сервера
Основний протокол/мова	Внутрішній API ГІС	SQL / Spatial SQL Extensions	HTTP / REST / OGC Standards

4. Роль баз даних у моніторингу та охороні ґрунтів

Бази даних є інформаційно-технологічним фундаментом сучасних систем державного й виробничого моніторингу та охорони ґрунтів. Специфіка ґрунтово-географічної інформації полягає в її високій просторовій гетерогенності, тривимірності (врахування глибини ґрунтового профілю) та часовій динамічності (зміна агрохімічних показників під впливом антропогенного навантаження й кліматичних чинників) [12, 7, 8].

Застосування просторових СУБД дозволяє перейти від статичного обліку якісного стану земель до динамічного просторово-часового моделювання деградаційних процесів, прогнозування ризиків та прийняття обґрунтованих управлінських рішень у сфері землекористування [19].

Основні завдання баз даних у ґрунтознавстві та охороні земель

– Накопичення та просторова прив'язка ретроспективних даних: Збереження часових серій спостережень (вміст гумусу, рухомого фосфору, калію, рівень pH, засолення) для аналізу динаміки родючості протягом десятиліть.

– Інтеграція векторних та растрових моделей: Поєднання просторових меж контурів ґрунтових відмін (вектор) із растровими даними вегетаційних індексів

(NDVI, EVI з Sentinel-2/Landsat) та ЦМР (DEM) для моделювання ерозійних процесів.

– Оцінка та моніторинг деградаційних процесів: Автоматизований розрахунок ризиків водної та вітрової ерозії, забруднення важкими металами та пестицидами.

– Підтримка рішень у точному землеробстві (Precision Agriculture): Збереження картограм агрохімічного обстеження для диференційованого внесення добрив та вапнування.

– Облік порушених та забруднених земель: Ведення баз даних технологічно забруднених, деградованих та потенційно небезпечних територій (включаючи зони військового впливу та механічного пошкодження ґрунтового покриву).

o Інтеграція баз даних із ГІС дає змогу створювати карти ризиків, моделі просторової агрегації та інші інструменти для прийняття рішень у сфері охорони ґрунтів

Просторово-часова (4D) специфіка ґрунтових даних

Особливістю баз даних у ґрунтознавстві є необхідність оперування чотиривимірними координатами (X, Y, Z, T) :

– Площинні координати (X, Y) : Визначають просторове положення об'єкта у вибраній системі координат.

– Вертикальна координата (Z) : Відповідає глибині відбору проби або межам генетичних ґрунтових горизонтів (H, HP, Ph, P).

– Часова координата (T) : Фіксує дату та час проведення агрохімічного обстеження, що дозволяє формувати хронологічні зрізи (Time-Series Data) для аналізу трендів дегуміфікації або закислення [19].

Контрольні питання

1. Що є база даних і які основні функції вона виконує?
2. Які етапи розвитку баз даних можна виділити від файлових систем до сучасних моделей?
3. У чому полягає сутність реляційної моделі баз даних?
4. Які переваги мають об’єктно-орієнтовані та документно-орієнтовані бази даних?
5. Назвіть приклади сучасних систем управління базами даних (СУБД), що застосовуються у ГІС.
6. Як інтеграція баз даних у ГІС сприяє аналізу просторових процесів?
7. Які можливості надає використання баз даних для моніторингу стану ґрунтів?
8. Чому бази даних вважаються стратегічним ресурсом у сфері охорони ґрунтів та екологічної безпеки?

ТЕМА 2. МОДЕЛІ ПРОСТОРОВИХ ДАНИХ

План

1. Поняття та класифікація просторових даних
2. Векторна модель просторових даних
3. Растрова модель просторових даних
4. Інтегровані та тривимірні моделі просторових даних

1. Поняття та класифікація просторових даних

Просторові дані – це відомості про географічні об’єкти, явища або процеси реального світу, що містять інформацію про їхнє просторове положення, геометричну форму, топологічні відношення, атрибутивні властивості та часову динаміку [7, 8; 12].

Просторові дані – це будь-який тип даних, що прямо чи опосередковано вказують на певну географічну територію або місцезнаходження. Найчастіше їх обробляють і аналізують за допомогою географічних інформаційних систем (ГІС), що дозволяє користувачам із різних галузей розуміти географічний контекст, закономірності та взаємозв’язки для подальшого аналізу, комунікації та прийняття рішень [10].

Просторові дані (які також називають геопросторовими даними або географічною інформацією) найчастіше представляють фізичний об’єкт (природний або створений людиною) у числовому вигляді в системі географічних координат. У цій системі використовуються двовимірні (2D) географічні координати, відомі як декартові координати (або координати x та y). Однак поняття просторових даних виходить далеко за межі географічних координат, відображених на двовимірній карті. Просторова інформація може також включати геометричні фігури (наприклад, лінії чи полігони), описи конкретних географічних об’єктів, а також зображення у форматі растру (сітки пікселів) [10].

Оскільки просторові дані можуть містити не лише інформацію про місцезнаходження, різні їх типи можуть зберігатися в різних форматах. Збір та аналіз просторових даних забезпечують краще розуміння того, як кожен чинник у географічному просторі впливає на окремих людей, громади, популяції тощо [10].

Просторові дані можуть збиратися шляхом наземних геодезичних вимірювань або переписів населення, на основі земельних реєстрів, а також за допомогою супутникових знімків, аерофотозйомки чи приймачів глобальної системи позиціювання (GPS). Дедалі частіше для збору просторових даних використовують також безпілотні літальні апарати (дрони) та бездротові датчики [10].

Виділяють два основні види просторових даних: геометричні та географічні [10].

Геометричні дані. Геометричні дані відображаються на двовимірній плоскій поверхні, наприклад, на карті або плані приміщення. Відомим прикладом практичного застосування геометричних даних є сервіс Google Maps.

Геометричні геопросторові дані дозволяють Google Maps прокладати точні маршрути між різними локаціями.

Географічні дані. Географічні дані - це інформація, прив’язана до сферичної поверхні, зазвичай Землі. Вони визначають широту й довготу конкретного об’єкта або місця. Знайомим прикладом географічних даних є інформація, що генерується системою GPS [10].

Двома найпоширенішими форматами зберігання геопросторових даних є векторний і растровий [10].

Векторний формат. Векторні елементи, такі як точки, лінії та полігони, відображають реальний світ у графічному вигляді. Векторні дані найчастіше використовуються для позначення об’єктів на поверхні Землі. Це можуть бути як природні об’єкти (наприклад, дерева), так і штучні (дороги чи будівлі). Саме тому векторні дані також називають даними про об’єкти (feature data) [10].

Лінії утворюються шляхом з’єднання точок; з’єднання кількох ліній формує замкнені ділянки або полігони. Точки позначають об’єкти, для визначення місцезнаходження яких достатньо однієї пари координат (наприклад, епіцентр землетрусу). Лінії (які не обов’язково мають бути прямими — це залежить від розташування точок) часто позначають дороги, річки та інженерні мережі. Полігони відображають ділянки з чіткими межами, наприклад, земельні зони або будівлі [10].

Векторні дані зазвичай зберігаються у файлах формату shapefile (або .shp). Крім того, розмір векторних зображень можна змінювати (збільшувати чи зменшувати) без втрати якості [10].

Растровий формат. Растрові дані (також відомі як дані покриття) — це графічна інформація, що зазвичай подається у вигляді растрових зображень (хоча вони можуть зберігатися і у форматах JPEG, TIFF, GIF чи PNG). Такі зображення формуються з пікселів — дрібних точок, що утворюють сітку. Растрові дані представлені як набір значень, розподілених по цій сітці пікселів [10].

Кожен піксель у растрі має певне значення, що зазвичай стосується кольору або тону. Сукупність пікселів утворює зображення. Чим більше пікселів містить зображення і чим вища їхня щільність (кількість пікселів на дюйм), тим вищою є якість або чіткість зображення. Зміна розміру растрового зображення впливає на його якість, оскільки пікселі розтягуються на більшу площу [10].

У просторовому аналізі під растром часто розуміють ортофотоплан або знімки, отримані з повітряних апаратів чи супутників. Цифрові моделі рельєфу (DTM) та цифрові моделі поверхні (DSM) також використовують растрові геопросторові дані. Ці моделі містять геопросторову (растрову) інформацію про земну поверхню та її природні й штучні (створені людиною) об'єкти; вони створюються за допомогою картографічних систем на базі безпілотних літальних апаратів (БПЛА) для широкого спектра застосувань [10].

Трискладова концептуальна структура геоданих

Будь-який просторовий об'єкт у геоінформаційних системах описується триадою взаємопов'язаних компонентів [7]:

Просторова (геометрична та топологічна) складова: визначає абсолютне положення об'єкта в обраній системі координат (X, Y, Z) , його конфігурацію,

протяжність, а також просторові взаємозв'язки з сусідніми об'єктами (прилягання, перетин, включення).

Атрибутивна (семантична) складова: якісні та кількісні характеристики об'єкта, що описують його фізичні, хімічні, технологічні чи правові властивості (наприклад, кадастровий номер ділянки, вміст гумусу, показник pH , тип ґрунту) [14].

Часова (темпоральна) складова (T): відображає хронологічний стан об'єкта або його динаміку в часі (дата проведення агрохімічного обстеження, часові серії вегетаційних індексів NDVI) [12].

Базова класифікація просторових даних – за жерелами отримання та первинними форматами [7]:

Дистанційні дані: космічна та аерофотозйомка, матеріали з БПЛА, дані оптичних, мультиспектральних та радарних сенсорів.

Наземні інструментальні дані: геодезичні вимірювання (GNSS-приймачі, електронні тахеометри), лазерне сканування (LiDAR), наземне SLAM-сканування.

Польові та лабораторні дослідження: координатно-прив'язані точки відбору ґрунтових проб, описи ґрунтових розрізів, дані агрохімічних лабораторій.

Картографічні та фондові джерела: оцифровані паперові тематичні карти (ґрунтові, топографічні, геологічні).

Базова класифікація просторових даних – за просторовою розмірністю (Dimensionality) [12]:

Розмірність	Позначення	Сфера застосування в ГІС та агромоніторингу
Двовимірні	2D (X, Y)	Планові картографічні матеріали, кадастрові межі, контури полів.
Псевдотривимірні	2.5D (X, Y, Z_A)	Поверхні, де Z виступає як одиночна атрибутивна функція від X,Y (ЦМР, картограми вмісту N, P, K).
Повнотривимірні	3D (X, Y, Z)	Воксельні моделі ґрунтового профілю, геологічні пласти, хмари точок LiDAR/SLAM, 3D-моделі об'єктів.
Чотиривимірні	4D (X, Y, Z, T)	Просторово-часові базі даних для аналізу деградації ґрунтів, моніторингу ерозійних процесів у динаміці.

Системи просторової координації та метадані

Для забезпечення точності суміщення геоданих із різних джерел використовується точна прив'язка до систем координат за допомогою унікальних ідентифікаторів SRID (Spatial Reference Identifier) згідно з реєстром EPSG та специфікацій OGC [27]: Планові системи координат: Державна геодезична система координат УСК-2000 (наприклад, проекція Gauss-Kruger / WGS84 UTM zone 36N) та Глобальні системи: WGS 84 (EPSG:4326), що використовується у системі GPS та веб-сервісах.

Обов'язковим елементом систем баз геоданих є метадані (дані про дані), регламентовані міжнародними стандартами ISO 19115 та Директивою INSPIRE [12]. Вони описують походження, просторове охоплення, актуальність, точність та системи координат конкретного набору просторових даних.

2. Векторна модель просторових даних

Векторна модель просторових даних базується на дискретному представленні об'єктів реального світу за допомогою точних геометричних координат у двовимірному або тривимірному просторі (X, Y, Z) [7, 8; 12]. Векторне моделювання орієнтоване на відображення об'єктів із чітко визначеними межами та дискретним характером розповсюдження.

Векторні дані – це модель просторових даних, що використовується в геоінформаційних системах (ГІС) для представлення географічних об'єктів за допомогою точок, ліній та полігонів. Кожен елемент визначається точними парами координат (x, y), які вказують на його точне розташування на поверхні Землі. Точки позначають конкретні місця, лінії відображають лінійні об'єкти (наприклад, дороги чи річки), а полігони окреслюють площі, такі як земельні ділянки або водойми [9].

На відміну від растрових даних, що базуються на сітках комірок, векторні дані зберігають високу точність і масштабованість, що робить їх ідеальними для завдань, які потребують детального картографування та просторового аналізу. Вони широко застосовуються в навігації, містобудуванні та моніторингу довкілля [9].

Ключові компоненти векторних даних

Точки — окремі пари координат (x, y), що позначають конкретні місця, наприклад, визначні пам'ятки, станції датчиків або навігаційні контрольні точки.

Лінії (полілінії) — послідовності з'єднаних точок, що утворюють лінійні об'єкти, такі як дороги, річки чи трубопроводи.

Полігони — замкнені області, утворені з'єднанням кількох точок; використовуються для представлення земельних ділянок, водойм або адміністративних меж.

Вершини — окремі координатні точки, що визначають форму лінії або полігона.

Ребра — лінії зв'язку між вершинами, що формують структуру ліній та полігонів.

Топологія — просторові взаємозв'язки між векторними об'єктами, що забезпечують точність даних і їхню взаємопов'язаність у ГІС-застосунках.

Атрибутивні дані — описова інформація, пов'язана з кожним векторним об'єктом (наприклад, назви доріг, висота будівель або типи землекористування).

Система координат — система відліку, що використовується для точного прив'язування векторних даних до реальних географічних місць.

Масштабованість — здатність зберігати високу деталізацію незалежно від рівня масштабування, що робить цей формат ідеальним для точного просторового аналізу [9].

Векторні дані представлені у вигляді координат. Ці координати визначають просторове розташування, а також можуть характеризувати розмір або межі об'єкта. У векторній моделі даних виділяють три типи об'єктів: точка, лінія та полігон. Точка — це безрозмірний об'єкт, що позначає місцезнаходження елемента, який є занадто малим для відображення у вигляді лінії чи полігона в заданому масштабі карти. Лінія — це набір точок, що використовується для представлення лінійних об'єктів (наприклад, доріг), які є надто вузькими для відображення у вигляді площі в масштабі карти. Кожна лінія має початкову та кінцеву точки, а також проміжні точки. Кожна точка на лінії визначається набором координат. Початкова та кінцева точки лінії називаються вузлами, тоді як проміжні точки, що також визначають форму лінії, називаються вершинами. Таким чином, лінія не обов'язково має бути прямою; вона може складатися з кількох з'єднаних прямих відрізків або інших лінійних геометричних форм, як-от криві. Такі лінії також називають полілініями. Аналогічно, полігон — це сукупність з'єднаних ліній або поліліній, що утворюють замкнену геометричну фігуру, яка представляє площу об'єкта. Початковий і кінцевий вузли полігона збігаються в одній точці. Кожен полігон має внутрішню область, яка може охоплювати інший полігон. Суміжні полігони можуть мати спільні межі [30].

Оскільки дані записуються у вигляді координат, що описують точку, лінію або полігон, фіксація інформації відбувається лише там, де це необхідно. Завдяки цьому обсяг даних у векторній моделі значно менший порівняно з растровою моделлю. Менший обсяг векторних даних забезпечує швидше відображення та обробку інформації, а також потребує менших обчислювальних ресурсів. Векторні дані не втрачають чіткості (не пікселізуються) при зміні масштабу (наближенні), тому карти, створені на основі векторних даних, не мають обмежень щодо масштабування. Використання координат дозволяє точно вимірювати розташування, довжину та площу об'єктів, хоча масштаб, у якому здійснювався збір даних, може накладати певні обмеження. Крім того, у векторній моделі межі об'єктів чітко визначені. Векторні моделі дозволяють користувачам додавати до таблиці атрибутів будь-яку кількість полів без обмежень щодо формату даних. Ці поля можна легко оновлювати, а також об'єднувати чи пов'язувати із зовнішніми таблицями. Вартість друку карт, створених на основі векторних моделей, значно нижча порівняно з картами, створеними за допомогою растрових моделей. Векторні дані можна легко відображати поверх растрових, тоді як растрові дані не можна розміщувати поверх векторних, оскільки вони перекриватимуть останні [30].

Векторні базові моделі також називають об'єктно-орієнтованими моделями, де кожен об'єкт має чітко визначені межі або розміри. Дані для таких

моделей збирають за допомогою традиційних геодезичних методів, оцифрування на екрані або шляхом інтерпретації растрових даних, наприклад аерофотознімків [30].

Векторну модель даних поділяють на дві підмоделі.

Модель даних «спагеті» — це найперша модель, розроблена для представлення лінійних об'єктів, де кожна лінія має чітко визначені початкову та кінцеву точки (вузли). Кожна лінія записується та відображається окремо. Спільна межа сусідніх полігонів може бути записана або відображена двічі, причому ці межі можуть не повністю збігатися. Модель «спагеті» проста й ефективна для візуалізації та побудови графічних зображень. Однак це відносно неструктурований спосіб представлення векторних даних. Відсутність інформації про зв'язність об'єктів може призвести до помилок під час аналізу. У місцях з'єднання лінійних об'єктів можуть виникати розриви або накладання. Так само полігони можуть не утворювати замкнену область, через що неможливо обчислити площу такого об'єкта. Отже, модель «спагеті» надає дуже обмежені можливості для просторового аналізу в середовищі ГІС. У певних ситуаціях виконання мережевого аналізу з використанням цієї моделі взагалі неможливе, оскільки вона не зберігає топологію об'єктів. Виконання просторового або мережевого аналізу може призвести до помилкових результатів, оскільки об'єкти можуть не з'єднуватися там, де це необхідно. Перед проведенням такого аналізу дані слід очистити, а топологічні помилки — усунути [30].

Топологічні векторні моделі. Обмеження спагеті-моделей для просторового аналізу були усунені за допомогою топологічної моделі даних, яка значно покращила швидкість і точність просторового аналізу в середовищі ГІС. Цього було досягнуто завдяки суворому дотриманню інформації про зв'язність та суміжність. Просторовий зв'язок між подібними об'єктами суворо підтримується та забезпечується. Цей зв'язок зазвичай записується окремо від інформації про координати, і тому розтягування, вигин або масштабування не впливають на топологічну інформацію. У топологічній моделі немає перекриття меж ліній або полігонів. Лінії, що перетинаються, мають вузол, що відображає перетин. Відсутність вузла на лініях перетину вказуватиме на те, що лінії не є компланарними і, отже, не перетинаються, і їх слід розглядати окремо під час виконання просторового або мережевого аналізу. Вузли на лініях також можуть використовуватися для зберігання інформації "від" для полегшення лінійного посилання [30].

Аналогічно, межа полігону, що перекривається, матиме лише один відрізок лінії, що представляє спільну межу суміжних полігонів. Однак некомпланарні полігони не матимуть спільної межі. Як і лінійний об'єкт, вузли розміщуються на початку та в кінці спільної межі компланарних суміжних полігонів [30].

Топологічні векторні моделі даних дозволяють користувачам виконувати кілька геопросторових операцій і тому корисні для просторового аналізу. Однак, дані повинні бути чистими, щоб виправити топологічні помилки. Різне ГІС-програмне забезпечення може використовувати різні підходи та структури даних для зберігання топології. Деяке програмне забезпечення дозволяє користувачам будувати топологічні зв'язки, коли це необхідно для спрощення структур даних,

тоді як інше може зберігати топологічні зв'язки у своїх даних. Обчислювальні витрати на топологічну модель даних, як правило, вищі порівняно з моделлю "спагетті". Топологічні моделі також вимагають більше зусиль для створення чистіших даних, вільних від топологічних помилок; однак, ГІС-програмне забезпечення зазвичай надає інструменти, що допомагають у створенні чистіших даних для топологічної моделі. Незважаючи на більшу обчислювальну потужність та зусилля, необхідні для топологічних моделей, воно пропонує набагато більші переваги в аналітичних можливостях даних [30].

