

Київський національний університет імені Тараса Шевченка
Географічний факультет
Кафедра геодезії та картографії

МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ
ПРАКТИЧНИХ РОБІТ З КУРСУ
«ОСНОВИ ТЕОРІЇ БАЗ ДАНИХ»

КИЇВ - 2020

Методичні вказівки до виконання практичних робіт з курсу “Основи теорії баз даних” / О. Ю. Яценко – К.: КНУ імені Тараса Шевченка, 2020.

Методичне видання містить вказівки для виконання практичних робіт з дисципліни «Основи теорії баз даних». Виконане відповідно до навчального плану та робочої програми дисципліни. Призначено для підготовки студентів 2 курсу ОР «Бакалавр» зі спеціальності 193 Геодезія та землеустрій за програмою Геодезія та землеустрій.

Рецензент: Міхно О. Г., к. т. н., доцент кафедри геодезії та картографії Київського національного університету імені Тараса Шевченка;

Затверджено і рекомендовано до друку вченою радою географічного факультету Київського національного університету імені Тараса Шевченка

© О.Ю. Яценко 2020

© Київський національний університет імені Тараса Шевченка, 2020

ЗМІСТ

Вступ	4
Практична робота № 1. Основи проектування баз даних Діаграми класів. Побудова UML діаграм	6
Практична робота 2. Реляційна та об'єктно-реляційна моделі. Функціональна залежність, ключі та нормалізація.....	11
Практична робота 3. Перевірка відношень в ArcGIS	16
Практична робота 4. Створення схеми бази геоданих. Механізми побудови асоціацій та запитів	18
Рекомендовані джерела	28

ВСТУП

Зважаючи на обсяги геопросторових даних, які створюються на сучасному етапі, бази геопросторових даних та ефективне керування ними стають все більш важливими для підвищення продуктивності ГІС, особливо з урахуванням універсальності технології баз даних та її широкого застосування в інформаційних системах різного призначення. Все більше людей працюють з просторовими даними у своїй повсякденній діяльності. Типовий ГІС-менеджер все частіше буде стикатися із необхідністю вирішення завдань проектування БГД самостійно або у співпраці з консультантами, що виконують відповідну роботу.

В даних методичних вказівках представлені практичні роботи, які доповнюють лекційний матеріал з дисципліни “Основи теорії баз даних”. Виконуючи завдання, студенти набувають навичок проектування баз даних та їх практичної реалізації з використанням сучасного програмного забезпечення.

В процесі виконання практичних робіт, студенти на практиці ознайомляться з:

- основними конструкціями Уніфікованої мови моделювання (UML);
- особливостями реляційної моделі бази даних, включаючи функціональні залежності, ключі та нормалізацію;
- основними компонентами баз геоданих;
- особливостями створення та підтримки персональної бази геоданих ESRI, її редагування та завантаження і керування даними в ній;
- основними правилами утворення з'єднань та зв'язків між класами просторових об'єктів;
- принципами безпосереднього зберігання та оброблення просторових даних в середовищі об'єктно-реляційних СКБД;
- правилами побудова запитів до баз даних з використанням просторово розширеної мови SQL

Методичні вказівки демонструють методики обробки та аналізу баз геоданих, а також містять запитання для закріплення пройденого матеріалу.

Для зручності роботи, перед практичною частиною деяких робіт подані стислі теоретичні відомості, що вивчаються в теоретичному курсі, і які необхідні студентам для виконання практичних завдань.

База даних (БД) у загальному випадку визначається як уніфікована сукупність збережених і відтворених даних, що використовуються у рамках організації. Однак поняття БД сьогодні не ґрунтується на єдиній концепції, скоріше це ціле сімейство пов'язаних між собою понять з предметної сфери, програмного й апаратного забезпечення, аналізу та моделювання даних.

Цільова функція системи бази даних полягає в наданні актуальної та надійної інформації.

Система керування базами даних (СКБД) – система, що ґрунтується на апаратних, програмних та мовних засобах для визначення, створення, введення-виведення, маніпулювання, контролювання, керування та використання даних. За призначенням СКБД забезпечує:

- централізоване керування даними;
- дотримання стандартів;
- безпеку та цілісність даних;
- розподілене спільне використання даних;
- реплікацію даних.
- скорочує надмірність;
- усуває суперечливість даних;

Технологічно СКБД забезпечує підтримку процесів **проектування, реалізації та маніпулювання** базою даних.

Проектування бази даних – це процес побудови порожньої структури (нуль-бази даних), яка буде містити наші дані. Ми маємо визначити такі речі, як тип даних (наприклад, число, текстовий рядок або дата), обмеження на можливі значення або домени (наприклад, вік повинен бути більше нуля) та будь-які зв'язки між елементами у нашій базі даних (наприклад, посилання на кадастровий номер земельної ділянки, на яку оформляється право власності в реєстрі прав).

Реалізація бази даних є процесом наповнення визначеної структури даними. Програмне забезпечення СКБД буде зчитувати і записувати значення на носій (наприклад, комп'ютерний диск), яким вона управляє.

Маніпулювання базою даних включає здатність СКБД вибирати інформацію (за **запитами**) з бази даних або оновлювати значення, що містяться там.

СКБД повинна забезпечити **ефективне, зручне і безпечне** зберігання та доступ до **великих** обсягів **сталих** даних в багатокористувацькому режимі роботи.

Всі ці питання будуть розглянуті при виконанні практичних робіт студентами.

Практична робота № 1.

Основи проектування баз даних Діаграми класів. Побудова UML діаграм

Короткі теоретичні відомості:

Проектування – це процес моделювання або абстрагування реального світу у певну структуру, яка найкраще відповідає цілям нашої бази даних. Цільове призначення бази даних визначає структуру даних,

Модель – це абстракція, що описує систему або підсистему, її можна розглядати як спрощене відображення явища, процесу тощо.