Формати збереження та бази геоданих

Сучасні векторні дані зберігаються як у файлових форматах, так і у просторово-орієнтованих реляційних СУБД [29]:

Тип збереження	Формат / СУБД	Характеристика
Файловий	ESRI Shapefile (.shp)	Класичний векторний формат (складається з .shp, .dbf, .shx), має обмеження за розміром (2 ГБ) та довжиною назв полів (10 символів).
Сучасний відкритий файл	GeoPackage (.gpkg)	Відкритий стандарт OGC на основі бази даних SQLite. Зберігає векторні, растрові дані та атрибути в одному файлі без обмежень Shapefile.
Просторові СУБД	PostgreSQL + PostGIS	Професійна база геоданих. Зберігає векторні об'єкти у бінарному типі даних GEOMETRY / GEOGRAPHY із використанням просторового індексування R-Tree / GiST.

Прикладне застосування векторної моделі в охороні ґрунтів

У контексті моніторингу та охорони ґрунтового покриву векторна модель застосовується для:

Точкового шару: створення бази даних просторово-координатної прив'язки ґрунтових розрізів та лабораторних зразків (із прив'язкою даних про pH , вміст N, P, K , важких металів).

Лінійного шару: оцифрування протиерозійної гідротехнічної мережі, валів-канав, меж водозборів та тальвегів балково-яружної мережі.

Полігонального шару: ведення цифрових карт ґрунтових відмінностей, картограм якісної оцінки земель (бонітування), контурно-меліоративної організації території та ділянок, вилучених з обробітку (консервація деградованих земель).

3. Растрова модель просторових даних

Растрова модель просторових даних базується на регулярному дискретному розбитті простору на систему суміжних елементарних комірок

(пікселів), що утворюють двовимірну сітку (матрицю або решітку — *Grid*) [7; 12].

Геопросторові дані представлені у вигляді набору комірок однакового розміру, організованих у сітку. Зазвичай ці комірки мають квадратну форму та рівномірно розподілені у просторі координат «x, y». Прикладами растрових даних є цифрові фотографії або зображення, отримані, наприклад, із супутників чи аерофотозйомки. Кожна комірка зображення називається пікселем (елементом зображення) і містить лише одне значення, яке зазвичай відображає інтенсивність відбиття світла від об'єкта. Розмір пікселя або їх кількість визначають чіткість зображення та його просторову роздільну здатність. Що менший розмір пікселя, то вища просторова роздільна здатність. Усе зображення складається з пікселів; у растровій моделі даних відсутні пропуски, завдяки чому вона найкраще підходить для представлення безперервних даних, таких як крутизна схилу, вологість ґрунту та цифрова модель рельєфу (ЦМР). Оскільки кожен піксель містить певне значення, а пікселі суцільно заповнюють усе зображення, обсяг растрових даних зазвичай дуже великий, що сповільнює їх відображення. Для візуалізації великих масивів даних може знадобитися потужніша відеокарта. При збільшенні масштабу зображення розмір кожного пікселя на екрані зростає, доки окремі пікселі даних не стануть видимими; отже, растрова модель має обмеження щодо масштабування, і за певного рівня збільшення об'єкти можуть стати нерозпізнаваними. Розмір пікселя в растрових даних також зумовлює похибки при вимірюванні розмірів об'єкта. Зменшення розміру пікселя дозволяє знизити похибки вимірювання розмірів (об'єму або довжини). Аналогічно, розмір пікселя обмежує точність визначення місцеположення, тому растрові дані зазвичай мають обмежену точність позиціювання. Якщо зменшити розмір пікселя вдвічі, кількість пікселів у наборі даних зросте вчетверо. Хоча зменшення розміру пікселя покращує чіткість зображення та точність вимірювань і позиціювання, водночас збільшується обсяг даних. Тому необхідно знаходити баланс між розміром пікселя та обсягом даних [30].

Головним обмеженням растрових даних є те, що їхня атрибутивна таблиця може містити лише три поля: OID (ідентифікатор об'єкта), Value (значення) та Count (кількість). Ці поля можуть містити лише числові значення, як правило, цілі числа. Перше поле представлене первинним ключем або унікальним ідентифікаційним номером. Поле «Value» відображає значення пікселя, а останнє поле «Count» — кількість пікселів із таким самим значенням. Поле «value» (значення) переважно використовується для класифікації растрових даних за різними категоріями. У деяких випадках таблицю атрибутів растрових даних (особливо якщо вони представляють безперервні дані) неможливо відобразити, доки її не буде створено. Для цього деякі програмні продукти, як-от ArcMap, пропонують спеціальні інструменти, наприклад «Build Raster Attribute Table» (Створити таблицю атрибутів растра). Однак цей інструмент працює лише з одноканальними растровими даними й може не спрацювати з композитним растровим зображенням. Оскільки обсяг растрових даних зазвичай є значним і потребує потужніших обчислювальних ресурсів для обробки, доцільно

перетворювати їх на векторні дані перед початком обробки. Водночас існують певні аналітичні функції, які можна застосовувати лише до растрових даних і які неможливо реалізувати для векторних даних. Перед перетворенням на векторний формат растрові дані необхідно класифікувати [30].

Растрові моделі даних також називають моделями, що базуються на понятті поля (field-based models). Дані для таких моделей отримують безпосередньо за допомогою аерофотозйомки або супутникових знімків. Для формування даних для моделей цього типу також можуть використовуватися математичні та геостатистичні методи, такі як інтерполяція, вибірка або рекласифікація. Прикладами растрових даних, створених за допомогою геостатистичних методів (на основі точкових даних або ізоліній), є цифрові моделі рельєфу (ЦМР), карти погодних умов і теплові карти. Просторові явища, що описуються такими моделями даних, змінюються безперервно [30].

Формати растрових даних у ГІС

GeoTIFF / COG (Cloud Optimized GeoTIFF): де-факто світовий стандарт для дистанційного зондування та ГІС. COG дозволяє запитувати лише необхідні фрагменти растра через HTTP без завантаження всього файла.

NetCDF / HDF5: багатовимірні растрові формати, що зберігають часові серії та просторові сітки (X, Y, Z, T). Активно використовуються у кліматології та просторовому моделюванні вологості ґрунту.

PostGIS Raster: модуль СУБД PostgreSQL, що дозволяє зберігати растрові тайли безпосередньо в реляційній базі даних та виконувати просторові SQL-запити до растрів [29].

Застосування растрової моделі в охороні ґрунтів та моніторингу

Цифрові моделі рельєфу (ЦМР / DEM): є базою для розрахунку топографічного фактора ерозійної небезпеки (LS -фактор у моделі RUSLE) та моделювання шляхів стоку поверхневих вод.

Дистанційне зондування Землі (ДЗЗ): аналіз спектральних відбивальних властивостей відкритого ґрунтового покриву для оцінки забур'яненості, зволоження, дегуміфікації та засолення ґрунтів.

Цифрове картографування ґрунтів (Digital Soil Mapping — DSM): створення суцільних растрових поверхонь просторового розподілу агрохімічних та фізичних властивостей ґрунтів (картограми вмісту N, P, K , pH , щільності складення)

Порівняння растрових і векторних моделей даних

Геопросторові дані можна записувати, представляти та аналізувати за допомогою різних моделей даних. Кожна модель має свої переваги та недоліки. Одна модель може найкраще підходити для певного типу аналізу, тоді як інші ефективніші за інших умов. Деякі види геопросторових даних, наприклад цифрові моделі рельєфу (DEM), можна представляти за допомогою різних моделей даних. Користувач може обрати конкретну модель для виконання ГІС-аналізу. Водночас моделі даних у ГІС є взаємозамінними.

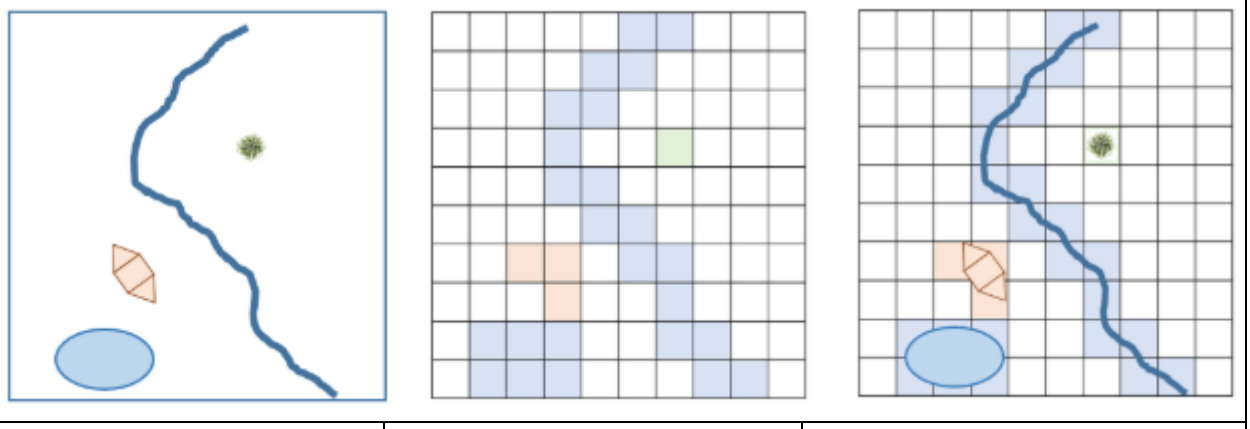
Таблиця 2.1.

Порівняльна характеристика [30]

№	ВЕКТОРНІ ДАНІ	РАСТРОВІ ДАНІ
1	Таблиця атрибутів може мати більше ніж три поля	Таблиця атрибутів має лише три поля (OID, Value, Count)
2	Числові та текстові (рядкові) атрибути	Лише числові атрибути
3	Точне просторове положення об'єктів	Обмежена точність визначення положення об'єктів
4	Межі чітко визначені	Межі визначені не дуже чітко
5	Точні вимірювання довжини/об'єму	Точність залежить від розміру пікселя
6	Записує значення тільки там, де необхідно	Записує значення для всієї площі даних
7	Використовує систему координат для збереження даних	Використовує сіткову (грид) систему для збереження даних
8	Масштабування (зумування) не змінює зовнішній вигляд	Масштабування може викликати ефект пікселізації («блочність»)
9	Складніше здійснювати прив'язку та накладання	Простіше здійснювати прив'язку та накладання
10	Сумісні з реляційними базами даних	Наявні деякі проблеми із сумісністю
11	Малий обсяг даних	Великий обсяг даних
12	Швидше відображення	Повільне відображення
13	Не диктує, як саме об'єкти мають виглядати в ГІС	Початково зберігає те, як об'єкти повинні виглядати в ГІС
14	Простіше оновлювати та підтримувати	Складніше оновлювати та підтримувати
15	Конвертація у растр є простою	Конвертація у вектор є складною
16	Підходить для дискретних геопросторових даних	Підходить для безперервних та дискретних даних
17	Менш ефективні для деяких видів просторового аналізу	Ефективні для просторового аналізу (растрової оверлейної обробки)

№	ВЕКТОРНІ ДАНІ	РАСТРОВІ ДАНІ
18	Трансформація проекції є простою та швидшою	Трансформація проекції потребує багато часу
19	Легко зберігати топологічні відношення	Складно зберігати топологічні відношення
20	Низька вартість друку	Висока вартість друку
21	Складна структура даних	Проста структура даних
22	Добре підходить для відображення геометричних об'єктів (точки, лінії, полігони)	Добре підходить для відображення зображень
23	Об'єктно-орієнтована модель даних	Полеорієнтована (континуальна) модель даних
24	Дані зазвичай збираються за допомогою традиційних методів знімання, GPS або екранного цифрування	Дані зазвичай збираються за допомогою фотографічного/сканувального обладнання або математико-статистичних методів (інтерполяція)
25	Обладнання для збору даних може бути дорогим і вимагати спеціальної підготовки	Обладнання для збору даних зазвичай недороге і може використовуватися з мінімальним навчанням або без нього

Ці параметри слід брати до уваги, обираючи між растровими та векторними даними для ГІС-проєкту. Зазвичай основними чинниками вибору формату даних є точність координат, швидкість обробки, вимоги до пам’яті та інші характеристики. Однак деякі дані, наприклад супутникові знімки, доступні лише в певному форматі, що не залишає користувачеві можливості вибору моделі даних. Важливим чинником може бути й простота методів збору даних, оскільки це один із найбільш трудомістких, марудних і дорогих процесів, вартість якого може сягати 50% від загального бюджету проєкту. У довготривалих ГІС-проєктах це завдання може посідати друге місце за важливістю після підтримки та оновлення даних. Загальновизнано, що растрові моделі даних найкраще підходять для картографування та аналізу ресурсів, тоді як векторні моделі полегшують мережевий аналіз.



Векторні	Растрові	Поєднання

Рис.2.1 Порівняння векторних і растрових моделей даних [30].

4. Інтегровані та тривимірні моделі просторових даних

Сучасний розвиток геоінформаційних систем та баз геоданих вимагає переходу від класичних двовимірних (2D) плоских карт до інтегрованих комплексних моделей, здатних відображати складні просторово-часові явища, геологічні структури, ґрунтові профілі та об'єкти штучного походження у тривимірному (3D) та чотиривимірному (4D = 3D + T) просторах.

Інтегровані моделі поєднують переваги об'єктного (векторного) та польового (растрового) підходів для забезпечення ефективного просторового аналізу [7; 12].

Нерегулярні тріангуляційні мережі (TIN) — векторне представлення поверхні

Векторне представлення даних про поверхню зазвичай використовується для відображення висот рельєфу. Положення кожної точки в наборі даних визначається трьома координатами (x, y, z). Точки можуть бути довільно розміщені в просторі, але з’єднані таким чином, щоб утворити трикутники мінімально можливого розміру. Кожна точка набору даних може бути вершиною одного або кількох трикутників. Отже, сусідні трикутники можуть мати спільні вершини, проте лінії, що з’єднують вершини й утворюють ребра різних трикутників, не повинні перетинатися. Побудова здійснюється шляхом проведення кіл через трійки точок: трикутник формується лише тоді, коли коло, описане навколо цих точок, не містить жодної іншої точки з набору даних. Така структура утворює нерегулярну мережу трикутників, де грань кожного трикутника відображає поверхню рельєфу. Для створення такої мережі трикутників можуть застосовуватися різні математичні підходи, однак отриманий набір даних є значно складнішим за растрову модель. Хоча моделі даних TIN можуть бути складними, завдяки векторному формату вони забезпечують ефективне візуальне відображення та потребують менше місця для зберігання даних. Вбудовані топологічні зв’язки в структурі TIN також роблять її придатною для геопросторового аналізу. Таким чином, TIN є ефективною альтернативою для представлення цифрової моделі рельєфу (ЦМР). У моделі TIN використовується підхід зі змінною щільністю даних, що дозволяє зберігати

вихідні точки при відображенні висот рельєфу. Це дає змогу описувати поверхню з різним рівнем деталізації: менші трикутники використовуються для ділянок із частими змінами висоти. Отже, TIN ефективно відображає різкі перепади висот, що може бути проблематичним для растрових моделей ЦМР; водночас ця модель менш придатна для представлення великих або територій складної конфігурації, а також може мати обмежені можливості візуалізації великих площ.

Масиви мільйонів дискретних точок з точними координатами (X, Y, Z) та додатковими атрибутами (інтенсивність відбиття, колір RGB, кількість відбитків), отримані методами [12]: Лазерного сканування (повітряного LiDAR, наземного або мобільного SLAM); Цифрової аерофотограмметрії з БПЛА.

Застосування інтегрованих та 3D -моделей в охороні ґрунтів

3D-моделювання ґрунтового профілю та горизонтів: побудова воксельних моделей розподілу фізико-хімічних властивостей (гумусу, щільності, кислотності) за глибинами (до 1.5–2.0).

Аналіз ерозійних процесів та яругоутворення: порівняльне 3D-моделювання на основі хмар точок LiDAR/БПЛА для розрахунку об'ємів винесеного ґрунту та швидкості росту ярів.

Цифрові двійники ландшафтів (Digital Twins): інтеграція 3D -моделей рельєфу, гідрографії, ґрунтового покриву та рослинності для моделювання поверхневого стоку, ризиків підтоплення та ерозійної деградації земель.

Контрольні питання

1. Що таке просторові дані та які їх основні типи?
2. У чому полягає сутність векторної моделі просторових даних?
3. Які геометричні примітиви використовуються у векторній моделі?
4. Для яких завдань у ГІС найдоцільніше застосовувати векторну модель?
5. Що таке растрова модель просторових даних і як вона формується?
6. Які переваги має растрова модель для аналізу безперервних процесів?
7. Які можливості відкривають тривимірні моделі просторових даних у дослідженні ґрунтів та екологічних процесів?

ТЕМА 3. АТРИБУТИВНІ ДАНІ ТА ЇХ ІНТЕГРАЦІЯ

План

1. Поняття атрибутивних даних
2. Джерела атрибутивних даних
3. Інтеграція атрибутивних даних у ГІС
4. Значення інтеграції атрибутивних даних для охорони ґрунтів

1. Поняття атрибутивних даних

Атрибутивні (семантичні або тематичні) дані — це якісні та кількісні характеристики географічних об'єктів, явищ або процесів, які описують їхні фізичні, хімічні, технологічні, економічні чи правові властивості, але не містять прямих відомостей про їхню просторову геометрію та координати [7; 14].

У геоінформаційних системах атрибутивні дані є невід'ємною складовою геоданих. Вони функціонують у нерозривному зв'язку з просторовою складовою (вектором або растром) через унікальні системні ідентифікатори (FID, ObjectID, Primary Key) [12].

Просторові дані можуть містити додаткову інформацію або непросторові дані, відомі як атрибути. Атрибут — це, як правило, інформація, що описує певний об'єкт. Просторові дані можуть мати будь-яку кількість атрибутів, що стосуються місця розташування (наприклад, карта, фотографії, історичні відомості тощо) [10].

Надаючи більше інформації про об'єкт, атрибути допомагають користувачам зрозуміти, де і чому відбуваються певні події. Наприклад, ГІС або інше спеціалізоване програмне забезпечення можуть створювати чи використовувати атрибути для доступу до географічного об'єкта, його візуалізації, маніпулювання ним або проведення просторового аналізу [10].

Часово-просторові дані — це дані, прив'язані до конкретного моменту часу (або часового інтервалу). Це можуть бути векторні точкові дані (наприклад, точка GPS-трекінгу) або растрові зображення (наприклад, знімок із супутника) [10].

Типи атрибутивних даних у базах геоданих

Для забезпечення цілісності та коректного збереження інформації у реляційних та об'єктно-реляційних СУБД (PostgreSQL/PostGIS, Oracle, SQLite) кожен атрибут описується чітким типом даних [14]:

Цілочисельні (INTEGER, BIGINT, SMALLINT): кількісні дискретні величини, коди ґрунтових відмінностей, кадастрові номери.

Дійсні числа з плаваючою/фіксованою крапкою (REAL, DOUBLE PRECISION, NUMERIC(p,s)): фізико-хімічні показники ґрунту з високою точністю (наприклад, NUMERIC(4,2) для pH).

Текстові / Символьні (VARCHAR(n), TEXT): морфологічні описи ґрунтових профілів, назви землекористувачів, класифікаційні шифри.

Часові (DATE, TIME, TIMESTAMP): дати проведення агрохімічного обстеження, час фіксації показників датчиками вологості.

Логічні (BOOLEAN): двозначні прапорці стану (наприклад, is_eroded = TRUE/FALSE).

Напівструктуровані (JSON / JSONB): збереження гнучких метаданих лабораторних аналізів, результатів спектрометрії або специфічних стандартів ISO.

Чому атрибутивні дані важливі для аналізу ГІС та прийняття рішень?

Атрибутні дані перетворюють основні географічні об'єкти на потужні аналітичні інструменти, забезпечуючи описовий контекст, необхідний для прийняття обґрунтованих рішень. Без атрибутивної інформації просторовий аналіз обмежувався б виключно геометричними зв'язками. Атрибути дозволяють виконувати складні запити, статистичний аналіз та аналіз на основі даних, що підтримує стратегічне планування та операційну ефективність у різних галузях [31].

Аналітична сила виникає, коли ви поєднуєте просторові зв'язки з характеристиками атрибутів. Ви можете виявити закономірності, тенденції та кореляції, які не були б видимими лише за допомогою просторових даних. Наприклад, аналіз даних про скарги клієнтів разом із атрибутами мережевої інфраструктури допомагає комунальним підприємствам визначати пріоритети робіт з технічного обслуговування та прогнозувати проблеми з обслуговуванням [31].

Атрибутні дані впливають на вибір візуалізації та картографування, визначаючи, як об'єкти символізуються, забарвлюються та позначаються. Атрибути щільності населення контролюють кольори хороплефа, тоді як атрибути типу дороги визначають символи та ширину ліній. Це візуальне представлення допомагає ефективно передавати складну інформацію зацікавленим сторонам та особам, які приймають рішення [31].

Реальні застосування демонструють цінність атрибутивних даних у багатьох секторах. Водоканали використовують атрибути віку та матеріалу труб для оцінки пріоритетів заміни та розрахунку ймовірності відмови. Постачальники енергії аналізують моделі споживання споживачів разом із даними про пропускну здатність мережі для оптимізації планування розподілу. Урядові установи поєднують демографічні атрибути з межами зони обслуговування для ефективного розподілу ресурсів [31].

Наші рішення для просторового аналізу ефективно синтезують детальні атрибутивні дані в практичну інформацію, додаючи маршрутизацію, топологію та просторові зв'язки до комплексного аналізу. Такий підхід дозволяє розпізнавати закономірності, виявляти тенденції та оцінювати ризики, що підтримує проактивні стратегії управління та обґрунтовані інвестиції в інфраструктуру[31].

Розуміння ролі атрибутивних даних у ГІС забезпечує основу для ефективного використання геоінформаційних систем. За умови правильного управління та аналізу атрибутивні дані перетворюють просторові об'єкти з простих форм на комплексні інформаційні ресурси, які сприяють прийняттю кращих рішень та покращенню результатів. У Spatial Eye ми допомагаємо організаціям максимізувати цінність своїх атрибутивних даних за допомогою інтегрованих аналітичних рішень, які поєднують описові характеристики з

просторовими зв'язками для отримання комплексної географічної інформації [31].

2. Джерела атрибутивних даних

Джерела атрибутивних даних у геоінформаційних системах охоплюють комплекс первинних інструментальних вимірювань, лабораторних аналізів, фондових картографічних матеріалів та міжнародних цифрових реєстрів [7; Longley et al., 2015].

У сфері землевпорядкування та охорони ґрунтів формування атрибутивного наповнення баз геоданих спирається на чотири основні групи джерел.

Основоположні джерелf атрибутивної інформації

– Агрохімічне обстеження ґрунтів: лабораторні аналізи відібраних точкових чи змішаних проб (вміст гумусу, рухомих форм азоту, фосфору, калію, pH сольової та водної витяжок, вміст важких металів та пестицидів).

– Морфологічний опис ґрунтових розрізів: польові журналы з атрибутами ґрунтових горизонтів (потужність горизонту, гранулометричний склад, структура, забарвлення, наявність новоутворень).

– Дані сенсорних мереж та IoT: автоматизовані датчики безперервно фіксують вологість, температуру та електропровідність ґрунту на різних глибинах у режимі реального часу.

– Державний земельний кадастр (ДЗК): містить атрибутивні відомості про цільове призначення, категорію земель, формальні межі, кадастровий номер, правовий статус та обмеження у використанні земельних ділянок.

– Матеріали суцільних великомасштабних ґрунтових обстежень: фондові карти та опис ґрунтових відмінностей (масштабів 1:10 000 – 1:25 000), які є базою для порівняльного аналізу деградаційних процесів.

– Агрохімічні паспорти полів та земельних ділянок: офіційні документи, що регламентують рівень родючості та ступінь екологічної безпеки ґрунтового покриву.

– Дані бонітування та грошової оцінки земель: атрибутивні бали бонітету, індекси придатності ґрунтів під різні сільськогосподарські культури.

– Та інші

Стандартизація та гармонізація атрибутивних даних

Для забезпечення сумісності (Interoperability) атрибутивних даних з різних джерел застосовуються міжнародні класифікаційні регламенти [2]:

Стандарт ISO 28258:2013 («Soil quality — Digital exchange of soil information»): визначає концептуальну модель даних (UML) для цифрового обміну інформацією про ґрунти, включаючи структури для ґрунтових профілів, шарів, горизонтів та методів вимірювань [15].

Специфікації INSPIRE (Data Specification on Soil): європейський стандарт структуризації баз геоданих для екологічного моніторингу та охорони земель.

При інтеграції різнорідних джерел обов'язковими етапами є нормалізація одиниць вимірювання (наприклад, перерахунок мг/кг у мг/100г ґрунту) та

стандартизація кодових класифікаторів (перехід від національних класифікацій ґрунтів до міжнародної системи WRB).

3. Інтеграція атрибутивних даних у ГІС

Інтеграція атрибутивних даних у ГІС - це процес просторово-логічного поєднання табличної семантичної інформації з геометричними об'єктами (точечними, лінійними, полігональними або растровими) для забезпечення комплексного аналізу, візуалізації та просторового моделювання [7, 12].

Механізми зв'язування просторових та атрибутивних даних

У базах геоданих та ГІС-пакетах (QGIS, ArcGIS Pro, PostgreSQL/PostGIS) використовуються три основні класи зв'язування таблиць [29]:

Метод інтеграції	Кардинальність (1:1,1:M,M:N)	Механізм реалізації	Особливості та застосування
Атрибутивне об'єднання (Attribute Join)	1:1 або M:1	Фізичне або віртуальне об'єднання таблиць за спільним кодовим полем (Primary Key = Foreign Key).	Атрибути зовнішньої таблиці додаються безпосередньо до таблиці просторового шару. *Приклад:* об'єднання полігонів полів із таблицею сівозміни за Field_ID.
Атрибутивна зв'язка (Relate / Relationship Class)	1:M або M:N	Логічне связування без об'єднання таблиць у єдину структуру.	Дозволяє звертатися до кількох підпорядкованих записів. *Приклад:* до одного контуру поля (1) прив'язано серію лабораторних аналізів ґрунту за різні роки (M).
Просторова інтеграція (Spatial Join)	1:1 , 1:M , M:1	Зв'язування атрибутів на основі топологічного відношення геометрій (DE-9IM предикати: ST_Intersects, ST_Within, ST_Contains).	Не вимагає наявності спільних текстових/числових ключів. *Приклад:* передача атрибутів ґрунтового відмінення точечним пробам відбору, які потрапляють у його контур.

Контрольні питання

1. Що таке атрибутивні дані та чим вони відрізняються від просторових?
2. Які приклади якісних та кількісних атрибутивних даних можна навести?

3. Які основні джерела атрибутивних даних використовуються у ґрунтозахисних дослідженнях?

4. Як лабораторні аналізи та польові вимірювання формують атрибутивні дані?

5. Яку роль відіграють державні кадастрові системи у формуванні атрибутивних даних?

6. Як здійснюється інтеграція атрибутивних даних у ГІС?

7. Які можливості відкриває поєднання геометрії об’єктів із їх атрибутивними характеристиками?

8. Яке значення має інтеграція атрибутивних даних для аналізу деградаційних процесів ґрунтів?

9. Чому атрибутивні дані у поєднанні з просторовими є ключовим інструментом для сталого землекористування?

ТЕМА 4. ПРОЕКТУВАННЯ БАЗ ДАНИХ

План

1. Основи проектування баз даних
2. Концептуальне та логічне моделювання
3. Нормалізація баз даних
4. Фізичне проектування та оптимізація
5. Проектування баз даних для охорони ґрунтів

1. Основи проектування баз даних

Проектування баз даних - це системний процес прийняття послідовних інженерно-технічних та методологічних рішень щодо створення структури бази даних, її концептуального узгодження з вимогами предметної області, забезпечення цілісності, оптимізації зберігання та швидкого доступу до інформації [8; 14].

У контексті геоінформаційних систем (ГІС) та земельно-інформаційних систем (ЗІС) проектування додатково враховує подвійну природу геоданих: поєднання складних векторно-просторових об'єктів (з їхніми топологічними та просторовими прив'язками) і якісно-кількісних атрибутивних характеристик [7; 12].

Основні кроки планування бази даних такі:

1. Збір інформації
2. Визначення ключових об'єктів або сутностей
3. Моделювання ключових об'єктів
4. Визначення типів інформації для кожного об'єкта або сутності
5. Визначення зв'язків між об'єктами або сутностями

Проєктування бази даних — це процес створення детальної моделі даних бази даних. Ця логічна модель даних містить усі необхідні рішення щодо логічного та фізичного проєктування, а також параметри фізичного зберігання, потрібні для формування опису мовою визначення даних (DDL), яку згодом можна використати для створення самої бази даних. Повноцінна модель даних містить детальні атрибути для кожної сутності.

Термін «проєктування бази даних» може вживатися для опису різних аспектів розробки всієї системи бази даних. Насамперед — і це найбільш точне визначення — його можна розглядати як логічне проєктування базових структур даних, що використовуються для їх зберігання. У реляційній моделі такими структурами є таблиці та представлення (views). В об'єктній базі даних сутності та зв'язки безпосередньо відображаються на класи об'єктів та іменовані зв'язки. Однак цей термін може також стосуватися загального процесу розробки, що охоплює не лише базові структури даних, а й форми та запити, які є частиною прикладного застосунку бази даних у межах системи управління базами даних (СУБД).

Процес проєктування бази даних зазвичай складається з низки етапів, які виконує проєктувальник бази даних. Як правило, проєктувальник повинен:

- Визначити зв'язки між різними елементами даних.

- Сформувати логічну структуру даних на основі цих зв'язків.

Етапи проєктування бази даних

Процес проєктування бази даних поділяється на окремі частини та складається з низки етапів, а саме:

· Аналіз вимог
· Концептуальне проєктування бази даних (ER-діаграма)
· Логічне проєктування бази даних (таблиці, нормалізація тощо)
· Фізичне проєктування бази даних (індексування таблиць, кластеризація тощо)

Аналіз вимог. На цьому етапі проводиться детальний аналіз вимог. Мета етапу — сформувати чітке розуміння вимог. Для цього використовуються різні методи збору інформації, зокрема:

· Інтерв'ю
· Аналіз документів
· Опитування
· Відвідування об'єкта (робочого місця)
· Спільне проєктування застосунків (JAD) та спільний аналіз вимог (JRA)
· Прототипування.

Концептуальне проєктування бази даних. Результати аналізу вимог відображаються в концептуальній моделі. На цьому етапі використовується модель «сутність-зв'язок» (ER-модель). Для візуалізації концептуального проєкту застосовується ER-діаграма, що складається із сутностей, атрибутів і зв'язків.

Логічне проєктування бази даних. Після визначення зв'язків і залежностей дані організовуються в логічні структури та відображаються у вигляді таблиць системи управління базами даних (СУБД). Виконується нормалізація для приведення відношень до відповідних нормальних форм.

Фізичне проєктування бази даних. Цей етап стосується фізичної реалізації бази даних у конкретній СУБД. Він включає визначення елементів даних, типів даних, індексації тощо. Уся ця інформація зберігається в словнику даних.

Концептуальна модель відображає бачення та розуміння даних у контексті реального світу; логічна модель являє собою узагальнену формальну структуру, що відповідає правилам інформатики; фізична модель визначає спосіб реалізації цієї структури в конкретному екземплярі СУБД.

Різні методології та інструменти моделювання даних забезпечують ці рівні абстракції по-різному. Деякі з них зосереджені лише на фізичній реалізації, інші моделюють лише логічну структуру, а треті можуть поєднувати елементи всіх трьох рівнів, не обов'язково розділяючи їх на окремі представлення. У будь-якому разі це допомагає фахівцеві з моделювання даних зрозуміти, до якого рівня абстракції належить конкретна функція чи завдання.

Цілі та завдання проектування баз даних

Ключовими завданнями ефективного проектування є [8; 14]:

1. Мінімізація виразності та дублювання: виключення дублювання атрибутивних та геометричних даних для зменшення обсягу пам'яті та відсутності аномалій оновлення.

2. Забезпечення цілісності даних: підтримка узгодженості інформації через систему декларативних обмежень (Primary Key, Foreign Key, NOT NULL, CHECK) та топологічних правил.

3. Незалежність даних від додатків: забезпечення можливості модернізації фізичної структури збереження без зміни бізнес-логіки та користувацьких запитів.

4. Масштабованість та ефективність: створення оптимальних схем таблиць та індексних структур для забезпечення високої швидкості обробки великих обсягів просторово-часових даних.

Специфіка проектування баз геоданих.

При проектуванні баз геоданих виникають специфічні вимоги, що відрізняють їх від традиційних обліково-фінансових СУБД [7, 12]:

1. Підтримка складних геометричних типів: необхідність збереження точок, ліній, полігонів, мультиполігонів, хмар точок та растрових масивів відповідно до стандартів OGC.

2. Топологічні правила та відношення: впровадження обмежень цілісності на рівні геометрії (наприклад, полігони земельних ділянок не повинні перекриватися, а контури ґрунтових відмінностей повинні покривати векторний шар без прогалин).

3. Просторова індексація: використання деревних та багатовимірних індексів для прискорення просторових операцій перетину, включення та визначення близькості.

4. Просторово-часовий вимір: забезпечення збереження динаміки часових змін (наприклад, щорічні зміни вмісту гумусу на конкретній земельній ділянці).

2. Концептуальне та логічне моделювання

Моделювання баз даних є серцевиною процесу проектування, яка забезпечує перехід від реальних об'єктів та явищ навколишнього середовища до їхнього систематизованого цифрового представлення [14; 12].

У базах геоданих цей процес ускладнюється необхідністю моделювання як семантичних (атрибутивних) характеристик, так і складних тополого-просторових відношень між геопросторовими об'єктами [7].

Концептуальне моделювання даних – це один із трьох типів моделювання даних, також відомих як етапи моделювання даних. Цей процес включає створення концептуальних моделей даних, які допомагають організаціям візуалізувати бази даних, що призводить до більш ефективного та комплексного побудови баз даних [22].

Перехід від концептуальних до логічних та фізичних моделей даних

Дотримуючись найкращих практик моделювання даних, організації спочатку розробляють концептуальні моделі, які потім уточнюються до логічних

і, нарешті, фізичних моделей даних. Кожен етап додає нові рівні абстракції та деталізації, враховуючи різні бізнес-вимоги [22].

Концептуальна модель даних — це модель високого рівня, яка пропонує глибоке уявлення про бізнес-концепції. Вона відіграє важливу роль у визначенні ключових бізнес- та системних сутностей і встановленні зв'язків між ними. Ці моделі зазвичай не залежать від платформи, що забезпечує гнучкість у виборі систем керування базами даних (СКБД) або методологій зберігання даних на подальших етапах розробки [22].

Платформно-незалежний характер концептуальних моделей даних дозволяє організаціям мислити за межами своїх поточних технологічних можливостей. Цей етап надає можливість розробити ідеальну базу даних, яка відповідає потребам бізнесу, виявляючи будь-які прогалини в існуючих інструментах або можливостях. Це також може допомогти виправдати інвестиції в нові технології [22].

Отримання найбільшої користі від моделювання даних вимагає лінійного просування через усі три етапи моделювання даних. Початок з концептуальної моделі – це найнадійніший спосіб гарантувати, що готове рішення для бази даних відповідає бізнес-вимогам.

Концептуальні моделі даних є важливими для визначення ключових сутностей, їхніх атрибутів та зв'язків між ними, що є критично важливим для прийняття рішень та стратегічного планування [22]:

- Сприяння комунікації: ці моделі забезпечують спільне розуміння вимог до даних, зменшуючи ризик невідповідності на пізніших етапах проектування бази даних.
- Керівництво розробкою: Вони закладають основу для логічних та фізичних моделей даних, забезпечуючи, щоб процес розробки залишався зосередженим на бізнес-цілях.

Концептуальна модель даних є найбільш абстрактним типом моделі даних. Одним із поширених підходів до моделювання даних є метод «сутність/зв'язок» (ER), який формує основу для моделей даних.

Основні нотації концептуального моделювання:

- ER-діаграми (Entity-Relationship Diagrams / Модель Пітера Чена): класичний підхід моделювання через сутності, атрибути та зв'язки.
- UML-діаграми класів (Unified Modeling Language): сучасний об'єктно-орієнтований стандарт для проектування складних геопросторових систем, рекомендований ISO та OGC [2, 15, 27].

Моделювання даних

Модель даних – це інтегрована сукупність понять для опису даних, зв’язків між ними та обмежень, що на них накладаються, а також для маніпулювання цими даними в межах організації [14].

Модель – це представлення об’єктів і подій «реального світу» та взаємозв’язків між ними. Це абстракція, що зосереджується на суттєвих, притаманних організації аспектах і не враховує другорядні властивості. Модель даних відображає саму організацію. Вона має надавати базові поняття та засоби нотації, які дозволять проєктувальникам баз даних і кінцевим користувачам

однозначно й точно обмінюватися розумінням даних організації. Модель даних можна розглядати як таку, що складається з трьох компонентів [14].

1) структурна частина, що містить набір правил побудови баз даних;

2) маніпуляційна частина, що визначає типи дозволених операцій із даними (сюди входять операції оновлення чи отримання даних із бази, а також зміни структури самої бази даних);

3) за потреби – набір обмежень цілісності, що гарантують точність і правильність даних.

Мета моделі даних – представити дані та зробити їх зрозумілими. Якщо модель виконує це завдання, її можна легко використовувати для проєктування бази даних [14].

Використання діаграм «сутність–зв’язок» (ER-діаграм). Часто легше візуалізувати складну систему, ніж розшифровувати довгі текстові описи вимог користувачів. Ми використовуємо діаграми «сутність–зв’язок» (ER-діаграми) для наочного представлення сутностей та зв’язків між ними. На етапі проєктування бази даних ми рекомендуємо використовувати ER-діаграми щоразу, коли це необхідно для формування цілісного уявлення про ту частину підприємства, яку ми моделюємо.

Тип сутності – це група об’єктів з однаковими властивостями, які підприємство розглядає як такі, що мають незалежне існування.

Основним поняттям ER-моделі є тип сутності, що представляє групу об’єктів «реального світу», які мають однакові властивості. Тип сутності має незалежне існування; це можуть бути об’єкти, що існують фізично (або «реально»), чи об’єкти, що існують концептуально (або «абстрактно»). Зауважте, що ми можемо навести лише робоче визначення типу сутності, оскільки суворого формального визначення не існує. Це означає, що різні проєктувальники можуть виділяти різні сутності [14].

Екземпляр сутності. Унікально ідентифікований об'єкт типу сутності. Кожен унікально ідентифікований об'єкт типу сутності називається просто екземпляром сутності. У цьому методичному посібнику ми використовуємо терміни «тип сутності» або «екземпляр сутності»; однак там, де зміст є очевидним, ми вживаємо більш загальний термін «сутність». Ми ідентифікуємо кожен тип сутності за допомогою назви та переліку властивостей. База даних зазвичай містить багато різних типів сутностей [14].