Нотація – це графічний або текстовий набір правил для відображення абстракції.




Кінцевою метою процесу моделювання даних є розроблення задокументованої та формально описаної схеми бази даних.

Уніфікована мова моделювання (UML – Unified Modeling Language) є стандартизованою, універсальною мовою моделювання у сфері розроблення програмного забезпечення. UML включає в себе множину методів графічних нотацій для створення візуальних моделей переважно об'єктно-орієнтованих програмних систем.

UML включає 14 видів діаграм, які розділяють на три категорії за призначенням (структурні, поведінки та взаємодії). Ми зосередимося на діаграмах класів для опису структури даних. **Діаграма класів** є основним конструктивним блоком об'єктно-орієнтованого моделювання.

Об'єкт - це концепція, абстракція або річ, яка має значення для прикладної сфери.

Класи – це групи об'єктів, які мають спільні характеристики або атрибути. Об'єкт є екземпляром класу об'єктів.

Асоціації – це відношення, або зв'язки, між класами об'єктів у моделі. Вони бувають трьох типів (агрегація , композиція , успадкування .

Кратність зв'язку показує скільки об'єктів одного класу може бути зв'язано з об'єктом іншого класу.

- 1..1 один і лише один;
- 0..1 нуль або один;
- 0..* нуль або будь-яке позитивне ціле число; нуль або більше;
- 1..* від одного до будь-якого позитивного цілого числа; один або більше.

В моделюванні даних зустрічається три типи відношень:

- 1:1** один до одного;
- 1:Б** один до багатьох, також може бути Б:1;
- Б:Б** багато до багатьох.

Класи асоціацій – властивості або атрибути долучаються самого зв'язку, а не до окремого класу. Це ситуація, коли атрибут відіграє роль лише тоді, коли виникає асоціація між двома екземплярами.

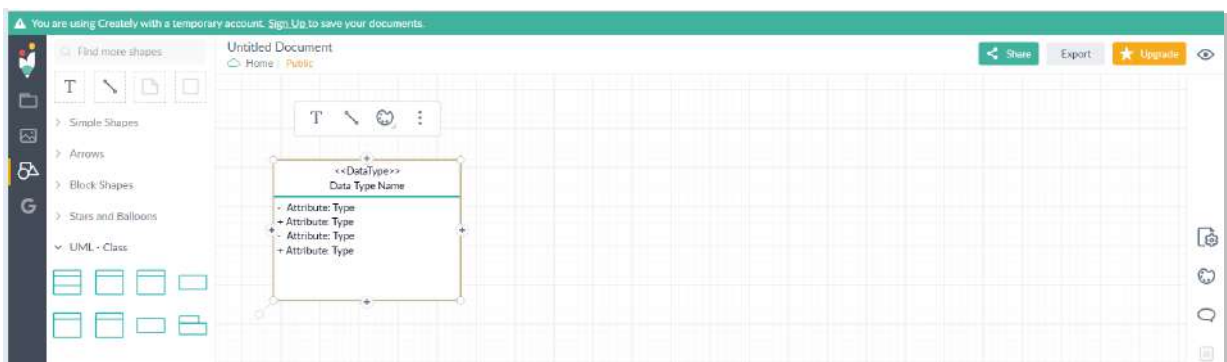
Для побудови UML діаграм можна використати спеціальне програмне забезпечення (наприклад, Microsoft Visio). На практиці розглянемо простий online додаток Creately (www.creately.com), який застосовується для створення різноманітних блок-схем, організаційних діаграм, моделей бізнес-процесів та серед іншого має потужні можливості та бібліотеку шаблонів для створення UML діаграм.

Цей додаток є досить простим у використанні, а online робота з ним дозволяє уникнути необхідності вирішення питань встановлення програмного забезпечення.

Для побудови UML діаграми виберіть розділ **Class Diagrams (UML)** та почніть з чистого листка (**Blank Diagram**).

Creately має декілька різновидів символів класів, які ви можете використовувати (ліворуч).

Для виконання практичного завдання можна використати символ класу з двома частинами для назви класу та його атрибутів (ліворуч). Перетягніть цей символ у зону для креслення (центральна панель інтерфейсу).



Редагуючи елементи діаграми, змініть назву класу на *Земельна ділянка*, а атрибути на *(Ідентифікаційний номер, площа, зонування)*. Створіть другий клас із назвою *Власник* та атрибутами *Ім'я, Поштова адреса і Номер телефону*. Розташуйте ваш новий клас праворуч від вашого класу *Земельна ділянка*.

Створення асоціації. Натисніть на кнопку *Connect* (З'єднати) над вибраним класом, а потім натисніть на інший клас. Якщо ви хочете, щоб лінія сполучалася завжди у певній точці рамки класу, то натисніть на одну із трьох сірих точок на стороні рамки. Якщо ж ви хочете, щоб зв'язок підходив до рамки з будь-якої сторони в найкращому положенні (з точки зору розташування класів між собою), то просто натисніть посередині класу.