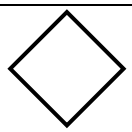
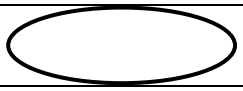
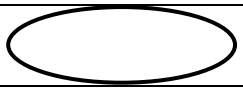
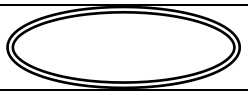
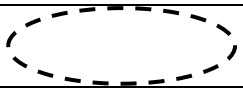
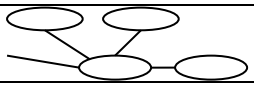
Тип зв'язку – це сукупність асоціацій між одним або кількома типами сутностей, що беруть у ньому участь. Кожен тип зв'язку має назву, що описує його функцію. Як і у випадку з типами сутностей та сутностями, необхідно розрізняти терміни «тип зв'язку» та «екземпляр зв'язку» [14].

Екземпляр зв'язку. Унікально ідентифікована асоціація, що включає по одному екземпляру кожного з типів сутностей, які беруть участь у зв'язку.

Атрибут. Властивість сутності або типу зв'язку.

Конкретні властивості типів сутностей називаються атрибутами.

Тип зв'язку, що пов'язує сутності, також може мати атрибути, подібні до атрибутів типу сутності. У цьому розділі ми зосередимося на загальних характеристиках атрибутів.

Нотація для діаграми Сутність-Зв’язок	
Символ	**Значення**
	Сутність
	Слабка Сутність
	Зв’язок
	Атрибут
	Ключовий атрибут
	Багатозначний атрибут
	Похідний атрибут
	Складений атрибут

Домен атрибута. Множина допустимих значень для одного або кількох атрибутів [14].

З кожним атрибутом пов'язана множина значень, що називається доменом. Домен визначає потенційні значення, які може набувати атрибут, і є аналогом поняття домену в реляційній моделі. Наприклад, вміст гумусу в грунтах, для кожного екземпляра сутності становить від 0 до 10%.

Визначити домен атрибута «назва» складніше, оскільки він охоплює всі можливі назви. Безумовно, це символьний рядок, але він може містити не лише літери, а й дефіси чи інші спеціальні символи. Повноцінна модель даних включає визначення доменів для кожного атрибута ER-моделі [14].

Атрибути можна класифікувати як: прості або складені; однозначні або багатозначні; а також похідні [14].

Простий атрибут. Атрибут, що складається з одного компонента, який має самостійне існування.

Складений атрибут. Атрибут, що складається з кількох компонентів, кожен з яких має самостійне існування.

Однозначний атрибут. Атрибут, що має одне значення для кожного екземпляра типу сутності.

Багатозначний атрибут. Атрибут, що має кілька значень для кожного екземпляра типу сутності.

Похідний атрибут. Атрибут, значення якого можна отримати на основі значення пов’язаного атрибута або набору атрибутів (не обов’язково в межах того самого типу сутності).

Ключі

Кандидат у ключі (або потенційний ключ). Мінімальний набір атрибутів, що однозначно ідентифікує кожен екземпляр типу сутності.

Кандидат у ключі – це мінімальний набір атрибутів, значення яких однозначно ідентифікують кожен екземпляр сутності [14].

Кандидат у ключі повинен мати значення, унікальні для кожного екземпляра типу сутності. Це означає, що кандидат у ключі не може містити невизначене значення (null).

Первинний ключ. Кандидат у ключі, обраний для однозначної ідентифікації кожного екземпляра типу сутності.

Тип сутності може мати більше одного кандидата у ключі.

Вибір первинного ключа для сутності ґрунтується на таких чинниках, як довжина атрибута, мінімальна необхідна кількість атрибутів та впевненість у збереженні унікальності в майбутньому.

Складений ключ. Кандидат у ключі, що складається з двох або більше атрибутів.

У деяких випадках ключ типу сутності складається з кількох атрибутів, значення яких у сукупності є унікальними для кожного екземпляра сутності, хоча окремо вони такими не є.

Типи сутностей можна класифікувати як сильні або слабкі [14].

Сильний тип сутності. Тип сутності, існування якого не залежить від іншого типу сутності.

Тип сутності вважається сильним, якщо його існування не залежить від існування іншого типу сутності. Характерною рисою сильного типу сутності є те, що кожен екземпляр сутності можна однозначно ідентифікувати за допомогою атрибута (або атрибутів) первинного ключа цього типу сутності.

Слабкий тип сутності. Тип сутності, існування якого залежить від іншого типу сутності.

Слабкий тип сутності залежить від існування іншого типу сутності. Характерною рисою слабкої сутності є те, що кожен її екземпляр неможливо однозначно ідентифікувати, використовуючи лише атрибути, пов’язані з цим типом сутності. Слабкі типи сутностей іноді називають дочірніми, залежними або підпорядкованими сутностями, а сильні типи сутностей — батьківськими, власниками або домінуючими сутностями.

Типи зв’язків.

Кардинальність – це узагальнена міра кількості сутностей (однієї або більше), що пов’язані з іншою сутністю (або сутностями). Наприклад, існує чотири способи, у які сутності «ЗЕМЕЛЬНА ДІЛЯНКА» та «ЗЕМЛЕВЛАСНИК» можуть бути «чисельно залучені» до зв’язку: «один-до-одного» (1:1), «багато-до-одного» (M:1), «один-до-багатьох» (1:M) та «багато-до-багатьох» (M:N) [14].

Один-до-одного (1:1). Кожен запис у першій таблиці відповідає чітко одному запису в другій таблиці, і навпаки. Наприклад, одна земельна ділянка має лише один кадастровий номер, і один кадастровий номер присвоюється тільки одній ділянці (ДІЛЯНКА : КАДАСТРОВИЙ_НОМЕР :: 1:1).

Один-до-багатьох (1:M). Один запис із першої таблиці може бути пов'язаний із кількома записами в другій, але кожен запис із другої таблиці посилається лише на один з першої. Наприклад, один власник може володіти кількома земельними ділянками, але кожна окрема ділянка належить тільки цьому одному власникові (ВЛАСНИК : ДІЛЯНКА :: 1:M).

Багато-до-одного (M:1). Кілька записів із першої таблиці посилаються на один і той самий запис у другій таблиці (це зворотна сторона зв’язку 1:M). Наприклад, багато окремих земельних ділянок розташовані в межах однієї

територіальної громади, але конкретна ділянка належить лише до однієї громади (ДІЛЯНКА : ГРОМАДА :: M:1).

Багато-до-багатьох (M:N). Численні екземпляри однієї сутності пов’язані з численними екземплярами іншої, причому кількість об'єктів з обох боків не обов’язково однакова ($M \neq N$). Наприклад, кілька співвласників спільно володіють кількома земельними ділянками, де кожен із власників має частку в кожній із цих ділянок (СПІВВЛАСНИК : ДІЛЯНКА :: M:N).

Логічне моделювання даних – це наступний етап моделювання, який використовується для визначення структури елементів даних та встановлення зв'язків між ними. Логічні моделі додають більше деталей до концептуальної моделі, але все ще є по суті загальними та не вказують на певну платформу реалізації. Логічні моделі використовуються як основа для створення фізичних моделей даних [14].

Логічне моделювання даних слугує мостом між бізнес-концепціями та технічною реалізацією. Воно перетворює високорівневі бізнес-вимоги на детальні, технологічно незалежні моделі, які визначають, як інформація пов'язана в межах підприємства. Фіксуючи сутності, атрибути, зв'язки, бізнес-правила та ключові обмеження перед вибором платформи бази даних, логічні моделі забезпечують спільний план, який узгоджує дії зацікавлених сторін бізнесу, архітекторів даних та команд розробників.

Логічне моделювання даних – це процес визначення структури бізнес-інформації незалежно від будь-якої конкретної технології баз даних. Воно фіксує дані, необхідні організації, те, як різні бізнес-суб'єкти пов'язані один з одним, та правила, що регулюють ці взаємозв'язки.

На відміну від фізичного дизайну бази даних, логічна модель даних не визначає деталі реалізації, такі як таблиці, індекси чи структури сховища. Натомість вона зосереджена на точному представленні бізнес-інформації таким чином, щоб її могли зрозуміти як технічні, так і бізнес-зацікавлені сторони.

Логічна модель даних зазвичай включає [23]:

- Суб'єкти господарювання
- Атрибути, що описують ці сутності
- Зв'язки між сутностями
- Первинні, зовнішні та альтернативні ключі
- Бізнес-правила та обмеження
- Вимоги до даних та визначення

Оскільки логічні моделі залишаються платформонезалежними, організації можуть удосконалювати бізнес-вимоги, не прив'язуючись до Oracle, SQL Server, PostgreSQL, Snowflake чи будь-якої іншої платформи баз даних. Така гнучкість дозволяє командам оцінювати різні варіанти впровадження, забезпечуючи при цьому узгодженість базової бізнес-логіки [23].

На цьому етапі абстрактні сутності перетворюються на відношення (таблиці), атрибути – на стовпчики з чітко визначеними типами даних, а зв'язки - на систему первинних (Primary Key – PK) та зовнішніх (Foreign Key – FK) ключів.

Правила трансформації ER / UML у Реляційну схему геобази [15]:

Крок 1: Відображення типів регулярних сутностей. Для кожного типу регулярної (сильної) сутності E в ER-схемі створіть відношення R, що включає всі прості атрибути сутності E. Включайте лише прості складові атрибути складеного атрибута. Оберіть один із ключових атрибутів сутності E як первинний ключ для відношення R. Якщо обраний ключ сутності E є складеним, то набір простих атрибутів, що його утворюють, разом формуватимуть первинний ключ відношення R [14].

Якщо під час концептуального проектування для сутності E було визначено кілька ключів, інформація про атрибути, що утворюють кожен додатковий ключ, зберігається для визначення вторинних (унікальних) ключів відношення R. Відомості про ключі також зберігаються для цілей індексації та інших видів аналізу.

Крок 2: Відображення типів слабких сутностей. Для кожного типу слабкої сутності W в ER-схемі, що має тип сутності-власника E, створіть відношення R і включіть усі прості атрибути (або прості складові складених атрибутів) сутності W як атрибути відношення R. Крім того, включіть до складу атрибутів відношення R (як зовнішній ключ) атрибут(и) первинного ключа відношення (або відношень), що відповідає типу сутності-власника; це забезпечує відображення типу ідентифікуючого зв'язку для сутності W. Первинним ключем відношення R є комбінація первинного ключа (або ключів) власника та часткового ключа типу слабкої сутності W (за наявності останнього). Якщо існує тип слабкої сутності E2, власником якого також є тип слабкої сутності E1, то тип E1 слід відобразити раніше за E2, щоб спочатку визначити його первинний ключ [14].

Крок 3: Відображення типів бінарних зв'язків «один-до-одного» (1:1). Для кожного типу бінарного зв'язку 1:1 (позначимо його R) в ER-схемі визначте відношення S і T, що відповідають типам сутностей, які беруть участь у зв'язку R. Існує три можливі підходи: (1) підхід із використанням зовнішнього ключа, (2) підхід із об'єднанням зв'язку та (3) підхід із використанням перехресних посилань або окремого відношення для зв'язку. Перший підхід є найбільш доцільним; його слід застосовувати, якщо немає особливих умов, що вимагають іншого рішення (про це йдеться далі). 1. Підхід із використанням зовнішнього ключа: Оберіть одне з відношень — скажімо, S — і включіть до нього первинний ключ відношення T як зовнішній ключ. У ролі S краще обирати тип сутності, що має повну участь у зв’язку R. Включіть усі прості атрибути (або прості компоненти складених атрибутів) типу зв’язку 1:1 (R) як атрибути відношення S [14].

Зауважте, що замість цього можна включити первинний ключ відношення S як зовнішній ключ до відношення T. Ще один варіант — розмістити зовнішні ключі в обох відношеннях (S і T) одночасно, однак це створює надмірність даних і ускладнює підтримання їхньої узгодженості.

2. Підхід з об’єднанням відношень: Альтернативним способом відображення типу зв’язку 1:1 є об’єднання двох типів сутностей і самого зв’язку в одне відношення. Це можливо, коли участь обох сутностей є повною, оскільки це свідчить про те, що обидві таблиці завжди матимуть однакову кількість кортежів.

3. Підхід із використанням відношення зв'язку (або перехресного посилання): Третій варіант полягає у створенні окремого відношення R для пов'язування первинних ключів двох відношень (S і T), що представляють типи сутностей. Як ми побачимо далі, такий підхід є обов'язковим для бінарних зв'язків типу M:N. Відношення R називають відношенням зв'язку (або іноді таблицею пошуку), оскільки кожен кортеж у ньому представляє екземпляр зв'язку, що поєднує один кортеж із S з одним кортежем із T. Відношення R міститиме атрибути первинних ключів відношень S і T як зовнішні ключі, що посилаються на S і T. Первинним ключем R буде один із цих двох зовнішніх ключів, а інший зовнішній ключ слугуватиме унікальним ключем R. Недоліком цього підходу є поява додаткового відношення та необхідність виконання додаткової операції з'єднання (join) під час об'єднання пов'язаних кортежів із таблиць.

Крок 4: Відображення бінарних типів зв'язку 1:N. Для кожного звичайного бінарного типу зв'язку 1:N (R) визначте відношення S, що представляє тип сутності, який бере участь у зв'язку з боку «N». Включіть до відношення S як зовнішній ключ первинний ключ відношення T, що представляє інший тип сутності, залучений до зв'язку R; це робиться тому, що кожен екземпляр сутності з боку «N» пов'язаний щонайбільше з одним екземпляром сутності з боку «1» цього типу зв'язку. Включіть усі прості атрибути (або прості складові складених атрибутів) типу зв'язку 1:N до складу атрибутів відношення S [14].

Альтернативний підхід полягає у використанні варіанта з окремим відношенням для зв'язку (перехресним посиланням), аналогічного третьому варіанту для бінарних зв'язків 1:1. У цьому разі створюється окреме відношення R, атрибутами якого є первинні ключі відношень S і T; ці атрибути також слугують зовнішніми ключами, що посилаються на S і T. Первинним ключем відношення R є первинний ключ відношення S. Цей варіант доцільно застосовувати, якщо у зв'язку бере участь лише невелика кількість кортежів відношення S, щоб уникнути надмірної кількості значень NULL у зовнішньому ключі.

Крок 5: Відображення бінарних типів зв'язків M:N. Для кожного бінарного типу зв'язку M:N (R) створіть нове відношення S для представлення R. Включіть до S як атрибути зовнішнього ключа первинні ключі відношень, що представляють типи сутностей-учасників; їхня комбінація утворить первинний ключ відношення S. Також включіть до S будь-які прості атрибути типу зв'язку M:N (або прості компоненти складених атрибутів). Зауважте, що ми не можемо представити тип зв'язку M:N за допомогою одного атрибута зовнішнього ключа в одному з відношень-учасників (як це робилося для типів зв'язків 1:1 або 1:N) через кардинальність M:N; необхідно створити окреме відношення S для цього зв'язку [14].

Крок 6: Відображення багатозначних атрибутів. Для кожного багатозначного атрибута A створіть нове відношення R. Це відношення R міститиме атрибут, що відповідає A, а також атрибут первинного ключа K (як зовнішній ключ у R) відношення, яке представляє тип сутності або тип зв'язку, що має A як багатозначний атрибут. Первинним ключем R є комбінація A та K.

Якщо багатозначний атрибут є складеним, включаються його прості компоненти [14].

Крок 7: Відображення n-арних типів зв’язків. Для кожного n-арного типу зв’язку R (де n > 2) створіть нове відношення S для представлення R. Включіть до S як атрибути зовнішнього ключа первинні ключі відношень, що представляють типи сутностей-учасників. Також включіть до S будь-які прості атрибути n-арного типу зв’язку (або прості компоненти складених атрибутів). Первинним ключем S зазвичай є комбінація всіх зовнішніх ключів, що посилаються на відношення, які представляють типи сутностей-учасників. Однак, якщо обмеження кардинальності для будь-якого типу сутності E, що бере участь у R, дорівнює 1, то первинний ключ S не повинен включати атрибут зовнішнього ключа, що посилається на відношення E’, відповідне сутності E [14].

Відповідність між ER-моделлю та реляційною моделлю

ER-модель (ER Model)	Реляційна модель (Relational Model)
Тип сутності (*Entity type*)	Відношення «сутність» (*"Entity" relation* / таблиця)
Тип зв'язку 1:1 або 1:N (*1:1 or 1:N relationship type*)	Зовнішній ключ (або відношення «зв'язок») (*Foreign key / "relationship" relation*)
Тип зв'язку M:N (*M:N relationship type*)	Відношення «зв'язок» та два зовнішні ключі (*"Relationship" relation and two foreign keys*)
N-арний тип зв'язку (*N-ary relationship type*)	Відношення «зв'язок» та n зовнішніх ключів (*"Relationship" relation and n foreign keys*)
Простий атрибут (*Simple attribute*)	Атрибут (*Attribute* / поле)
Складений атрибут (*Composite attribute*)	Набір простих компонентних атрибутів (*Set of simple component attributes*)
Багатозначний атрибут (*Multivalued attribute*)	Відношення та зовнішні ключі (*Relation and foreign keys*)
Множина значень (*Value set*)	Домен (*Domain*)
Ключовий атрибут (*Key attribute*)	Первинний (або вторинний) ключ (*Primary (or secondary) key*)

3. Нормалізація баз даних

Нормалізація – це формалізований математично обґрунтований процес декомпозиції відношень (таблиць) бази даних на основі функціональних залежностей їхніх атрибутів. Головною метою нормалізації є усунення надмірності (дублювання) даних, забезпечення цілісності та запобігання аномаліям модифікації [11; 8].

Простіше кажучи, нормалізація – це формальний процес визначення того, які поля мають належати до яких таблиць у реляційній базі даних. Нормалізація ґрунтується на наборі правил, розроблених ще на етапі появи реляційних баз даних. Нормалізована реляційна база даних має низку переваг [14]:

- Усунення надлишкового зберігання даних.
- Точне моделювання об'єктів реального світу, процесів та взаємозв'язків між ними.
- Структурування даних, що забезпечує гнучкість моделі.

Нормалізація гарантує, що ви отримаєте всі переваги, які пропонують реляційні бази даних. Час, витрачений на вивчення нормалізації, почне окупатися миттєво.

Існує кілька термінів, що є ключовими для обговорення нормалізації: «ключ» і «залежність». Ці поняття, ймовірно, знайомі кожному, хто створював системи реляційних баз даних, навіть якщо ці люди не вживали саме такі терміни. Ми визначаємо та розглядаємо їх тут як необхідну основу для подальшого обговорення нормальних форм.

2. Первинний ключ

Первинний ключ – це фундаментальне поняття в проектуванні реляційних баз даних. Його суть проста: кожен запис повинен мати елемент, що однозначно його ідентифікує. Первинний ключ може складатися з одного поля або комбінації полів. Первинний ключ таблиці також слугує основою для встановлення зв'язків з іншими таблицями.

Первинний ключ повинен бути унікальним, обов'язковим і незмінним. Класична помилка, якої припускаються під час навчання створенню реляційних баз даних, - використання мінливого поля як первинного ключа [14].

Правильний підхід до створення первинного ключа передбачає використання унікального та незмінного значення.

З поняттям ключа тісно пов’язана особлива концепція нормалізації – функціональна залежність. Друга та третя нормальні форми дозволяють перевірити правильність функціональних залежностей. То що ж таке «функціональна залежність»? Вона описує, як одне поле (або комбінація полів) визначає інше поле [14].

Принципи нормалізації описуються через низку «нормальних форм», вимоги до яких стають дедалі суворішими. Першу нормальну форму (1НФ) найлегше забезпечити, другу (2НФ) — складніше, і так далі. Існує 5 або 6 нормальних форм, залежно від того, чиїх поглядів дотримуватися. Зручно говорити про нормальні форми, використовуючи їхні традиційні назви, оскільки ця термінологія є загальноприйнятою в галузі реляційних баз даних. Утім, до нормалізації можна підійти й без використання цієї термінології [14].

Перша нормальна форма (1НФ)

Першу нормальну форму легко зрозуміти й застосувати: Таблиця перебуває в першій нормальній формі, тоді і лише тоді, коли всі аотрибути атомарні і ключ не має пустих значень [14].

Друга нормальна форма (2НФ)

Друга нормальна форма допомагає виявити випадки, коли дві таблиці було об’єднано в одну. Визначення другої нормальної форми ґрунтується на поняттях первинного ключа та функціональної залежності. Друга нормальна форма визначається так:

Відношення перебуває в другій нормальній формі (2НФ) тоді й тільки тоді, коли воно перебуває в 1НФ і кожен неключовий атрибут повністю залежить від первинного ключа.

Іншими словами, таблиця перебуває у 2НФ, якщо:

1. Вона не містить груп, що повторюються. 2. Кожне поле, що не входить до складу ключа, функціонально залежить від усього ключа.

Якщо використовується ключ, що складається з одного поля, то таблиця, яка вже перебуває в першій нормальній формі (1НФ), автоматично відповідає вимогам другої нормальної форми (2НФ).

Третя нормальна форма (3НФ)

Третя нормальна форма забезпечує додатковий рівень перевірки того, що таблиці не були об’єднані. Таблиця повинна мати поле, яке однозначно ідентифікує кожен її запис, а кожне поле в таблиці має описувати об’єкт, який ця таблиця представляє. Щоб перевірити, чи відповідає таблиця, яка вже перебуває у другій нормальній формі (2NF), також і третій нормальній формі (3NF), слід поставити запитання: «Чи залежать будь-які неключові стовпці від інших неключових стовпців?» [14].

Під час проєктування бази даних досить легко випадково об’єднати інформацію, яка має зберігатися в різних таблицях. У згаданому вище прикладі з поштовими індексами таблиця індексів містила як скорочену назву штату, так і повну назву штату. Повна назва штату визначається скороченою назвою, тому третя нормальна форма вимагає перенести це поле до окремої таблиці.

Вищі нормальні форми

Існують й інші, вищі нормальні форми, зокрема 4NF, 5NF, BCNF та PJ/NF. Ми обмежимося розглядом 3NF, оскільки її достатньо для більшості практичних потреб.

Денормалізація

Денормалізація – це процес модифікації ідеально нормалізованої структури бази даних задля підвищення продуктивності. Денормалізація є природною та необхідною складовою проєктування баз даних, але вона має виконуватися після належної нормалізації [14].

4. Фізичне проектування та оптимізація

Фізичне проектування бази геоданих – це процес трансформації логічної схеми бази даних у конкретні фізичні структури збереження на дискових носіях СУБД (PostgreSQL/PostGIS, Oracle Spatial, SQLite/SpatiaLite) з урахуванням апаратних ресурсів, специфіки файлових систем, внутрішніх алгоритмів індексування та вимог до швидкодії систем [8; 14].

Фізична модель перетворює логічний дизайн на дизайн бази даних, що відповідає конкретній реалізації.

Він включає [23]. :

- Таблиці
- Колонки
- Схеми баз даних
- Індекси
- Обмеження
- Склади для зберігання
- Оптимізація для платформи

Після завершення фізичної моделі команди розробників отримують детальні специфікації, необхідні для створення бази даних на обраній платформі.

Разом ці три моделі забезпечують структурований перехід від бізнес-стратегії до технічної реалізації, зменшуючи ризики протягом усього процесу проектування [23].

5. Проектування баз даних для охорони ґрунтів

Проектування баз даних для охорони ґрунтів є спеціалізованим напрямом розробки геоінформаційних систем (ГІС) та земельно-інформаційних систем (ЗІС), який поєднує методики реляційного та об'єктно-орієнтованого моделювання з вимогами ґрунтознавства, агрохімії, землевпорядкування та екологічного моніторингу [1; 7].

Головна мета створення таких баз геоданих – забезпечення системного збору, збереження, просторового аналізу та оперативного надання інформації про стан ґрунтового покриву для запобігання деградаційним процесам, проведення контурно-меліоративного впорядкування та обґрунтування заходів із відновлення родючості [1; 2].

Опис основних таблиць та їхніх атрибутів:

LandParcels (Земельні ділянки / поля): містить кадастровий номер, площу, цільове призначення, орендаря та векторні межі (MultiPolygon).

SoilPolygons (Контури ґрунтових відмінностей): векторний шар ґрунтової карти з шифром ґрунту, генетичною назвою, гранулометричним складом та класом придатності.

SoilSamples (Точки пробовідбору / моніторингові станції): точкова геометрія (Point), дата та час відбору, висотна відмітка (Z), метод відбору.

LabSoilAnalyses (Результати лабораторних аналізів): таблиця семантичних даних (1:N до проб), що включає вміст органічного вуглецю/гумусу (%), pH водної та сольової витяжок, вміст рухомого фосфору та калію, щільність складення, вміст важких металів та залишків пестицидів.

SoilDegradationEvents (Записи про деградаційні процеси): класифікація еродованості (водна, вітрова), ступень дегуміфікації, деструкції, підтоплення чи замивання.

ProtectionMeasures (Ґрунтоохоронні та меліоративні заходи): перелік запланованих та виконаних заходів (консервація, заліснення, вапнування, створення протиерозійних лісосмуг, диференційоване внесення добрив).

Контрольні питання

1. Що таке проектування баз даних і які його основні етапи?

2. У чому полягає різниця між концептуальним та логічним моделюванням?
3. Які елементи відображаються на ER діаграмі?
4. Що таке нормалізація баз даних і яка її мета?
5. Які проблеми виникають у базі даних без нормалізації?
6. Що означає перша нормальна форма (1НФ) і які вимоги вона встановлює?
7. У чому полягає друга нормальна форма (2НФ) та як вона усуває надлишковість?
8. Які особливості третьої нормальної форми (3НФ) і чому вона важлива?
9. Як нормалізація сприяє ефективності роботи з даними у ґрунтозахисних дослідженнях?
10. Чому фізична модель даних є важливою?

ТЕМА 5. МОВА ЗАПИТІВ SQL

План

1. Основи SQL та його роль у базах даних
2. Основні оператори SQL
3. Запити з умовами та об’єднаннями
4. Просторові запити у SQL
5. SQL і реляційна алгебра

1. Основи SQL та його роль у базах даних

Мова структурованих запитів (SQL — Structured Query Language) є загальноприйнятим міжнародним стандартом (ISO/IEC 9075) декларативного програмування, призначеним для створення, модифікації, управління та вибірки даних у реляційних (РСУБД) та об'єктно-реляційних системах управління базами даних (ОРСУБД) [8; 14].

На відміну від імперативних мов програмування (Python, C++), SQL є **декларативною мовою**: користувач описує *що саме* потрібно отримати у результаті, а *як саме* виконувати запит (вибір індексів, алгоритмів об'єднання таблиць та порядок сканування диска) визначає внутрішній **планувальник запитів (Query Planner / Optimizer)** СУБД [8].

Визначення SQL-схеми

Мову SQL можна вважати однією з головних причин комерційного успіху реляційних баз даних. Оскільки вона стала стандартом для реляційних баз даних, користувачі менше переймалися питаннями перенесення своїх застосунків з інших типів систем (наприклад, мережевих або ієрархічних) на реляційні системи.

Це пояснюється тим, що навіть якщо користувачів переставав задовольняти конкретний продукт — реляційна СКБД, — перехід на іншу реляційну СКБД не вимагав надмірних витрат часу чи коштів, адже обидві системи дотримувалися одних і тих самих мовних стандартів. Звісно, на практиці між різними комерційними пакетами реляційних СКБД існує чимало відмінностей. Проте, якщо користувач сумлінно використовує лише ті можливості, що входять до стандарту, і якщо обидві реляційні системи точно цей стандарт підтримують, то перехід між ними має бути значно простішим. Ще однією перевагою наявності такого стандарту є можливість писати команди в прикладній програмі для роботи з базами даних, які можуть отримувати доступ до даних, що зберігаються у двох або більше реляційних СКБД, без необхідності змінювати підмову роботи з базою даних (SQL), за умови, що обидві СКБД підтримують стандартний SQL. Назва SQL нині розшифровується як Structured Query Language (мова структурованих запитів). Спочатку ця мова мала назву SEQUEL (Structured English Query Language — мова структурованих запитів англійською мовою); її було розроблено та впроваджено в підрозділі IBM Research як інтерфейс для експериментальної системи реляційних баз даних під назвою SYSTEM R. Сьогодні SQL є стандартною мовою для комерційних реляційних СКБД. Спільна робота Американського національного інституту

стандартів (ANSI) та Міжнародної організації зі стандартизації (ISO) призвела до появи стандартної версії SQL (ANSI 1986), відомої як SQL-86 або SQL1. Згодом було розроблено переглянутий і значно розширений стандарт SQL-92 (також відомий як SQL2). Наступним визнаним стандартом став SQL:1999, який на етапі розробки мав назву SQL3. Двома пізнішими оновленнями стандарту стали SQL:2003 та SQL:2006; серед іншого, вони додали до мови підтримку можливостей XML. Ще одне оновлення, випущене у 2008 році, інтегрувало в SQL додаткові функції об'єктно-орієнтованих баз даних.

SQL — це комплексна мова для роботи з базами даних: вона містить команди для визначення даних, виконання запитів та оновлення інформації. Отже, ця мова поєднує в собі функції як DDL, так і DML. Крім того, вона містить засоби для визначення представлень (views) у базі даних, налаштування системи безпеки та авторизації, встановлення обмежень цілісності та керування транзакціями. Також передбачено правила вбудовування інструкцій SQL у мови програмування загального призначення, такі як Java, COBOL або C/C++.

Пізніші стандарти SQL (починаючи з SQL:1999) поділяються на базову специфікацію та спеціалізовані розширення. Базова частина має бути реалізована всіма розробниками реляційних систем керування базами даних (РСКБД), що відповідають стандарту SQL. Розширення можуть реалізовуватися як додаткові модулі, що купуються окремо для вирішення специфічних завдань, таких як інтелектуальний аналіз даних (data mining), робота з просторовими та часовими даними, створення сховищ даних, оперативна аналітична обробка (OLAP), робота з мультимедійними даними тощо.

Визначення даних і типи даних у SQL

У SQL терміни «таблиця», «рядок» і «стовпець» використовуються як відповідники формальних термінів реляційної моделі: «відношення», «кортеж» і «атрибут» відповідно. Ми будемо вживати ці терміни як взаємозамінні. Основною командою SQL для визначення даних є інструкція CREATE; її можна використовувати для створення схем, таблиць (відношень) і доменів (а також інших об'єктів, як-от представлень, тверджень і тригерів). Оскільки специфікація SQL є дуже об’ємною, тут наведено опис лише найважливіших можливостей. Детальнішу інформацію можна знайти у відповідних стандартах SQL.

Типи даних атрибутів і домени в SQL. До основних типів даних, доступних для атрибутів, належать: числові типи, рядки символів, рядки бітів, логічний тип (Boolean), а також типи для зберігання дати й часу.

■ Числові типи даних включають цілі числа різного розміру (INTEGER або INT та SMALLINT) і числа з плавною комою (дійсні числа) різної точності (FLOAT або REAL та DOUBLE PRECISION). Числа з фіксованою точністю можна оголошувати за допомогою конструкції DECIMAL(i,j) — або DEC(i,j) чи NUMERIC(i,j), — де і (точність) — це загальна кількість десяткових цифр, а j (масштаб) — кількість цифр після десяткової крапки.

■ Типи даних для рядків символів можуть бути фіксованої довжини (CHAR або CHARACTER, де n — кількість символів) або змінної довжини (VARCHAR, CHAR VARYING або CHARACTER VARYING, де n — максимальна кількість символів). Літеральне рядкове значення записується в одинарних лапках

(апострофах); при цьому враховується регістр символів (розрізняються великі та малі літери).

■ Типи даних для рядків бітів можуть бути фіксованої довжини n (BIT) або змінної довжини (BIT VARYING, де n — максимальна кількість бітів). Значенням за замовчуванням для n (довжини рядка символів або бітів) є 1. Літеральні рядки бітів записуються в одинарних лапках, але перед ними ставиться літера B, щоб відрізняти їх від рядків символів; ...наприклад, B‘10101’.5 Також доступний інший тип даних для бітових рядків змінної довжини — BINARY LARGE OBJECT (або BLOB); він використовується для визначення стовпців, що містять великі бінарні значення, як-от зображення. Як і у випадку з CLOB, максимальну довжину BLOB можна задавати в кілобітах (K), мегабітах (M) або гігабітах (G). Наприклад, BLOB(30G) визначає максимальну довжину в 30 гігабіт.

■ Тип даних Boolean (логічний тип) має традиційні значення TRUE (істина) або FALSE (хиба). Оскільки в SQL існують значення NULL, тут застосовується тризначна логіка, тому третім можливим значенням для типу Boolean є UNKNOWN (невідомо). Питання щодо необхідності значення UNKNOWN та тризначної логіки розглядаються в розділі 5.

■ Тип даних DATE (дата) займає десять позицій і складається з компонентів YEAR (рік), MONTH (місяць) і DAY (день) у форматі YYYY-MM-DD. Тип даних TIME (час) займає щонайменше вісім позицій і складається з компонентів HOUR (година), MINUTE (хвилина) та SECOND (секунда) у форматі HH:MM:SS.

■ Тип даних TIMESTAMP (часова мітка) включає поля DATE і TIME, а також щонайменше шість позицій для десяткових часток секунди та необов’язковий кваліфікатор WITH TIME ZONE (з урахуванням часового поясу). Літеральні значення записуються як рядки в одинарних лапках із ключовим словом TIMESTAMP перед ними та пробілом між датою й часом; наприклад: TIMESTAMP ‘2008-09-27 09:12:47.648302’.

■ Ще одним типом даних, пов’язаним із DATE, TIME та TIMESTAMP, є тип INTERVAL (інтервал). Він визначає інтервал — відносне значення, яке можна використовувати для збільшення або зменшення абсолютного значення дати, часу чи часової мітки. Інтервали поділяються на інтервали типу YEAR/MONTH (рік/місяць) або DAY/TIME (день/час). Формати DATE, TIME та TIMESTAMP можна розглядати як особливий вид рядків. Тому їх, як правило, можна використовувати в операціях порівняння рядків, перетворюючи (або приводячи) їх до еквівалентних рядкових значень.

Роль SQL у геоінформаційних системах (ГІС) та охороні ґрунтів

У сучасних земельно-інформаційних системах (ЗІС) та просторових базах даних (PostgreSQL/PostGIS, Oracle Spatial, SpatiaLite) SQL виконує роль універсального містка між просторовими даними та аналітичними додатками [29]:

1. Серверна обробка великих масивів геоданих (Server-Side Processing): виконання векторних операцій (перетин, буферизація, розрахунок площ) безпосередньо на сервері баз даних без вивантаження мільйонів об'єктів у оперативну пам'ять клієнтського ГІС-пакета (QGIS, ArcGIS Pro).

2. Забезпечення специфікації OGC Simple Features for SQL: розширення стандартного SQL спеціалізованими просторовими типами даних (GEOMETRY, GEOGRAPHY) та просторовими функціями (префікс ST_ — Spatial and Temporal) [27; 29].

3. Інтеграція семантики та геометрії: можливість у межах єдиного запиту поєднувати лабораторні аналізи ґрунту (вміст гумусу, pH , важкі метали) з просторовим розташуванням ділянок та рельєфом.

4. Стандартизований інтерфейс для Web-ГІС: виступає бекенд-інструментом для сервісів OGC (WMS, WFS, WPS), серверних геопорталів та мобільних додатків польового моніторингу.

2. Основні оператори SQL

Класифікація операторів мови SQL

За функціональним призначенням підмножини команд SQL поділяються на 5 основних категорій [8; 14]:

DDL (Data Definition Language — мова визначення даних): оператори створення, зміни та видалення схем, таблиць, індексів, просторових обмежень та уявлень (CREATE, ALTER, DROP, TRUNCATE).

DML (Data Manipulation Language — мова маніпулювання даними): оператори додавання, оновлення та вилучення атрибутивних і просторових записів (INSERT, UPDATE, DELETE).

DQL (Data Query Language — мова запитів): оператор вибірки даних (SELECT), що становить основу аналітичної роботи в СУБД.

DCL (Data Control Language — мова управління доступом): адміністрування прав доступу користувачів та ролей до об'єктів бази геоданих (GRANT, REVOKE).

TCL (Transaction Control Language — мова управління транзакціями): забезпечення атомарності та цілісності при групових операціях з даними (COMMIT, ROLLBACK, SAVEPOINT).

Функціонування бази геоданих базується на трьох ключових групах операторів SQL: DDL (визначення структури), DML (управління вмістом) та DQL (формування запитів) [8; 14]. У геоінформаційних СУБД ці оператори розширені для роботи з об'єктами OGC Simple Features [27; 29].

Оператори визначення даних (DDL — Data Definition Language)

DDL-оператори відповідають за формування фізичної схеми бази геоданих: створення, модифікацію та видалення таблиць, просторових колон, індексів та обмежень цілісності [8].

- CREATE TABLE: створення нової таблиці з визначенням типів даних (числові, текстові, часові, просторові) та обмежень (Primary Key, Foreign Key, CHECK, NOT NULL).
- ALTER TABLE: модифікація структури існуючої таблиці (додавання нових атрибутів, зміна типів даних, додавання просторових індексів чи обмежень).
- DROP TABLE: повне вилучення таблиці та її даних із баз геоданих.

- TRUNCATE TABLE: миттєве очищення всіх рядків таблиці із збереженням її DDL-структури.

Оператори маніпулювання даними (DML — Data Manipulation Language)

DML-оператори забезпечують внесення, оновлення та вилучення атрибутивних і просторових записів у таблицях [14].

- INSERT INTO: додавання нових рядків. Просторова геометрія передається у текстовому форматі WKT (Well-Known Text) або EWKT за допомогою функцій ST_GeomFromText() або ST_SetSRID() [27; 29].
- UPDATE: коригування значення атрибутів або геометричного положення об'єктів за визначеною умовою.
- DELETE FROM: вилучення записів, що відповідають умові предикату.

Оператор вибірки (DQL — Data Query Language): Структура SELECT

Оператор SELECT є головним інструментом геоаналітики, витягування та трансформації даних [8].

Написання команд SQL

У цьому розділі ми стисло описуємо структуру інструкції SQL та позначення, що використовуються для визначення формату різних конструкцій SQL. Інструкція SQL складається із зарезервованих слів і слів, визначених користувачем. Зарезервовані слова є незмінною частиною мови SQL і мають фіксоване значення. Їх слід писати точно так, як вимагає стандарт; вони не можуть бути розірвані переходом на новий рядок. Слова, визначені користувачем, створюються самим користувачем (відповідно до певних синтаксичних правил) і позначають назви різних об'єктів бази даних, таких як таблиці, стовпці, представлення, індекси тощо. Розміщення слів в інструкції також підпорядковується певним синтаксичним правилам. Хоча стандарт цього не вимагає, багато діалектів SQL передбачають використання спеціального символу для позначення завершення кожної інструкції (зазвичай це крапка з комою «;»).

Більшість компонентів інструкції SQL нечутливі до регістру, тобто літери можна вводити як у верхньому, так і в нижньому регістрі. Важливим винятком із цього правила є символьні літерали (рядкові константи), які слід вводити точно так, як вони представлені в базі даних.

Хоча SQL має вільний формат запису, інструкція або набір інструкцій легше читаються, якщо використовувати відступи та відповідне розташування рядків. Наприклад:

- кожна частина (пункт) інструкції має починатися з нового рядка;
- початок кожної частини має бути вирівняний відносно початку інших частин;
- якщо частина інструкції складається з кількох елементів, кожен з них слід розміщувати в окремому рядку з відступом відносно початку цієї частини, щоб показати взаємозв'язок між ними.