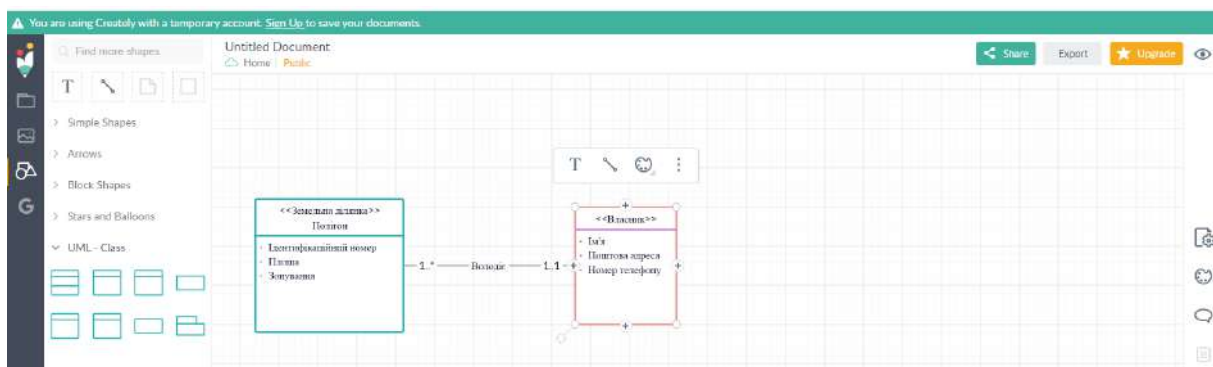
При виділеній лінії асоціації ви маєте бачити над нею між двома класами панель інструментів. Переконайтеся, що на спадаючому меню у правій частині панелі вибрано *Association* (Асоціація).

Ми використаємо просту лінію асоціації між цими двома класами, але саме тут для цілей вашої практичної роботи вам потрібно буде обирати *Composition* (Композиція), *Aggregation* (Агрегація) або Спадкування (в Creately називається *Extends*). Отже, ви повинні створити символізацію на ваших лініях асоціації.

На лінії асоціації текст можна доповнювати або змінювати у трьох місцях: посередині (для назви асоціації), а також на лівій та правій сторонах (для позначення кратності). Пропишіть кратності для відповідних класів, враховуючи, що Власник може мати 1 або більше Земельних ділянок, а Земельна ділянка може бути у власності одного або більше Власників

Посередині лінії асоціації пропишіть текст “Володіє” відповідно до назви асоціації.

Результат має бути подібним до нижче наведеного:



Ви можете переміщувати класи та змінювати їх розміри у ході роботи з ними, можливо з метою звільнення простору для введення нових класів та асоціацій. Асоціації та нотації асоціації (наприклад, кратності) перемістяться разом із класом. За допомогою піктограм параметрів, які з'являються при наведенні курсору на елементи діаграми, ви можете змінити оформлення цих елементів.

За неможливості скористатися спеціальним програмним забезпеченням для створення UML-діаграм, їх можна намалювати як звичайні схеми засобами графічних редакторів або загальнонавжених офісних програм.

Питання для самостійного опрацювання:

1. Дайте приклад класу об'єктів.
2. Використовуючи відповідь на попереднє запитання, поясніть відмінність між класом об'єктів та зразком у цьому класі.
3. Використовуючи відповідь попереднього запитання, поясніть, що є прикладом атрибуту для цього класу?
4. опишіть або створіть UML діаграму класів, що буде прикладом відношення спадкування.
5. опишіть або створіть діаграму класів UML, що буде прикладом відношення композиції.
6. Створіть діаграми класів для класів ВІДДІЛ та СПІВРОБІТНИК, що повинні мати відношення типу 1:M. Припустіть, що відділ може мати нуль співробітників і що кожен співробітник повинен бути у складі відділу. Включіть класи та кратність відношень, але не включайте жодних атрибутів.
7. Побудуйте діаграму класів для компанії зі страхування автомобілів, що має набір клієнтів, кожен з яких має один або більше автомобілів. Кожен автомобіль пов'язаний з нулем або більше зафіксованих аварій. Використайте класи АВАРІЯ, АВТОМОБІЛЬ та КЛІЄНТ.
8. Побудуйте діаграму класів для відношень між КОНТИНЕНТАМИ та КРАЇНАМИ. Дайте пояснення, чому ви вибрали саме такі асоціації та кратності.
9. Проаналізуйте запропоновану модель даних для бази даних генеалогічного дерева. Назвіть щонайменше три проблеми або обмеження щодо діаграми класів, наведеної нижче.



10. Спроектуйте базу даних, що дозволить дослідити гендерні особливості у сфері стану здоров'я та зайнятості. Така база даних є ключовою для розуміння того, як і чому відрізняються проблеми зайнятості та стану здоров'я серед жінок і чоловіків. Аналіз даних має дати відповіді на такі питання:
 - Яка середня тривалість зайнятості або безробіття українців, чи вона відрізняється залежно від віку, статі та рівня освіти?
 - У якій сфері зайнятості роботодавці віддають перевагу працевлаштуванню жінок, а в якій – чоловіків? Чи мають деякі сфери певні уподобання щодо віку співробітників або їх сімейного стану?

- Яка середня заробітна плата українців і як різниться заробітна плата людей різного віку, статі та з різних регіонів?
- Чи кількість певних захворювань вища для чоловіків, ніж для жінок, або для людей певного віку? Чи залежить це від сфери зайнятості?

Створіть діаграму класів для бази даних з метою керування чотирма класами: Регіони, Люди, Вакансія та Захворювання. Ми хочемо включити до запису Люди їх імена, стать, дату народження та сімейний стан. Регіони мають містити назву регіону та його площу в квадратних кілометрах. Ми також маємо записати набір вакансій, що повинен містити посаду, назву роботодавця, сферу (таку, як “виробництво” або “фінанси”) та рівень освіти, потрібний для відповідної вакансії. Зрештою, ми повинні заповнити відомості до позиції Захворювання, включаючи назву хвороби або проблеми зі здоров’ям та її категорія (тобто інфекційні, спадкові).

Модель даних повинна зберігати кілька вакансій для кожної особи в часі, фіксуючи дати початку та завершення періоду, коли особа працювала на відповідній посаді, та середню заробітну плату, яку особа одержувала, займаючи відповідну посаду. Це дозволить нам визначити періоди безробіття для осіб, а також відслідкувати роботу різних людей на однакових посадах з різною заробітною платою. Нам також потрібно вести записи щодо захворювань, які мала кожна особа. Немає потреби фіксувати дати початку та завершення кожної хвороби, яку перенесла особа, достатньо просто записати, які проблеми зі здоров’ям мала особа на певному етапі її життя. Зрештою, нам потрібно зберігати і відомості про те, в якому регіоні зараз мешкає кожна особа. Занотуйте всі припущення, які ви зробили при моделюванні.

11. Створіть діаграму класів для керування об’єктами облаштування парків для довільної державної установи України. Модель повинна містити інформацію щодо парків, об’єктів облаштування конкретного вибраного парку, персоналу, що обслуговує та доглядає об’єкти облаштування. Призначенням такої бази даних є можливість визначення, які об’єкти найбільше потребують догляду і до кого слід звертатися у випадку виявлення проблем щодо об’єкту.

Парки є територіями, що створюються для збереження та рекреації, а їх розмір може змінюватися від дуже малих скверів до великих територій дикої природи. Нам потрібно буде зберігати назву та розмір (в гектарах) кожного парку. Об’єкти облаштування це штучно створені елементи, що потребують обслуговування, наприклад: вбиральні, дороги, столи для пікніку та стежки. Нам потрібно буде мати інформацію про усі об’єкти облаштування у парках, включаючи тип об’єкту, дату встановлення, дату та характер будь-якого технічного обслуговування (ремонт), що був виконаний (наприклад, фарбування столу, перекладання покриття дороги, ремонт даху вбиральні). Нам також буде потрібно зберігати дані про обслуговуючий персонал для кожного парку, включаючи їх імена, номер телефону та посаду.

Практична робота 2.
Реляційна та об'єктно-реляційна моделі.
Функціональна залежність, ключі та нормалізація

Короткі теоретичні відомості:

Модель даних – модель структурування даних в базі даних, яка відповідає формальним описам предметної сфери в інформаційній системі та вимогам СКБД, що використовується. Моделі поділяються на **ієрархічні, мережні, реляційні**.

В основі реляційної моделі лежить **відношення** (таблиця), а також встановлення зв'язків між багатьма таблицями в базі даних.

Геореляційна структура не є моделлю бази даних в сенсі розглянутих ієрархічної, мережної або реляційної моделі, а скоріше є структурою, що адаптує реляційну модель для застосування з метою інтегрування геопросторових даних з уніфікованими середовищами зберігання атрибутивних даних в СКБД.

Об'єктно-орієнтована моделі ґрунтуються на фундаментальних концепціях об'єктно-орієнтованого програмування (ООП), а саме, на поняттях класу та об'єкта. Успішна імплементація об'єктно-орієнтованого підходу в більшості реляційних СКБД дає підстави розглядати об'єктно-реляційну модель даних (ОРМД) як засіб реалізації сучасних баз геопросторових даних.

Знання основ реляційної та об'єктно-реляційної моделей є ключовими для розуміння структури даних сучасних ГІС та ефективного їх використання

Основою реляційної моделі є поняття «відношення», яке дуже схоже з поняттям «таблиця». Для того щоб таблиця була відношенням у формальному сенсі, вона повинна задовольняти такі п'ять умов:

1. Всі записи у певному стовпчику мають бути одного типу.
2. Кожний стовпчик має унікальне ім'я в межах однієї таблиці.
3. Таблиця не повинна містити однакових рядків. Якщо два рядки мають повністю ідентичну інформацію у кожному зі стовпчиків, то вона не є відношенням.
4. Порядок стовпчиків неважливий і не повинен впливати на інформаційний зміст таблиці.
5. Порядок рядків неважливий і не повинен впливати на інформаційний зміст таблиці.

Функціональна залежність – це зв'язок між атрибутами одного відношення. Знаючи значення одного атрибуту, можна однозначно визначити, підрахувати чи

знайти значення іншого. У відношенні R атрибут Y функціонально залежить від атрибуту X, тоді й лише тоді, коли кожному значенню X відповідає в точності єдине значення Y. X при цьому є **детермінантом** відношення.

Ключем називається група з одного чи декількох атрибутів, які однозначно визначають решту атрибутів у рядку відношення. Значення ключа однозначно визначає не просто значення одного з інших атрибутів, як у функціональному відношенні, а повністю значення всіх атрибутів у рядку. Можливі ситуації, за яких існує декілька ключів, тоді всі вони називаються **потенційними ключами**.

Добре організована структура зв'язків повинна задовольняти такі умови:

1. Мінімізація надлишковості. Інформація не повинна повторюватись без потреби.

2. Забезпечення ефективного доступу до даних.

3. Вставка, редагування та вилучення даних повинно відбуватись без негативних побічних ефектів: помилок чи несумісностей. Такі побічні ефекти називаються **Аномаліями**. У базах даних розглядають три типи аномалій: Аномалії **Вилучення**, Аномалії **Вставки** та Аномалії **Редагування**.

Нормалізація – це процес структурування відношень для мінімізації таких негативних побічних ефектів. Суть нормалізації полягає в тому, що кожне відношення повинно мати одну “тему”: містити інформацію лише про один тип залежності.