Прості запити

Призначення оператора SELECT — отримання та відображення даних з однієї або кількох таблиць бази даних. Це надзвичайно потужна команда, здатна

в межах одного оператора виконувати дії, еквівалентні операціям реляційної алгебри: вибору (selection), проєкції (projection) та з'єднанню (join).

Вибір рядків (умова WHERE)

Однак часто виникає потреба обмежити кількість рядків, що отримуються в результаті запиту. Це можна зробити за допомогою умови WHERE, яка складається з ключового слова WHERE та умови пошуку, що визначає, які саме рядки слід вибрати. Існує п'ять основних умов пошуку (або предикатів, згідно з термінологією ISO):

- Порівняння — порівняння значення одного виразу зі значенням іншого виразу.
- Діапазон — перевірка того, чи потрапляє значення виразу в заданий діапазон значень.
- Належність до множини — перевірка того, чи дорівнює значення виразу одному зі значень у заданій множині.
- Відповідність шаблону — перевірка того, чи відповідає рядок заданому шаблону.
- Null — перевірка того, чи містить стовпець значення NULL (невідоме значення).

Умова WHERE є еквівалентом операції вибору (selection) у реляційній алгебрі, що розглядалася раніше.

У SQL доступні такі прості оператори порівняння:

= дорівнює

< > не дорівнює (стандарт ISO)

< менше ніж

> більше ніж

! = не дорівнює (допускається в деяких діалектах)

< = менше або дорівнює

> = більше або дорівнює

Використання агрегатних функцій SQL

Окрім отримання рядків і стовпців із бази даних, часто виникає потреба виконати певне підсумовування або агрегування даних — подібно до підрахунку підсумкових значень у нижній частині звіту.

Стандарт ISO визначає п'ять агрегатних функцій:

COUNT — повертає кількість значень у вказаному стовпці;

SUM — повертає суму значень у вказаному стовпці;

AVG — повертає середнє значення у вказаному стовпці;

MIN — повертає найменше значення у вказаному стовпці;

MAX — повертає найбільше значення у вказаному стовпці.

3. Запити з умовами та об'єднаннями

Ефективний аналіз даних у реляційних СУБД базується на здатності прецизійно фільтрувати записи за певною сукупністю умов та комплексувати інформацію з декількох пов'язаних таблиць схеми бази геоданих [8; 14].

Запити до кількох таблиць

Усі розглянуті досі приклади мають суттєве обмеження: стовпці, що мають відображатися в результуючій таблиці, повинні походити з однієї таблиці. У багатьох випадках цього недостатньо. Щоб об'єднати стовпці з кількох таблиць у результуючу таблицю, необхідно використовувати операцію з'єднання (join). Операція з'єднання в SQL об'єднує інформацію з двох таблиць, утворюючи пари пов'язаних рядків із цих таблиць. До результуючої таблиці потрапляють ті пари рядків, у яких відповідні стовпці обох таблиць мають однакові значення.

Якщо потрібно отримати інформацію з більш ніж однієї таблиці, вибір стоїть між використанням підзапиту та операції з'єднання. Якщо результуюча таблиця повинна містити стовпці з різних таблиць, необхідно використовувати з'єднання. Для виконання з'єднання достатньо вказати кілька імен таблиць у реченні FROM, розділивши їх комами, і, як правило, додати речення WHERE для визначення стовпця (або стовпців), за якими здійснюється з'єднання. Також можна використовувати псевдонім (аліас) для таблиці, зазначеної в реченні FROM. У такому разі псевдонім відокремлюється від імені таблиці пробілом. Псевдонім можна використовувати для уточнення імені стовпця, якщо виникає неоднозначність щодо джерела цього стовпця. Його також можна використовувати як скорочене позначення імені таблиці. Якщо псевдонім визначено, його можна використовувати замість імені таблиці в будь-якому місці запиту.

Концептуально процедура формування результату оператора SELECT із використанням з'єднання виглядає так:

1) Утворюється декартів добуток таблиць, зазначених у реченні FROM.

2) Якщо присутнє речення WHERE, до кожного рядка отриманої таблиці застосовується умова пошуку; зберігаються лише ті рядки, що задовольняють цю умову. З погляду реляційної алгебри ця операція являє собою обмеження декартового добутку.

3) Для кожного з отриманих рядків визначаються значення всіх елементів зі списку SELECT, що формує один рядок результуючої таблиці.

4) Якщо вказано SELECT DISTINCT, із результуючої таблиці видаляються всі дубльовані рядки. У реляційній алгебрі кроки 3 і 4 еквівалентні операції проекції результату обмеження на стовпці, зазначені у списку SELECT.

5) Якщо присутній оператор ORDER BY, слід відсортувати результуючу таблицю відповідним чином.

Об'єднання результуючих таблиць (UNION, INTERSECT, EXCEPT)

У SQL для об'єднання результатів двох або більше запитів в одну результуючу таблицю можна використовувати стандартні теоретико-множинні операції: об'єднання (Union), перетин (Intersection) та різницю (Difference):

- Об'єднання (Union) двох таблиць, A та B, — це таблиця, що містить усі рядки, які є в першій таблиці A, у другій таблиці B або в обох таблицях.
- Перетин (Intersection) двох таблиць, A та B, — це таблиця, що містить усі рядки, спільні для обох таблиць A та B.
- Різниця (Difference) двох таблиць, A та B, — це таблиця, що містить усі рядки, які є в таблиці A, але відсутні в таблиці B.

Існують обмеження щодо таблиць, які можна об'єднувати за допомогою цих операцій; найважливіше з них полягає в тому, що таблиці повинні бути сумісними для об'єднання (union-compatible), тобто мати однакову структуру. Це означає, що таблиці повинні містити однакову кількість стовпців, а відповідні стовпці — мати однакові типи даних і розміри. Користувач самостійно відповідає за те, щоб значення даних у відповідних стовпцях належали до одного й того самого домену.

ПРИКЛАД

Вибірка ділянок із вмістом гумусу від 3.0 до 5.0% та високою кислотністю (pH < 5.5)

```
SELECT polygon_id, soil_type, humus_pct, ph_water
FROM soil_chemistry_samples
WHERE (humus_pct BETWEEN 3.0 AND 5.0)
  AND ph_water < 5.5
  AND land_use_type IN ('Рілля', 'Пасовище')
  AND inspector_notes IS NOT NULL;
```

4. Просторові запити у SQL

Просторові запити є розширенням реляційної мови SQL, що дозволяє виконувати вибірку, обчислення та аналіз даних на основі їхнього географічного положення, форми та топологічних взаємозв'язків [27; 29].

Просторовий SQL (Spatial SQL) – це сумісна мова для роботи з просторовими даними, що дозволяє використовувати геопросторовий аналіз, просторову науку про дані та розробку додатків [20].

Вона надає користувачам ГІС та геопросторових даних можливість працювати в тому ж місці, що й інші дані в базах даних та сховищах даних, усуваючи традиційну ізоляцію між ГІС та іншими сферами організації.

Просторовий SQL підтримує розширені операції, такі як просторове моделювання та машинне навчання, в тому ж місці, де знаходяться дані [20].

Spatial SQL - це не просто мова запитів. Вона переростає ці межі, а сфера її застосування виходить далеко за межі простого зберігання просторових даних у базах даних. Вона забезпечує швидший просторовий аналіз, підтримує роботу просторових застосунків, уможливлює аналіз просторових даних (spatial data science) і машинне навчання, а також багато іншого [20].

Spatial SQL — це універсальна мова для роботи з просторовими даними, що забезпечує можливості геопросторового аналізу, аналізу просторових даних і розробки застосунків.

Вона дозволяє фахівцям у сфері ГІС та геопросторових технологій працювати з даними безпосередньо там, де вони зберігаються (у базах даних і сховищах даних), усуваючи традиційні бар'єри між ГІС та іншими підрозділами організації.

Spatial SQL підтримує виконання складних операцій, таких як просторове моделювання та машинне навчання, безпосередньо в середовищі зберігання даних [20].

Spatial SQL використовує ті самі елементи й структуру, що й стандартна мова SQL, але дозволяє працювати з додатковими типами даних: GEOMETRY (геометрія) або GEOGRAPHY (географія).

Як змінюється Spatial SQL?

Spatial SQL еволюціонує від компонента мови запитів до бази даних у повноцінний інтегрований елемент сучасного стеку ГІС та геопросторових технологій. Щоб зрозуміти суть цих змін, розгляньмо широкий спектр сценаріїв використання Spatial SQL та ролі фахівців, які їх застосовують [20]:

Адміністратор бази даних може використовувати Spatial SQL для зберігання, керування, оновлення та надання кінцевим користувачам доступу до геопросторових даних у різних екземплярах баз даних (це найбільш традиційний сценарій використання Spatial SQL). Такий фахівець може створювати в базі даних збережені процедури та тригери для оновлення даних у міру їх надходження, а також автоматизувати такі процеси, як зміна проєкцій, маніпуляції з даними, форматування тощо. База даних також може бути інтегрована з іншими інструментами, що надають користувачам можливість переглядати дані, виконувати запити та отримувати до них доступ. Зазвичай робота з базою даних здійснюється через такі інструменти, як pgAdmin або psql (командний рядок) у PostGIS, або ж через графічні інтерфейси чи інструменти командного рядка для хмарних сховищ даних [20].

ГІС-аналітик може виступати кінцевим користувачем даних, що зберігаються в базі даних із підтримкою Spatial SQL, використовуючи їх для виконання запитів, аналізу та візуалізації. Аналітики використовують набори даних (окремо або в поєднанні) для отримання цінної інформації, застосовуючи функції просторових відношень у Spatial SQL та інші аналітичні інструменти. Як правило, вони виконують запити до даних, не вносячи в них змін чи оновлень (за винятком випадків, коли створюють власні копії даних). Доступ до даних та їх візуалізація здійснюються за допомогою Python, Jupyter Notebooks, QGIS, CARTO та інших інструментів. Використання PostGIS та QGIS [20].

Фахівець із просторових даних (spatial data scientist) застосовує просторовий SQL для вирішення різноманітних завдань. Це включає отримання даних із баз даних для подальшого аналізу в Python шляхом їх об’єднання з іншими просторовими або непросторовими наборами даних. Також можуть використовуватися складніші концепції SQL для ранжування, застосування віконних функцій, просторової кластеризації, агрегації та аналізу просторових взаємозв’язків — операцій, які ефективніше виконувати безпосередньо в SQL. Крім того, просторовий SQL застосовується для аналізу просторових зв’язків і збагачення даних, що слугують ознаками (features) для моделей машинного навчання (процес, відомий як просторова інженерія ознак). Зазвичай доступ до цих даних здійснюється через середовища на кшталт Jupyter Notebooks, а в роботі використовуються як бази даних (наприклад, PostGIS), так і сховища даних (наприклад, BigQuery) [20].

Інженер із геопросторових даних (geospatial data engineer) може отримувати доступ до просторового SQL та використовувати його різними способами. Наприклад, він може застосовувати SQL у межах GDAL для

виконання операцій із перетворення та керування даними (зокрема, з використанням SpatiaLite). Також можливе використання бази даних разом із сервісами для роботи з тайлами або просторовими об'єктами (feature services), такими як pg_featureserv чи pg_tileserv. SQL використовується для перетворення та об'єднання різних наборів даних із метою створення нової інформації, очищення й перевірки даних, видобування та трансформації значень із різних стовпців, а також для підготовки наборів даних для бізнес-користувачів. Для керування великими обсягами даних та їх завантаження застосовуються сховища на кшталт BigQuery та інструменти на зразок Dataflow з бібліотекою geobeam. У роботі використовуються різноманітні засоби: від командного рядка та настільних програм (як-от QGIS) до інтерфейсів баз даних, скриптів на Python тощо.

Звісно, просторова SQL (spatial SQL) як мова - це , по суті, звичайна SQL, однак вона дозволяє реалізувати безліч додаткових сценаріїв використання; саме ці можливості (і не тільки вони) роблять просторову SQL такою унікальною та потужною.

Як у базах даних, так і у сховищах даних просторову SQL можна використовувати для [20]:

– Запитів до просторових наборів даних та управління ними

– Управління системами координат (проєкціями) та перепроєціювання даних

– Зберігання просторових даних у різних форматах (WKT, GeoJSON тощо)

– Аналізу та розуміння різноманітних типів просторових взаємозв'язків

– Виконання просторової кластеризації та аналізу найближчих сусідів

– Перетворення геометрій за допомогою буферних зон, діаграм Вороного та опуклих чи увігнутих оболонок (hulls)

– Обчислення відстаней — як за прямою лінією, так і з урахуванням кривизни Землі

– Роботи з 3D-даними (запити та управління)

– Створення тригерів для зміни й оновлення даних на основі певних подій, наприклад, операції INSERT у конкретну таблицю

– Підключення до API та інших зовнішніх сервісів

– Створення користувацьких функцій іншими мовами програмування, такими як Python, JavaScript та SQL

– Виконання статистичного аналізу та завдань машинного навчання за допомогою спеціальних функцій та інструментів (наприклад, BigQuery ML)

– Зберігання неструктурованих даних (наприклад, у форматі JSON) та управління ними

– Виведення даних у різних форматах, зокрема GeoJSON або Shapefile

– Створення та генерування тайлів карти для фронтенд-застосунків

5. SQL і реляційна алгебра

Мова запитів SQL базується на математичному фундаменті реляційної алгебри, сформульованої Едгаром Коддом у 1970 році [11; 8]. Розуміння цього

зв'язку є критично важливим для оптимізації складних аналітичних і просторових запитів у геоінформаційних СУБД.

Основна відмінність між реляційною алгеброю та мовою SQL полягає у парадигмі опису операцій [8; 14]:

Реляційна алгебра є процедурною мовою: вона визначає послідовність алгебраїчних операцій, які потрібно виконати над відношеннями (таблицями), щоб отримати підсумкове відношення.

SQL є декларативною мовою: розробник описує у запиті який саме результат потрібен (що отримати), а СУБД самостійно трансформує декларативний запит у еквівалентне алгебраїчне дерево операцій (як обчислити).

Властивість замкненості (Relational Closure) гарантує, що аргументами будь-якої операції реляційної алгебри є відношення (таблиці), і результатом її виконання також завжди є відношення [8].

Реляційна алгебра визначає, як отримати потрібну інформацію та як побудувати відношення на основі одного або кількох інших відношень. Реляційна алгебра, що походить від логіки першого порядку (та алгебри множин), оперує множиною скінченних відношень, замкненою відносно певних операторів. Ці оператори діють на одне або кілька відношень, результатом чого є нове відношення. Реляційна алгебра є частиною інформатики [14].

З реляційною алгеброю пов’язане поняття замкненості. Замкненість означає, що реляційні вирази можна вкладати один в одного. Наприклад, операцію, результатом якої є нове відношення, можна взяти в дужки та використати в іншому виразі. Відношення є замкненими відносно операцій реляційної алгебри.

Визначення реляційної алгебри

Домен: множина відношень

Базові оператори: вибір (select), проекція (project), об'єднання (union), різниця множин (set difference), декартів (перехресний) добуток (Cartesian product) [14].

Похідні оператори: перетин множин (intersection), ділення (division), з'єднання (join)

Процедурність: реляційний вираз визначає запит шляхом опису алгоритму (послідовності застосування операторів) для отримання результату виразу

Е. Ф. Кодд (1972) запропонував 8 операцій, що використовуються в реляційній алгебрі. Нижче наведено 5 фундаментальних операторів: вибір; проекція; декартів добуток; об'єднання; різниця множин [14].

Інші 3 оператори можуть бути визначені через ці фундаментальні оператори; вони наведені нижче: з'єднання; перетин; ділення.

Унарні оператори над відношеннями: вибір і проекція. Операції вибору та проекції є унарними, оскільки вони виконуються над одним відношенням. Інші операції працюють із парами відношень і тому називаються бінарними. У наведених нижче визначеннях нехай R і S — два відношення, визначені відповідно на наборах атрибутів A = (a1, a2, ..., aN) та B = (b1, b2, ..., bM).

Оператори: **<, ≤, ≥, >, =, ≠**

Проста умова вибору:

– <атрибут> оператор <константа>
– <атрибут> оператор <атрибут>
– <умова> AND <умова>
– <умова> OR <умова>
– NOT <умова>

Вибір (або обмеження) σ предикат(R). Операція вибору виконується над одним відношенням R і визначає відношення, що містить лише ті кортежі з R, які задовольняють задану умову (предикат) [14].

Проекція π a1, ..., an(R). Операція проекції виконується над одним відношенням R і визначає відношення, що містить вертикальну підмножину R, виділяючи значення вказаних атрибутів і усуваючи дублікати.

Бінарні реляційні операції [14].

Об'єднання R ∪ S. Об'єднання двох відношень R і S визначає відношення, що містить усі кортежі з R, або з S, або з обох відношень (R і S), причому дубльовані кортежі усуваються. R і S повинні бути сумісними для об'єднання.

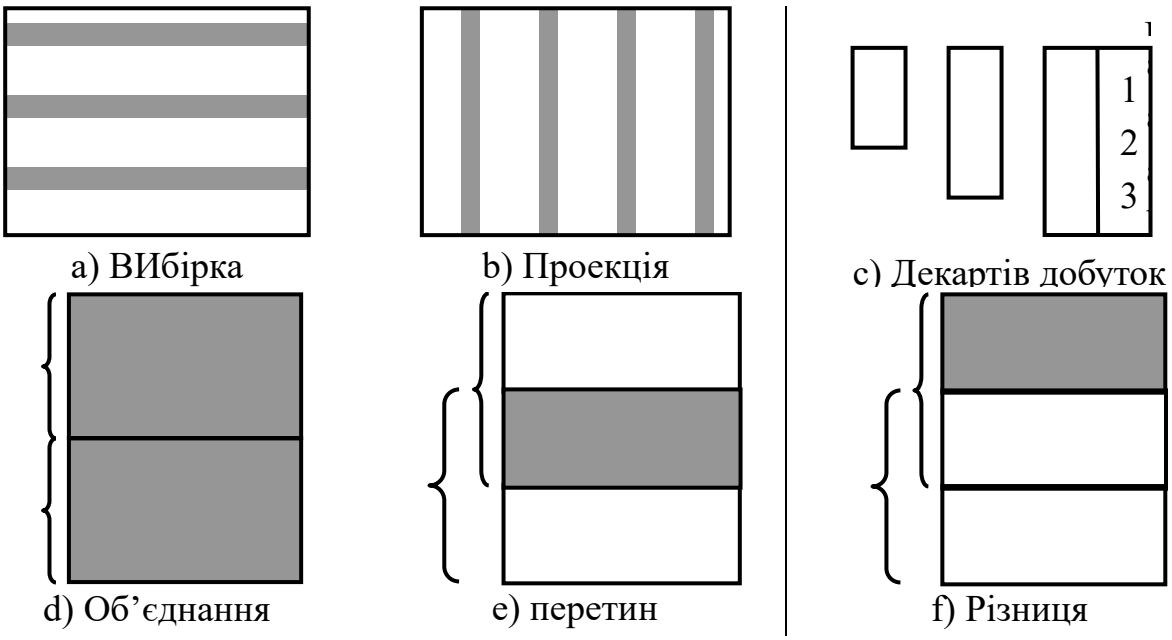
Якщо R і S мають відповідно I та J кортежів, їхнє об'єднання отримують шляхом злиття в одне відношення, що містить щонайбільше (I + J) кортежів. Об'єднання можливе лише тоді, коли схеми двох відношень збігаються, тобто вони мають однакову кількість атрибутів, а кожна пара відповідних атрибутів має однакову доменну область. Іншими словами, відношення повинні бути сумісними для об'єднання. Зауважте, що назви атрибутів не враховуються при визначенні сумісності для об'єднання. У деяких випадках для забезпечення сумісності двох відношень можна використовувати операцію проекції [14].