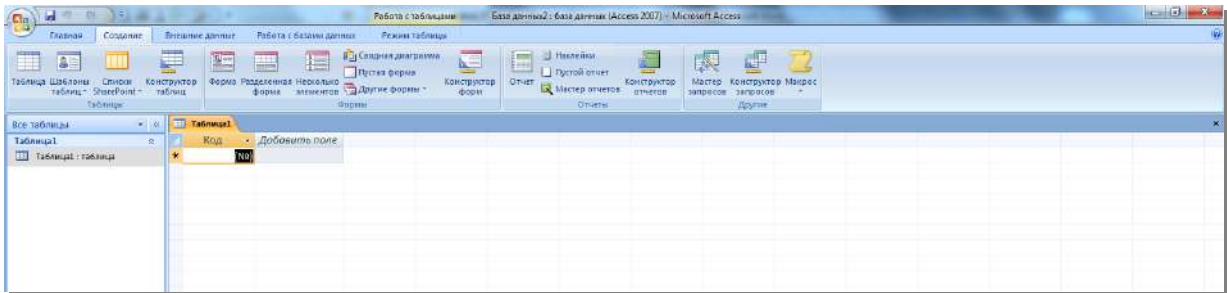
Нормалізація класифікується за ступенем вимог до структури відношень. У зростаючому порядку вимог до структурування, визначено такі нормальні форми:

1. Перша нормальна форма (1НФ)
2. Друга нормальна форма (2НФ)
3. Третя нормальна форма (3НФ)
4. Нормальна форма Бойса-Кодда (НФБК)
5. Четверта нормальна форма (4НФ)
6. П'ята нормальна форма (5НФ)
7. Нормальна форма домен/ключ (ДК/НФ)

Ознайомитись з реляційною структурою сучасних баз даних, основними принципами створення відношень та особливостями роботи з ними на практиці, можна за допомогою Microsoft Access – системи управління базами даних, що входить до складу пакету офісних програм Microsoft Office. Access має широкий спектр функцій зі створення багатofункціональних таблиць, зв'язаних запитів, побудови форм та звітів як в режимі конструктора (*Design*) для підготовленого

користувача з використанням багатьох можливостей налаштування параметрів так і за допомогою "майстрів", що суттєво спрощує роботу початківців.

Програма має інтуїтивно зрозумілий інтерфейс, що змінюється інтерактивно, в залежності від компонента в якому ви працюєте. З лівого боку міститься вікно для перегляду створених компонентів та переходу між ними. Панель інструментів Створення (*Создание / Create*) забезпечує простий доступ до можливостей створення всіх компонентів бази (таблиць, запитів, форм та звітів) в різних режимах:



Всі компоненти бази даних зберігаються в одному файлі, що зручно при зберіганні та обміні даними.

Для ознайомлення з можливостями створення таблиць та використання різних типів даних в полях таблиці, створіть нову базу даних "Факультет", яка складатиметься з двох-трьох взаємопов'язаних таблиць.

Таблиця "Студенти" повинна містити докладну інформацію про студентів (№, Прізвище, Ім'я, По-батькові, Дата народження, Номер студентського квитка, Курс, Група, Фотографія, Характеристика, Адреса проживання, Телефон, Профіль в FB, Стипендія). Яке з цих полів може бути ключем? Проаналізуйте, які типи даних необхідно використати для того чи іншого поля, в яких випадках можна використати майстер підстановки. Змініть параметри відображення полів. Проставте значення за замовчуванням та умови на значення певних числових полів. Продумайте, для заповнення яких полів доцільно використати маски вводу.

Аналогічно створіть таблиці "Кафедри" та "Іспити". Встановіть зв'язки між цими таблицями (інструмент *Зв'язки*) та проаналізуйте їх тип та кратність.

На основі створених таблиць створіть запити, форму та звіт.

Питання для самостійного опрацювання:

A. Функціональна залежність та ключі.

1. Наведіть приклад двох атрибутів, що мають функціональну залежність.
2. Наведіть приклад двох атрибутів, що не мають функціональної залежності.
3. Якщо A є ключем відношення $R(A, B, C)$, то чи може певне (одне й теж саме) значення A з'являтися більше ніж одного разу в $R(A, B, C)$?
4. Розгляньте наступне відношення PROJECT:

Ідентифікатор_Проекту	Ім'я_Співробітника	Заробітна_Плата
100A	Григоренко	6 000
100A	Петренко	5 000
100B	Петренко	5 000
200A	Григоренко	6 000
200B	Григоренко	6 000
200C	Петренко	5 000
200D	Іванов	5 000

Припускаючи, що ви можете визначити усі функціональні залежності на основі лише наведених тут даних, встановіть чи є кожне з наступних тверджень правильним чи хибним:

Ідентифікатор_Проекту \rightarrow Ім'я_Співробітника	так/ні
Ідентифікатор_Проекту \rightarrow Заробітна_Плата	
Ім'я_Співробітника \rightarrow Заробітна_Плата	
Заробітна_Плата \rightarrow Ідентифікатор_Проекту	
Заробітна_Плата \rightarrow (Ідентифікатор_Проекту, Ім'я_Співробітника)	

5. Який ключ у вище наведеному відношенні PROJECT?
6. Визначте потенційні ключі для заданих наборів функціональних залежностей:

Відношення	Залежності	Ключ(і)
$R(A, B)$	$B \rightarrow A$	
$R(A, B, C)$	$A \rightarrow B, B \rightarrow C$	
$R(A, B, C)$	$B \rightarrow C$	
$R(A, B, C)$	$AC \rightarrow B, AB \rightarrow C$	
$R(A, B, C, D)$	$AC \rightarrow B, BC \rightarrow D$	
$R(A, B, C, D, E)$	$A \rightarrow B, A \rightarrow D, B \rightarrow D, B \rightarrow E$	

V. Нормальні форми.

7. Обґрунтуйте, для чого потрібна хороша реляційна структура та нормалізація.
8. Якщо відомо, що $R \in 3NF$, то чи це гарантує її наявність у $BCNF$?
9. Якщо відомо, що $R \in 3NF$, то чи це гарантує її наявність у $2NF$?
10. До якої нормальної форми може бути віднесено відношення *PROJECT* (з Частини *A*) і чому?
11. Опишіть, використовуючи приклади, три різновиди аномалій, пов'язаних з відношенням *PROJECT*.
12. У відношенні $R(A, B, C, D)$ з функціональними залежностями:

$$A \rightarrow B$$

$$BC \rightarrow D$$

$$A \rightarrow C$$

визначте, чи наступні твердження є правильними, чи хибними, з урахуванням наведених вище залежностей:

<i>A</i> є ключем	так чи ні
<i>B</i> є ключем	
<i>C</i> є ключем	
<i>D</i> є ключем	

На основі ваших відповідей щодо ключів R визначте, в якій нормальній формі є відношення R і чому?

13. У відношенні $R(A, B, C)$: $AB \rightarrow C$ та Ключ: AB
В якій нормальній формі є це відношення і чому?
14. У відношенні $R(A, B, C)$: $AB \rightarrow C$, $C \rightarrow B$ та Ключі: AB, AC
В якій нормальній формі є це відношення і чому?

Практична робота 3. Перевірка відношень в ArcGIS

Для роботи Запустіть ArcGIS, за допомогою каталогу підключіться до бази геоданих Assignment.gdb, та завантажте до проекту два класи просторових об'єктів: Street_Polyline та Block, що містяться у базі геоданих Assignment.gdb. Налаштуйте перегляд цих шарів. Відкрийте атрибутивні таблиці цих класів та ознайомтеся з їх вмістом:

Street_Polyline	
<i>Клас полілінійних просторових об'єктів, що відображає вісі вулиць.</i>	
OBJECTID	<i>Унікальний ідентифікатор для кожної вулиці у класі просторових об'єктів. Цей атрибут прописується та керується ArcGIS</i>
Shape	<i>Закодована геометрія (X, Y координати) для кожної полілінії вулиць. Цей атрибут прописується та керується ArcGIS</i>
STR_NAME	<i>Назва вулиці</i>
TYPE_NAME	<i>Тип вулиці, такий як вулиця (вул.) або провулок (пров.) STR_NAME у поєднанні з TYPE_NAME створюють повну назву вулиці (таку, як Кривоноса вул.).</i>
Shape_Length	<i>Лінійна довжина просторового об'єкту вулиці. Вона розрахована в поточних одиницях, встановлених для системи координат класу просторових об'єктів та керується ArcGIS.</i>
Category	<i>Класифікація вулиць за розміром або завантаженістю. Вулиці категорії 1 є найбільш важливими та зазвичай мають найбільший транспортний потік, дороги категорії 5 є найменш важливими.</i>

Block	
<i>Полігони, що відображають ділянки, які обмежені міськими вулицями і зазвичай містять кілька будинків.</i>	
OBJECTID	<i>Унікальний ідентифікатор для кожного кварталу у класі просторових об'єктів. Цей атрибут прописується та керується ArcGIS.</i>
Shape	<i>Закодована геометрія (X, Y координати) для периметру кожного кварталу. Цей атрибут прописується та керується ArcGIS</i>
Shape_Area	<i>Площа просторового об'єкту - кварталу. Вона розрахована в поточних одиницях, встановлених для системи координат класу просторових об'єктів та керується ArcGIS.</i>
Shape_Length	<i>Периметр просторового об'єкту кварталу. Вона розрахована в поточних одиницях, встановлених для системи координат класу просторових об'єктів та керується ArcGIS.</i>

Type1 – Type5	Серія атрибутів, які визначають використання або класифікацію будинків в межах кварталу. При цьому використовуються такі типи, як Хлібокомбінат або Приватний житловий будинок. Подано п'ять стовпчиків атрибутів Type, оскільки в межах кварталу може бути кілька будинків і вони можуть належати до різних типів використання. Ці атрибути подані за ступенем повторюваності, тобто атрибут Type1 містить найбільш повторюваний тип використання в межах полігону, а Type5 містить найменш повторюваний тип. Якщо в межах полігону є лише один тип використання, то поля Type2 – Type5 будуть порожні (<Null>). Там, де дані є недоступними, всі п'ять атрибутів Type можуть бути порожніми.
---------------	--

Розглядаючи функціональну залежність, продивіться всю таблицю. Вам може знадобитися відсортувати таблицю за тим чи іншим полем, що допоможе вам легше виявити функціональні залежності. Щоб це зробити, натисніть правою клавішею миші на сірий заголовок стовпчика та оберіть Сортировать по возрастанию (**Sort Ascending**) або Сортировать по убыванию (**Sort Descending**), відповідно.

Ви можете припустити, що всі порожні або недійсні значення є просто додатковим можливим значенням атрибута. Таким чином, вважатимемо, що для атрибута TYPE_NAME є чотири можливі значення: вул., пров., просп. та невизначений/пустий.

Виходячи із встановлених вами функціональних залежностей, дайте відповіді на наступні питання:

1. Які інші атрибути (якщо такі є) визначає атрибут OBJECTID ?
2. Які інші атрибути (якщо такі є) визначає атрибут STR_NAME ?
3. Які інші атрибути (якщо такі є) визначає атрибут TYPE_NAME ?
4. Які інші атрибути (якщо такі є) визначає атрибут Category ?
5. Виходячи із встановлених вами функціональних залежностей, визначте який атрибут є ключем цієї таблиці? Вам не потрібно враховувати багато атрибутівних залежностей, а лише залежності, які ви знайшли у попередніх чотирьох запитаннях.
6. Виходячи з відповідей на попередні запитання, визначте до якої нормальної форми належить таблиця атрибутів для Street_Polyline та чому?

7. Розгляньте таблицю атрибутів для класу просторових об'єктів *Block*. Опишіть переваги та недоліки зберігання п'яти атрибутів *Type* у пропонуваній спосіб, зважаючи на такі аспекти, як простота використання та ефективність. Запропонуйте альтернативну структуру для цих даних, що може пом'якшити вплив деяких або усіх проблем, які ви визначили (Надайте модель структури даних на мові UML та текстовий опис до неї). Ви можете припустити, що дані будуть використовуватися в ArcMap.