Різниця множин. R − S. Операція різниці множин визначає відношення, що складається з кортежів, які є у відношенні R, але відсутні в S. R і S повинні бути сумісними для об'єднання [14].

Перетин. R ∩ S. Операція перетину визначає відношення, що складається з усіх кортежів, які є як у R, так і в S. R і S повинні бути сумісними для об'єднання. Декартів добуток. R × S. Операція декартового добутку визначає відношення, що є результатом з’єднання кожного кортежу відношення R з кожним кортежем відношення S [14].

Декартів добуток (R × S). Операція декартового добутку визначає відношення, що утворюється шляхом з’єднання кожного кортежу відношення R з кожним кортежем відношення S [14].

Операція декартового добутку перемножує два відношення, утворюючи нове відношення, яке складається з усіх можливих пар кортежів із цих двох відношень. Отже, якщо одне відношення має I кортежів і N атрибутів, а інше — J кортежів і M атрибутів, то результуюче відношення міститиме (I × J) кортежів із (N + M) атрибутами. Можлива ситуація, коли обидва відношення мають атрибути з однаковими назвами. У такому разі до назв атрибутів додається префікс із назвою відношення, щоб забезпечити унікальність назв атрибутів у межах відношення [14].



a) ВИбірка b) Проекція c) Декартів добуток

d) Об’єднання e) перетин f) Різниця

Ілюстрація, що демонструє дію операцій реляційної алгебри.

Контрольні питання

1. Що є SQL і для чого він використовується?
2. Які основні оператори SQL ви знаєте?
3. Для чого використовується оператор SELECT?
4. Яка різниця між INSERT, UPDATE та DELETE?
5. Як працює оператор WHERE у SQL?
6. Що таке JOIN і які його види існують?
7. Які агрегатні функції SQL застосовуються для аналізу даних?
8. Які операції реляційної алгебри реалізуються у SQL та як вони співвідносяться з його командами?

ТЕМА 6. ПРОСТОРОВІ ОПЕРАЦІЇ ТА АНАЛІЗ ДАНИХ У ГІС

План

1. Основи просторових операцій у ГІС
2. Основні типи просторових операцій
3. Просторовий аналіз та моделювання
4. Практичне застосування просторових операцій у ґрунтозахисних дослідженнях

1. Основи просторових операцій у ГІС

Просторова операція у ГІС – це процедура обробки геопросторових даних, яка використовує геометричні та топологічні властивості об'єктів (координати, форму, взаємне розташування) для створення нових даних, обчислення метрик або виявлення просторових закономірностей [12].

Теоретичною основою просторових операцій є Перший закон географії Волдо Тоблера (*Tobler's First Law of Geography*): *"Усе пов'язано з усім іншим, але близькі речі пов'язані більше, ніж далекі"* [12].

За рівнем просторової локалізації даних, що беруть участь у розрахунках (насамперед для растрових моделей геоданих), операції поділяють на чотири фундаментальні класи [7, 12]:

1. Локальні операції (Local operations): значення нової комірки чи об'єкта є функцією значень лише відповідної комірки в одному або кількох вхідних шарах. *Приклад:* арифметична комбінація растрових шарів вмісту N , P , K для оцінки комплексної забезпеченості елементами живлення.
2. Фокальні операції (Focal operations): значення у вихідній комірці обчислюється на основі значень у її просторовому околі. *Приклад:* розрахунок кутів нахилу та експозиції схилів за цифровою моделлю рельєфу (ЦМР/DEM), фільтрація матриць висот [32].
3. Зональні операції (Zonal operations): обчислення агрегованих статистичних показників (середнє, сума, мінімум/максимум) для комірок аналітичного растру, які потрапляють у межі дискретних зон другого (зонуючого) векторного чи растрового шару. *Приклад:* розрахунок середнього ухилу або середнього вегетаційного індексу NDVI у межах кожного окремого поля чи робочої ділянки.
4. Глобальні операції (Global operations): значення у кожній комірці залежить від усіх комірок вхідного растрового покриття. *Приклад:* побудова карт евклідових відстаней до гідрографічної мережі, розрахунок кумулятивного стоку або шляхів найменшого опору (Cost Path).

Відмінності просторових операцій у векторній та растровій моделях даних

Параметр / Модель	Векторна модель (Vector)	Растрова модель (Raster)
Базовий елемент	Точка, лінія, полігон (координати X, Y)	Комірка / Піксель (Grid cell)
Точність геометрії	Висока (визначається точністю знімання чи векторизації)	Обмежена розміром комірки (просторовою розроздільністю)
Топологічний аналіз	Потребує збереження або обчислення топологічних зв'язків	Неявна топологія (за рахунок суміжності пікселів у матриці)
Переважні операції	Точний векторний оверлей, мережевий аналіз, буферизація	Картографічна алгебра, аналіз поверхонь, моделювання безперервних полів

2. Основні типи просторових операцій

Основні типи просторових операцій у ГІС класифікують за метою перетворення геометрії та атрибутів об'єктів. До базових належать оверлейні операції, аналіз близькості (буферизація, зонування), просторова агрегація та генералізація [12, 29].

Оверлейні операції (Overlay Operations)

Оверлей – це операція накладання двох або більше просторових шарів, в результаті якої створюється новий геометричний шар, а підсумкова таблиця атрибутів поєднує атрибутивні характеристики всіх вхідних шарів [12].

Базові класичні алгоритми векторного оверлею:

Операція	Схема геометричного перетворення	Логічний зміст та атрибутивне поєднання
Перетин (Intersect)	Зберігаються лише ті геометрії, що є спільними для обох шарів	Зберігаються атрибути обох шарів для перетнутих ділянок.
Об'єднання (Union)	Об'єднує всі геометрії обох шарів із розбиттям у точках перетину меж.	Зберігаються всі атрибути. Для неперетнутих областей атрибути іншого шару отримують NULL.
Ідентичність (Identity)	Перетинає вхідний шар A з оверлейним шаром B . Межі A зберігаються повністю.	Зберігаються всі атрибути A ; атрибути B додаються лише туди, де відбувся перетин.
Вирізання (Clip)	Геометрія A обрізається за зовнішньою межею полігону B (як "печиво формочкою").	Атрибути B не додаються — зберігається лише структура атрибутів A .
Стирання (Erase / Difference)	З шару A видаляються всі території, що перетинаються з шаром B (A \setminus B).	Зберігаються лише атрибути шару A .

*Операції аналізу близькості та позиціонування (Proximity Analysis***)**

Аналіз близькості визначає просторові взаємозв'язки між об'єктами на основі вимірювання відстаней між ними [12].

А. Буферизація (Buffering)

Формування просторової зони заданої ширини r навколо точкових, лінійних або полігональних об'єктів. Типи буферів: фіксованого радіуса, змінного радіуса (значення ширини береться з атрибутивного поля, наприклад, залежно від ширини річки або санітарної зони), багатокільцеві (Multi-ring buffer).

Б. Полігони Тіссена / Вороного (Voronoi / Thiessen Polygons)

Розбиття території навколо дискретної множини точок на полігони за принципом найближчого сусідства. Будь-яка точка всередині полігону знаходиться ближче до вихідної точки, ніж до будь-якої іншої точки з множини *Застосування:* інтерполяція зон впливу метеостанцій, свердловин чи пунктів спостереження за ґрунтовим покривом [32].

Операції агрегації та генералізації (Dissolve & Generalization)

1. Розчинення меж (Dissolve): операція просторової агрегації, під час якої видаляються спільні межі між суміжними полігонами, якщо вони мають однакові значення визначеного атрибута (наприклад, об'єднання ділянок з однаковим типом ґрунту або землекористуванням). Під час розчинення часто виконується атрибутивна агрегація (обчислення SUM, AVG, COUNT) [32].
2. Спрощення геометрії (Simplify / Generalization): зменшення кількості вершин у лінійних та полігональних об'єктах із збереженням їхньої загальної топологічної форми. Найпоширенішим є алгоритм Дугласа-Пекера (Douglas-Peucker Algorithm), який відкидає вершини, що відхиляються від прямолінійного сегмента на величину, меншу за заданий поріг толерантності [32] .

3. Просторовий аналіз та моделювання

Просторовий аналіз передбачає застосування математико-статистичних та геометричних методів для виявлення прихованих просторових структур, взаємозв'язків, трендів і аномалій у геопросторових даних [12].

Просторова інтерполяція (Spatial Interpolation) [32]

Просторова інтерполяція – це процес оцінки значень безперервного поля у необстежених точках на основі дискретної множини виміряних опорних точок x_i [12].

Основні класи методів просторової інтерполяції**:**

Метод інтерполяції	Математична сутність та формула	Особливості та обмеження застосування
Обернені зважені відстані (IDW)	$Z(x_0) = \frac{\sum_{i=1}^{n} \frac{Z(x_i)}{d_i^p}}{\sum_{i=1}^{n} \frac{1}{d_i^p}}$	Детерміністичний метод. Значення залежить від відстані d_i та степеневого коефіцієнта p . Вразливий до нерівномірності точки вибірки; утворює ізольовані "плями" ("bull's eye effect").

Звичайний Крігінг (Ordinary Kriging)	$Z(x_0) = \sum_{i=1}^{n} \lambda_i Z(x_i), \quad \sum \lambda_i = 1$	Геостатистичний метод (Оптимальний Незміщений Лінійний Оцінювач — BLUE). Ваги \lambda_i обчислюються на основі експериментальної та теоретичної варіограми \gamma(h) . Обчислює помилку інтерполяції (кригінгову дисперсію).
Сплайнова інтерполяція (Spline)	Радіальні базисні функції (RBF)	Гладка математична поверхня з мінімізацією загальної кривизни. Ефективна для неперервно мінливих полів (наприклад, температур, ухилів), але може давати викиди на екстремумах.
Природні сусіді (Natural Neighbor)	Зважування за площею діаграм Вороного / Тіссена	Автоматично адаптується до місцевої щільності точок. Не екстраполює значення за межі опуклої оболонки (Convex Hull) вибірних точок.

Аналіз поверхонь та морфометричний аналіз (DEM / ЦМР)

Цифрові моделі рельєфу (ЦМР / DEM) є основою для розрахунку морфометричних характеристик поверхні, що визначають перерозподіл речовини, вологи та енергії в ландшафтах [12].

Ключові морфометричні параметри:

1. Кут нахилу.
2. Експозиція - визначає інсоляційний та термічний режим схилу.

- Кривизна поверхні: *Профільна кривизна* - уздовж лінії схилу; визначає прискорення або уповільнення поверхневого стоку (зони акумуляції чи змиву), *Планова кривизна* - перпендикулярно до лінії схилу; визначає конвергенцію (сходження) або дивергенцію (розходження) стоку.

Просторова статистика та просторова автокореляція

Просторова статистика враховує географічне розташування об'єктів для оцінки рівномірності, кластеризації чи випадковості їхнього розподілу [12; 21].

А. Оцінка щільності ядра (Kernel Density Estimation, KDE) [32]

Б. Глобальний та локальний індекси Морана (Moran's I) [32]

B. Статистика Ґетіса-Орда (Hot Spot Analysis)

Багатокритеріальне просторове моделювання (MCE / AHP)

Просторове моделювання передбачає поєднання декількох просторових факторів (тематичних шарів) для розрахунку комплексних індексів або оптимізаційних рішень [12].

4. Практичне застосування просторових операцій у ґрунтозахисних дослідженнях

Використання ГІС і баз даних у ґрунтозахисних дослідженнях та контурно-меліоративній організації території (КМОТ) базується на комплексуванні оверлейних операцій, аналізу поверхонь за ЦМР та геостатистичного моделювання [1]

Контурно-меліоративна система землеробства передбачає адаптацію меж сівозмін та технологічних операцій до топографічних ізогіпс та ступеня еродованості ґрунтів [1].

Основні прикладні задачі КМОТ у ГІС:

- ✓ Проектування прибережних захисних смуг (ПЗС) та водоохоронних зон: застосування інструменту зі змінним радіусом залежно від крутизни схилу та категорії водотоку.
- ✓ Трансформація малопродуктивних та еродованих земель: використання оверлейних операцій для виділення ріллі з кутами нахилу понад або із втратою вмісту гумусу понад 30% для їх подальшого вилучення з інтенсивного обробітку (консервація, залуження, заліснення).
- ✓ Оптимізація контурів полів: розчинення меж робочих ділянок із близькими ерозійними характеристиками та оптимізація напрямку обробітку ґрунту паралельно до горизонталей рельєфу.

Просторова статистика в оцінці ризиків вітрової та водної деградації. Для аналізу просторової неоднорідності деградаційних процесів застосовуються інструменти просторової статистики:

Ядерна оцінка щільності – картографування щільності яружної мережі, промоїн або лінійних ерозійних порушень для визначення осередків максимальної деструкції ландшафту [32].

Аналіз гарячих плям – локалізація кластерів із високими показниками втрат ґрунту або дефляційної небезпеки (вітрової ерозії) на відкритих водозборах.

Просторове оцінювання епізоотичних та техногенних ризиків: оцінка просторового суміщення еродованих територій із потенційно небезпечними об'єктами (худобомогильниками, зонами поствоєнного забруднення) за допомогою просторових предикатів та аналізу просторової автокореляції.

ГІС-операції поділяють на кілька ключових категорій, кожна з яких вирішує конкретні завдання у ґрунтозахисній сфері:

1. Оверлейний аналіз (Накладання шарів) [32]. Об'єднання або перетин двох і більше тематичних карт (наприклад, карти крутизни схилів та карти типів ґрунтів). Застосовується для виділення ділянок з найвищим ризиком деградації

(наприклад, поєднання крутого схилу, легкозмивного ґрунту та відсутності рослинного покриву).

2. Аналіз поверхонь (Цифрові моделі рельєфу – ЦМР) [32]. Розрахунок похідних показників рельєфу на основі матриці висот. Застосовується для визначення крутизни та експозиції схилів: визначення швидкості стоку води та ступеня сонячної інсоляції, що впливає на висушування ґрунту; напрямку стоку (Flow Direction) та акумуляція стоку (Flow Accumulation): моделювання шляхів руху та концентрації талих і дощових вод, що дозволяє прогнозувати утворення промоїн та ярів.

3. Буферизація (Створення буферних зон) [32]. Генерація полігонів на заданій відстані навколо географічних об'єктів (точок, ліній чи інших полігонів). Застосовується для проектування прибережних захисних смуг навколо водойм для запобігання замуленню та змиву добрив з полів; визначення оптимальної ширини та розміщення полезахисних лісосмуг для зменшення вітрової ерозії (дефляції).

4. Зональна статистика (Zonal Statistics) [32]. Обчислення статистичних показників (середнє, максимум, сума) растрового шару в межах певних полігонів. Застосовується для розрахунку середнього обсягу втрат ґрунту (т/га) в межах конкретного поля, адміністративного району чи водозбірного басейну; оцінки динаміки індексу вегетації (NDVI) на різних типах ґрунтів для аналізу їхньої родючості.

Контрольні питання

1. Що таке просторові операції у ГІС і яку роль вони відіграють?
2. Які основні типи просторових операцій ви знаєте?
3. У чому полягає сутність операції буфер?
4. Як працює операція перетину (Intersection)?
5. Для чого використовується операція об’єднання (Union)?
6. Що таке операція вирізання (Clip) і які її практичні застосування?
7. Які методи просторового аналізу використовуються у ГІС?
8. Як просторовий аналіз допомагає моделювати процеси деградації ґрунтів?
9. Які приклади застосування просторових операцій можна навести у сфері охорони ґрунтів?
10. Чому просторові операції вважаються ключовим інструментом для прийняття рішень у ГІС

ТЕМА 7. ЗАСТОСУВАННЯ ГІС ТА БАЗ ДАНИХ У СФЕРІ ОХОРОНИ ҐРУНТІВ

План

1. Роль ГІС у моніторингу ґрунтів
2. Використання баз даних у ґрунтозахисних дослідженнях
3. Інтеграція ГІС та баз даних
4. Практичні приклади застосування
5. Перспективи розвитку та міжнародний контекст

1. Роль ГІС у моніторингу ґрунтів

Моніторинг ґрунтів – це система систематичних спостережень за станом ґрунтового покриву, оцінювання його параметрів, виявлення та прогнозування динаміки деградаційних процесів під впливом природних та антропогенних факторів [Burrough & McDonnell, 1998; 19].

Геоінформаційні системи (ГІС) виступають технологічним ядром сучасного моніторингу, забезпечуючи інтеграцію різнорідних даних (наземних, дистанційних, лабораторних) у єдиний просторово-часовий простір [Minasny & McBratney, 2016; 7].

Ключові функції ГІС у ґрунтовому моніторингу

1. Багаточасовий просторовий аналіз (Multi-temporal Analysis): відстеження просторової динаміки процесів деградації (дегуміфікації, деструкції структури, підтоплення, вторинного засолення) за часовими серіями спостережень [12, 19].

2. Інтеграція дистанційного зондування Землі (ДЗЗ): детектування відкритої поверхні ґрунту та динаміки рослинного покриву за допомогою оптичних, мультиспектральних та радіолокаційних супутникових систем (Sentinel-1/2, Landsat-8/9, PlanetScope) та знімання з БПЛА [29].

3. Моделювання та прогнозування: автоматизований розрахунок потенційних втрат ґрунту від водної та вітрової ерозії, виявлення осередків деструкції на основі інтеграції ЦМР та просторово-статистичних моделей [14, 26].

Спеціалізовані спектральні індекси відкритого ґрунту та деградації

При використанні даних ДЗЗ у ГІС застосовуються комбінації спектральних каналів для прямого або непрямого картографування стану ґрунтів:

Індекс відкритого ґрунту (Bare Soil Index — BSI):

$$BSI = \frac{(SWIR_1 + RED) - (NIR + BLUE)}{(SWIR_1 + RED) + (NIR + BLUE)}$$

Призначення: ідентифікація відкритих ґрунтових поверхонь та відділення їх від рослинності чи штучних покриттів.

Індекс засолення (Salinity Index — SI):

$$SI = \sqrt{BLUE \cdot RED}$$

Призначення: детектування виходів солей на поверхню ґрунту в аридних та напіваридних зонах або на зрошуваних землях.

Нормалізований диференційний вегетаційний індекс (NDVI):

$$NDVI = \frac{NIR - RED}{NIR + RED}$$

Призначення: непряма оцінка протиерозійної стійкості поверхні через коефіцієнт рослинного покриву [8].

Характеристика джерел даних для ГІС-моніторингу ґрунтів

Джерело даних	Просторове розрізнення	Періодичність	Застосування в моніторингу
Наземне агрохімічне обстеження	Точкова (пробовідбір з GPS-прив'язкою)	1 раз на 3–5 років	Визначення гумусу, NPK, pH, важких металів, щільності складення.
Супутникові дані (Sentinel-2, Landsat-8/9)	10–30 м	2–5 днів	Розрахунок вегетаційних індексів, індексу відкритого ґрунту, динаміки режиму вологості.
Дані з БПЛА	1–10 см	На запит	Детальна детектування промоїн, оцінка структури посівів, мікрорельєфу та дефляційних осередків.
Дані супутникової радарії (Sentinel-1 SAR)	10 м	6–12 днів	Оцінка вологості верхнього шару ґрунту та шорсткості поверхні незалежно від хмарності.

2. Використання баз даних у ґрунтозахисних дослідженнях

Сучасні ґрунтозахисні дослідження вимагають опрацювання великих обсягів різнорідної інформації: агрохімічних показників лабораторних аналізів, морфологічних описів ґрунтових розрізів, параметрів рельєфу, матеріалів ДЗЗ та нормативно-правових обмежень [12].

Ефективне управління цими масивами забезпечується за рахунок створення Реляційних СУБД із просторовими розширеннями (Spatial DBMS), які формують основу Ґрунтових Інформаційних Систем (ҐІС / Soil Information Systems — SIS) [5, 12, 26].

Концептуальна та логічна структура бази геопросторових даних ґрунтового покриву

Ґрунтові дані мають чітку ієрархічну та просторово-зв'язану структуру. Для забезпечення 3-ї нормальної форми (3NF) та підтримки просторових відносин застосовується концептуальна модель «Сутність — Зв'язок» (Entity-Relationship Model, ERM) [12, 5, 26]:

Просторові розширення СУБД та стандарти OGC

Для роботи з геопросторовими даними реляційні СУБД (наприклад, PostgreSQL) доповнюються розширеннями (наприклад, PostGIS), які реалізують стандарти Консорціуму Open Geospatial Consortium (OGC) — Simple Feature Access for SQL [29].

Ключові переваги використання Просторових СУБД у ґрунтознавстві:

1. Просторові типи даних: збереження векторних геометрій (POINT, LINESTRING, POLYGON, MULTIPOLYGON) та растрових масивів безпосередньо у таблицях бази даних.
2. Просторові індекси (GiST / R-Tree): забезпечення миттєвого пошуку, просторових перетинів та просторових об'єднань (Spatial Join) серед мільйонів об'єктів.
3. Просторові оператори та предикати: проведення обчислень на рівні SQL-запитів (ST_Area, ST_Buffer, ST_Intersection, ST_DWithin, ST_Contains) [29].
4. Контроль цілісності геопросторових даних: атрибутивні обмеження (CHECK, FOREIGN KEY) та геометричні валідації (ST_IsValid, ST_MakeValid) запобігають появі топологічних помилок (самоперетинів, накладань полігонів).

Міжнародні стандарти структуризації ґрунтових даних

Для забезпечення сумісності (Interoperability) ґрунтових баз даних на міжнародному та регіональному рівнях застосовуються єдині стандарти та специфікації [19]:

- INSPIRE Soil Data Specification (Директива ЄС INSPIRE): визначає уніфіковану концептуальну схему даних UML для обміну інформацією про ґрунти, розрізи, горизонти та оцінку деградації в країнах Європейського Союзу.
- ISO 19156:2011 (Observations and Measurements): концептуальна модель для опису результатів польових та лабораторних спостережень і вимірювань ґрунтових властивостей.
- GlobalSoilMap.net Specifications: глобальний стандарт опису ґрунтових властивостей у вигляді суцільних растрових сіток із фіксованими глибинними інтервалами (0–5, 5–15, 15–30, 30–60, 60–100, 100–200 см).

3. Інтеграція ГІС та баз даних

Інтеграція геоінформаційних систем (ГІС) та систем управління базами даних (СУБД) створює єдине інформаційне середовище, де просторове розташування об'єктів (геометрія) об'єднується з якісними та кількісними показниками ґрунтового покриву (атрибутами).

Архітектурні моделі інтеграції

У сучасних геоінформаційних проектах із моніторингу та охорони ґрунтів застосовують три основні рівні архітектурної інтеграції:

Файлово-зв'язана модель (Low-level integration. Просторова геометрія зберігається у векторних файлах (Shapefile, GeoJSON, GeoPackage), а табличні атрибутивні дані – у зовнішніх БД або таблицях (DBF, CSV, XLSX). Локальні наукові дослідження, камеральна обробка польових зразків, разові картографічні проекти.

Клієнт-серверна просторова СУБД (Spatial DBMS). Геометрія та атрибути зберігаються в єдиній реляційній базі даних із просторовими розширеннями (наприклад, PostgreSQL/PostGIS, SpatiaLite, Oracle Spatial). ГІС-клієнт (QGIS, ArcGIS Pro) підключається безпосередньо до бази. Підтримка транзакцій (ACID), розмежування прав доступу, просторове індексування (R-Tree, GiST) та висока швидкість виконання просторових запитів [29, 32].

Триланкова Web-ГІС архітектура (Cloud / Enterprise). База даних (Data Storage): Сховище геопросторових даних під управлінням PostgreSQL/PostGIS. Картографічний сервер (GIS Server): Сервіси публікації геоданих (GeoServer, MapServer, QGIS Server), які транслюють дані за допомогою протоколів OGC. Веб-клієнт (Frontend): Інтерактивний веб-інтерфейс (OpenLayers, Leaflet, Mapbox) для аналізу та моніторингу даних користувачами.

Механізми зв'язування даних

Для поєднання графічних шарів ґрунтових карт із результатами лабораторних аналізів використовують два типи зв'язків:

1. Атрибутивне зв'язування (Relational Joins & Relates)

Об'єднання здійснюється за спільним первинним та зовнішнім ключем (Primary Key / Foreign Key).

2. Просторові операції та зв'язування (Spatial Joins) [32]

Об'єднання виконується на основі топологічних відношень між геометріями без використання спільних текстових чи числових ключів:

ST_Intersects — перетин точок відбору проб із полігонами землекористування.

ST_Contains / ST_Within — визначення, у межах яких ґрунтових відмін знаходяться ділянки ерозійного ризику.

ST_Buffer — побудова буферних захисних зон навколо водних об'єктів для обмеження внесення добрив.

Стандартизація та сумісність (Interoperability)

Для обміну ґрунтово-екологічною інформацією між різними відомствами та організаціями застосовують міжнародні стандарти:

Стандарти Open Geospatial Consortium (OGC) [4, 27]:

WMS (Web Map Service): Трансляція растрових зображень картувальних шарів.

WFS (Web Feature Service): Передача векторних об'єктів разом із їх атрибутами для редагування та аналізу.

WCS (Web Coverage Service): Доступ до растрових матриць висот (DEM) або даних ДЗЗ.

Директива INSPIRE (ЄС): Визначає єдину специфікацію просторових даних для теми «Ґрунти» (*INSPIRE Data Specification on Soil*), що забезпечує гармонізацію класифікаторів ґрунтів на міжнародному рівні.

4. Практичні приклади застосування

Інтеграція геоінформаційних систем із просторовими СУБД (PostgreSQL/PostGIS, Oracle Spatial, SpatiaLite, Enterprise Geodatabases) дозволяє перейти від статичного картографування до створення масштабованих

корпоративних геоінформаційних систем, обробки великих масивів геопросторових даних (Big Spatial Data) та забезпечення багатокористувацького доступу в режимі реального часу.

1. Державні кадастрові системи та землевпорядкування. Ведення Державного земельного кадастру (ДЗК), облік земельних ділянок, обмежень й обтяжень у використанні земель, а також оцінка земель. Збереження сотень тисяч полігональних об'єктів у СУБД PostgreSQL/PostGIS із застосуванням просторових індексів (GiST) для миттєвої вибірки. Контроль топологічної цілісності: Перевірка відсутності накладань ділянок чи розривів між ними за допомогою просторових предикатів ST_Overlaps(), ST_Touches(), ST_Contains(). Розрахунок площ та буферних зон: Автоматичне визначення площ ділянок у проекційних системах координат та накладання обмежень від охоронних зон (водоохоронні смуги, ЛЕП) [29].

2. Агроекологічне моделювання та точне землеробство. Моніторинг стану ґрунтового покриву, аналіз деградаційних процесів (водна та вітрова ерозія), оцінка якісного стану земель та оптимізація внесення добрив. Інтеграція векторних контурів полів із часовими серіями спектральних індексів (NDVI, NDWI), точками агрохімічного обстеження ґрунтів та цифровими моделями рельєфу (ЦМР). Визначення ділянок із високим ризиком ерозії шляхом просторового перетину шарів крутості схилів, типів ґрунтів та опадів. Автоматизоване зонування полів для диференційованого обробітку.

3. Екологічний моніторинг та оцінка наслідків надзвичайних ситуацій. Картографування осередків забруднення (наприклад, важкими металами чи вибуховими речовинами на постконфліктних територіях), моніторинг гідроморфних ландшафтів, просторовий аналіз ризиків. Поєднання аналітичного інструментарію ГІС із потужністю SQL-запитів. Для оцінки щільності та просторових кластерів застосовуються просторово-статистичні методи (Kernel Density Estimation, Getis-Ord Gi*, Anselin Local Moran's I) [32]. Автоматичне моделювання зон впливу забруднювача за допомогою створення динамічних буферів та розрахунку векторів переносу речовин.

4. Містобудування, інфраструктура та цифрові двійники (Digital Twins). Ведення містобудівного кадастру, управління підземними та надземними інженерними мережами, створення цифрових двійників об'єктів культурної спадщини та критичної інфраструктури. Інтеграція 3D-даних (хмари точок після LiDAR або 3D SLAM сканування, фотограмметричні деталізовані моделі) із базованими на PostGIS репозиторіями. Використання 3D-просторових функцій PostGIS (ST_3DDistance, ST_Volume) для перевірки просторових конфліктів підземних комунікацій та моделювання об'ємно-просторових параметрів забудови [29, 32].

5. Перспективи розвитку та міжнародний контекст

Перспективи розвитку геоінформаційних технологій та СУБД у сфері охорони ґрунтів орієнтовані на перехід від статичного картографування до інтелектуального, високодеталізованого моніторингу в режимі реального часу та сценарійного прогнозування під впливом кліматичних змін.

Ключові технологічні тренди

- Цифрове картографування ґрунтів (Digital Soil Mapping – DSM) та AI/ML: Застосування алгоритмів машинного навчання (*Random Forest, XGBoost, Deep Neural Networks*) для автоматизованого просторового прогнозування властивостей ґрунту (запаси органічного вуглецю, pH, щільність) на основі коваріат – ЦМР, супутникових індексів та кліматичних матриць.
- Хмарні геопросторові обчислення (Cloud Geospatial Computing): Аналіз великих масивів даних ДЗЗ (Big Spatial Data) за допомогою платформ *Google Earth Engine, Microsoft Planetary Computer* та стандартів *STAC (SpatioTemporal Asset Catalog)* і *COG (Cloud Optimized GeoTIFF)* без потреби локального завантаження даних.
- Інтернет речей (IoT) та сенсорні мережі: Інтеграція мереж бездротових датчиків вологості, температури, солоності та всмоктувального тиску ґрунту безпосередньо у просторові СУБД для безперервного відстеження гідротермічного режиму [29].
- Цифрові двійники ландшафтів (Digital Twins): Побудова 3D/4D-моделей об'єктів та водозборів на основі даних LiDAR, БПЛА та 3D SLAM сканування для симуляції ерозійних процесів, затоплення та акумуляції наносів [33].

Міжнародний контекст та глобальні ініціативи

Ініціатива / Програма	Організація / Впровадження	Основне призначення у сфері охорони ґрунтів
GloSIS (Global Soil Information System)	GSP / FAO (ООН)	Глобальна федеративна система ґрунтових баз даних, заснована на єдиному стандарті обміну даними (SoilDataIE).
EU Soil Strategy for 2030 & EUSO	Європейська Комісія (ЄС)	Стратегія відновлення ґрунтів та Європейська ґрунтова обсерваторія. Гармонізація даних на основі мережі спостережень LUCAS Soil.
Land Degradation Neutrality (LDN)	UNCCD (ЦСР 15.3)	Моніторинг нейтрального рівня деградації земель за трьома субіндикаторами: динаміка покриву, продуктивність та запаси органічного вуглецю (SOC).
INSPIRE Soil Data Specification	Європейський Союз	Директива, що визначає єдину просторову структуру та UML-схеми для міжвідомчого обміну ґрунтовими даними у Європі.

Контрольні питання

1. Яку роль відіграє ГІС у моніторингу ґрунтів?
2. Які процеси ґрунтової деградації можна картографувати за допомогою ГІС?

3. Чому бази даних є важливими для ґрунтозахисних досліджень?
4. Які джерела інформації використовуються для формування баз даних про ґрунти?
5. Як інтеграція ГІС та баз даних забезпечує комплексний аналіз?
6. Що таке тематична карта і як вона створюється у ГІС?
7. Які приклади практичного застосування ГІС у сфері охорони ґрунтів можна навести?
8. Як ГІС допомагає оцінювати ризики ерозії та забруднення ґрунтів?
9. У чому полягає значення карт ризику для прийняття управлінських рішень?
10. Які міжнародні стандарти даних застосовуються у сфері ГІС?
11. Як глобальні бази даних (наприклад, FAO) підтримують ґрунтозахисні дослідження?
12. Які перспективи відкриває використання 3D моделювання у ґрунтозахисних дослідженнях?
13. Чому інтеграція ГІС та баз даних є стратегічним інструментом для сталого землекористування?

Список джерел

1. Добряк Д. С., Канаш О. П., Бабміндра Д. І., Розумний В. М. Класифікація та типізація земель як наукова основа їх раціонального використання й охорони. Київ: Урожай, 2019. 208 с.
2. ДСТУ ISO 19101:2009. Географічна інформація. Еталонна модель. Київ: Держспоживстандарт України, 2012. 44 с.
3. Зацерковний В. І., Бурачек В. Г., Железняк О. О., Терещенко А. О. Геоінформаційні системи і бази даних: монографія. Ніжин: НДУ ім. М. Гоголя, 2014. 492 с.
4. Карпінський Ю. О. та ін. Основи створення інтероперабельних геопросторових даних. Київ: КНУБА, 2023. 302 с.
5. Кузьменко І. С., Москаленко А. А., Заячківська Б. Б. Проєктування бази геопросторових даних для інформаційного забезпечення систем органічного землеробства. *Агроекологічний журнал*. 2026. № 1. DOI: https://doi.org/10.33730/2077-4893.1.2026.354126
6. Основи геоінформаційних систем і бази даних : підручник / О. Є. Поморцева ; Харків. нац. ун-т міськ. гос-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М . Бекетова, 2022. – 346 с.
7. Світличний О.О. Основи геоінформатики: Навчальний посібник /О.О.Світличний, С.В. Плотницький /За заг. ред. О.О. Світличного. Суми: ВТД«Університетська книга», 2006. 295 с.
8. Четверіков Б. В. Геоінформаційні системи і бази даних в агроінженерії: навч. посібник. Львів: Видавництво Львівської політехніки, 2024. 196 с.
9. Advanced Navigation. (n.d.). *Vector data*. Advanced Navigation. Retrieved September 20, 2026, from https://www.advancednavigation.com/glossary/vector-data/
10. Awati, R., Zola, A., & Fontecchio, M. (2025, June 12). *What is spatial data?* TechTarget. https://www.techtarget.com/data-technologies/definition/spatial-data
11. Codd E. F. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*. 1970. Vol. 13, No. 6. P. 377–387.
12. DeMers M. N. Fundamentals of Geographic Information Systems. 4th ed. Wiley, 2009. 480 p.
13. Desmet, P.J.J. and Govers, G. (1996) A GIS Procedure for Automatically Calculating the USLE LS Factor on Topographically Complex Landscape Units. *Journal of Soil and Water Conservation*, 51, 427-433.
14. Elmasri R., Navathe S. B. Fundamentals of Database Systems. 7th ed. Pearson, 2015. 1280 p.
15. ESRI. What is GIS? URL: https://www.esri.com/en-us/what-is-gis/overview
16. European Commission. EU Soil Strategy for 2030: Reaping the benefits of healthy soils for people, food, nature and climate. Brussels: EC, 2021. 26 p. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:52021DC0699
17. FAO. Global Soil Partnership: GloSIS Data Exchange Specifications. Rome: FAO, 2022. 88 p.

18. FAO. Soil Data and Information Systems for Sustainable Soil Management. Rome: FAO, 2021. 96 p.

19. FAO. State of the Art Report on Global Soil Spatial Information. Rome: 19. 142 p.

20. Forrest, M. (2021, December 20). *What is spatial SQL?* Matt Forrest. https://forrest.nyc/what-is-spatial-sql/

21. Getis A., Ord J. K. The Analysis of Spatial Association by Use of Distance Statistics. *Geographical Analysis*. 2010. Vol. 24 (3). P. 189–206. DOI:10.1111/j.1538-4632.1992.tb00261.x

22. IDERA. (2025, June 5). *What is conceptual data modeling?* ER/Studio. https://erstudio.com/blog/conceptual-data-modeling/

23. IDERA. (2024, June 3). *What is logical data modeling?* ER/Studio. https://erstudio.com/blog/logical-data-modeling/

24. Kaldate, S. (2024, May 23). *Evolution of Database Management System (DBMS)*. LinkedIn. https://ua.linkedin.com/pulse/evolution-database-management-system-dbms-soham-kaldate-k2kic

25. Moskalenko A. GIS support of forming spatial decisions on land use. *Mechanization in Agriculture & Conserving of the Resources*. 2021. Vol. 67, No. 3. P. 79–81.

26. Moskalenko A., Andriianov V. Development of a geospatial database for planning land protection measures against wind erosion. *Zemleustriy, kadastr i monitorynh zemel'*. 2024. No. 4. P. 107–113. DOI: https://doi.org/10.31548/zemleustriy2024.04.010

27. Open Geospatial Consortium (OGC). OpenGIS Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture. OGC 06-103r4, 2011.

28. Polytechnic. (2025). Chapter 5: Databases and queries. In *An introduction to GIS*. Saskatchewan Open Education Resources (SaskOER). https://www.saskoer.ca/slicegis/part/chapter-5-databases-and-queries/

29. PostGIS Documentation Team. PostGIS Manual (Versions 3.4 / 3.5). OSGeo. URL: https://postgis.net/documentation/

30. Saskatchewan Polytechnic. (2025). Chapter 4: Spatial data models. In *An Introduction to GIS*. https://www.saskoer.ca/slicegis/part/chapter-4-spatial-data-models

31. Spatial Eye. (n.d.). *What is attribute data in GIS?* https://spatial-eye.com/blog/spatial-analysis/what-is-attribute-data-in-gis/

32. Tool reference (n.d.) ArcGIS https://desktop.arcgis.com/en/arcmap/latest/tools/main/a-quick-tour-of-geoprocessing-tool-references.htm

33. USGS (n.d.) 3D Elevation Program https://www.usgs.gov/3d-elevation-program?

ЗМІСТ

ВСТУП 3
Тема 1. Основи сучасних баз даних та їх роль у ГІС 4
1. Поняття та еволюція баз даних 4
2. Основні моделі та технології баз даних 5
3. Інтеграція баз даних у геоінформаційні системи 9
4. Роль баз даних у моніторингу та охороні ґрунтів 10
Контрольні питання 11
Тема 2. Моделі просторових даних 12
1. Поняття та класифікація просторових даних 12
2. Векторна модель просторових даних 15
3. Растрова модель просторових даних 18
4. Інтегровані та тривимірні моделі просторових даних 23
Контрольні питання 24
Тема 3. Атрибутивні дані та їх інтеграція 25
1. Поняття атрибутивних даних 25
2. Джерела атрибутивних даних 27
3. Інтеграція атрибутивних даних у ГІС 28
Контрольні питання 28
Тема 4. Проектування баз даних 30
1. Основи проектування баз даних 30
2. Концептуальне та логічне моделювання 32
3. Нормалізація баз даних 40
4. Фізичне проектування та оптимізація 42
5. Проектування баз даних для охорони ґрунтів 43
Контрольні питання 43
Тема 5. Мова запитів SQL 45
1. Основи SQL та його роль у базах даних 45
2. Основні оператори SQL 48
3. Запити з умовами та об’єднаннями 50
4. Просторові запити у SQL 52
5. SQL і реляційна алгебра 54
Контрольні питання 57
Тема 6. Просторові операції та аналіз даних у ГІС 58

1. Основи просторових операцій у ГІС 58
2. Основні типи просторових операцій 59
3. Просторовий аналіз та моделювання 60
4. Практичне застосування просторових операцій у ґрунтозахисних дослідженнях 62
Контрольні питання 63
Тема 7. Застосування ГІС та баз даних у сфері охорони ґрунтів 64
1. Роль ГІС у моніторингу ґрунтів 64
2. Використання баз даних у ґрунтозахисних дослідженнях 65
3. Інтеграція ГІС та баз даних 66
4. Практичні приклади застосування 67
5. Перспективи розвитку та міжнародний контекст 68
Контрольні питання 69
Список джерел 71