Практична робота 4.

Створення схеми бази геоданих. Механізми побудови асоціацій та запитів

Ця практична робота надає можливість вивчити основні елементи бази геоданих, такі як класи об'єктів та класи просторових об'єктів. Додатково буде розглянуто декілька методів побудови зв'язків між двома об'єктами, що дозволить отримати досвід стосовно:

- створення атрибутивних з'єднань та побудови запитів до них;
- створення просторових з'єднань;
- створення зв'язків та побудови запитів до них;
- створення класів зв'язків та побудови запитів до них;
- виконання операції геообробки для створення класів зв'язків.

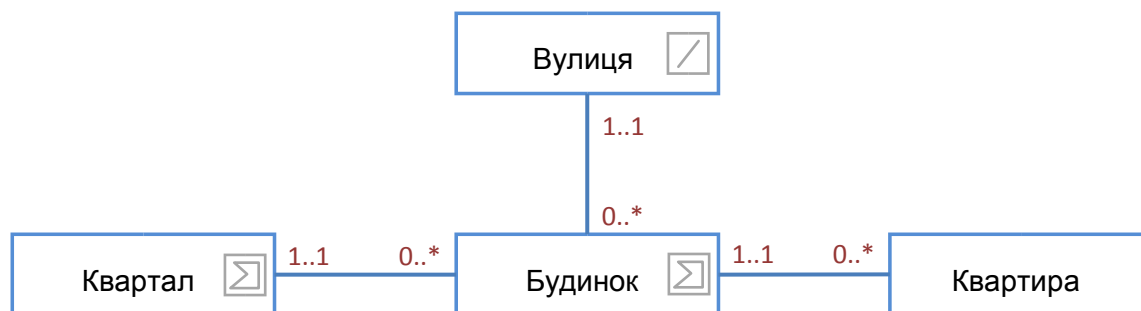
Робота виконується в середовищі ArcGIS. Для виконання цієї практичної роботи потрібно кілька файлів:

Набір даних	Опис	Формат
<i>Apartment</i>	<i>Клас непросторових об'єктів (квартири)</i>	<i>Робоча таблиця Excel</i>
<i>Block</i>	<i>Клас просторових полігональних об'єктів (квартали)</i>	<i>Шейп-файл</i>
<i>Building</i>	<i>Клас просторових полігональних об'єктів (будинки)</i>	<i>Шейп-файл</i>
<i>Street</i>	<i>Клас просторових полілінійних об'єктів (вулиці)</i>	<i>Шейп-файл</i>

Незв'язані набори даних щодо вулиць, міських кварталів та квартир будуть конвертовані в інтегровану базу геоданих. Атрибути включатимуть дані про кількість жителів у кожній квартирі, а також поділ жителів за статтю та віком.

На рисунку нижче показано модель навчального набору даних. Ця діаграма, наприклад, вказує, що будинок може мати нуль або більше квартир, а також, що конкретна квартира знаходиться лише в одному будинку. Вулиці є класом лінійних

просторових об'єктів, квартали та будинки є полігонами, а квартири є класом непросторових об'єктів. Вам уже знайомі ці нотації UML з першої практичної роботи.



Необхідні для цієї практичної роботи дані наявні у різних формах: деякі у вигляді шейп-файлів, інші – у файлі непросторових даних Excel. Слід наголосити на тому, що використані для цієї практичної роботи дані не повинні розглядатися, як точне відображення дійсності. Використані дані є навчальним набором, вони є неповними, а їх частина взагалі вигаданими. Вони будуть використані для ілюстрації того, як може бути виконано моделювання таких зв'язків.

Apartment (Квартира)

Клас непросторових об'єктів містить дані щодо мешканців кожної з квартир будинку. Дані щодо квартир є неповними; лише невелика частина даних щодо будинків та кварталів має відповідні записи щодо квартир.

B_SID	<i>Зовнішній ключ з таблиці Building. Визначає в якому будинку розташована відповідна квартира.</i>
APT_ID	<i>Унікальний ідентифікатор формату довге ціле число для кожної квартири у певному будинку. Тому в кожному з будинків може бути квартира "1".</i>
Residents	<i>Загальна кількість осіб, що мешкають в квартирі.</i>
Adult_Males	<i>Загальна кількість осіб чоловічої статі у віці понад 18 років, що мешкають в квартирі.</i>
Adult_Females	<i>Загальна кількість осіб жіночої статі у віці понад 18 років, що мешкають в квартирі.</i>
Dependents	<i>Загальна кількість дітей у віці до 18 років, що мешкають в квартирі.</i>

Building (Будинок)

Клас полігональних просторових об'єктів, що відображає розташування меж будинків, як би це виглядало при погляді зверху.

B_SID	<i>Унікальний ідентифікатор формату довге ціле число кожного будинку.</i>
--------------	---

NUM	<i>Частина адреси будинку, що стосується його номера на вулиці. Наприклад, якщо адреса будинку просп. Шевченка, 26а, то NUM буде відображати складову адреси “26а”.</i>
POPULATION	<i>Загальне число осіб, що мешкають в будинку.</i>
APARTMENT	<i>Загальна кількість квартир в будинку.</i>
V_OBJ_NAME	<i>Ідентифікатор типу будинку або його використання. Прикладами значень є гуртожиток, приватний житловий будинок та школа.</i>

Block (Квартал)

Клас полігональних просторових об’єктів, що відображає ділянки, обмежені міськими вулицями. Окремий квартал зазвичай містить кілька будинків.

BLK_SID	<i>Унікальний ідентифікатор формату довге ціле число кожного кварталу.</i>
----------------	--

Street (Вулиця)

Клас полілінійних просторових об’єктів, що відображає вісі вулиць.

STR_SID	<i>Унікальний ідентифікатор формату довге ціле число кожної вулиці.</i>
STR_NAME	<i>Назва кожної вулиці. Наприклад, Стуса</i>
TYPE_NAME	<i>Тип вулиці. Значення включають: вул., пров., просп. тощо</i>
CAPTION	<i>Складений підпис вулиці, що включає значення STR_NAME та TYPE_NAME. Наприклад, Стуса вул.</i>


Первинною метою цієї практичної роботи є створення бази геоданих, що містить усі чотири з описаних класів та класи зв’язків, що керують асоціаціями між класами. Будуть розглянуті механізми побудови асоціацій, методи керування ними, а також способи побудови запитів.

Розпочнемо з імпорту класів Building та Apartment і дослідимо деякі методи зв’язування цих об’єктів один з одним, а також побудови запитів до них в ArcGIS.

Запустіть ArcMap, використовуючи вікно ArcCatalog створіть новий файл бази геоданих у вашій робочій папці цієї практичної роботи. Дайте йому доречну назву. На зображеннях з екрану, наведених нижче, база геоданих має назву Assignment4.gdb.

Натисніть правою клавішею миші на назву вашої нової бази геоданих та оберіть *Імпорт/Клас об’єктів (одинич.) (Import / Feature Class (Single))*. Для *Вхідних об’єктів (Input Features)* оберіть Building.shp з вашої папки з вихідними даними.

Визначте назву вашої нової бази геоданих як *Вихідне місцезположення* (англ. Output Location), якщо воно автоматично не було введено, та введіть назву класу просторових об'єктів Building у поле Вихідний клас просторових об'єктів (**Output Feature Class**). Поле Вираз (**Expression**), що дозволяє вам імпортувати лише частину вихідних даних, має залишитись порожнім, а у вікні *Список полей* (**Field Map**), що дозволяє вам перейменувати або переміщувати атрибути з джерела у результуючий клас просторових об'єктів, мають залишитися значення за умовчанням. Натисніть **OK**.

Знову натисніть правою клав'яшею миші на назву вашої бази геоданих та оберіть *Імпорт / Таблиця (одинич.)* (**Import / Table (Single)**). Для *Вхідних рядків* (**Input Rows**) натисніть на кнопку  та перейдіть до вашої папки з вихідними даними. Натисніть двічі на назву Excel-файлу Apartment_Data.xlsx, щоб розгорнути робочі таблиці цього файлу, оберіть таблицю із назвою Apartment\$ та натисніть **OK**. Знову оберіть вашу нову базу геоданих як *Вихідне місцезположення* (**Output Location**) та назвіть *Вихідну таблицю* (**Output Table**) Apartment. Поле *Вираз* (**Expression**) залиште порожнім.

У розділі *Список полей* (**Field Map**) зверніть увагу на те, що для всіх атрибутів визначено тип даних *Double*. Цей вибір є традиційним, оскільки тип даних Double може містити як дробові, так і цілі числа значних розмірів. Усі ці атрибути є або внутрішніми ідентифікаторами (довгі цілі числа), або відносно невеликими значеннями чисельності населення (короткі цілі числа).

Натисніть правою клав'яшею миші на кожен із атрибутів, оберіть *Властивості* (**Properties**) та визначте тип даних для перших двох атрибутів (B_SID та APT_ID) – *Long*, а для решти атрибутів – *Short*.

Тепер у вашій новій базі геоданих має бути клас полігональних просторових об'єктів з назвою Building та клас непросторових об'єктів з назвою Apartment (зверніть увагу на піктограми).



Як показано раніше на діаграмі класів, зв'язки між будинками та квартирами є типу 1:Б; один будинок може мати нуль або більше квартир, кожна квартира розташована лише в одному будинку. Асоціація 1:Б може бути застосована розміщенням первинного ключа класу “1” у клас “Б” як зовнішнього ключа.

Відкрийте атрибутивні таблиці для цих двох класів та зверніть увагу на те, що атрибут B_SID таблиці Building є первинним ключем для будинків, кожен будинок має унікальний номер B_SID. У таблиці Apartment атрибут B_SID є зовнішнім

ключем. Зверніть увагу на те, що певний B_SID може зустрічатися багато разів, оскільки багато квартир можуть знаходитися в одному будинку.

OBJECTID	Shape	B_SID	B_KOATUU	NUM	STR_NAME
375	Polygon	375	4610800000		
376	Polygon	376	4610800000		
377	Polygon	377	4610800000	10	Франка вул.
378	Polygon	378	4610800000	4	Франка вул.
379	Polygon	379	4610800000	31	Грушевського вул.
380	Polygon	380	4610800000	25	Грушевського вул.
381	Polygon	381	4610800000	5	Шухевича вул.
382	Polygon	382	4610800000	46	Зелена вул.
383	Polygon	383	4610800000	6	Чорновола вул.
384	Polygon	384	4610800000	20	Грушевського вул.
385	Polygon	385	4610800000	14	Винниченка вул.
386	Polygon	386	4610800000		
387	Polygon	387	4610800000		

OBJECTID	B_SID	APT_ID	Residents	Adult_Males	Adult_Females	Dependants
145	150	21	4	2	1	1
146	150	22	1	0	1	0
147	150	23	2	0	1	1
148	150	24	3	1	1	1
149	378	1	6	1	1	4
150	378	2	4	1	1	2
151	378	3	5	1	1	3
152	378	4	3	0	1	2
153	378	5	2	2	0	0
154	378	6	4	1	1	2
155	378	7	5	1	2	2
156	378	8	4	2	2	0
157	148	1	5	1	1	3
158	148	2	1	0	1	0
159	148	3	1	0	1	0

Є декілька механізмів побудови асоціації двох об'єктів. Ми розглянемо **З'єднання**, **Зв'язування** та **Класи зв'язків** як засоби сполучення квартир з будинками.

З'єднання є популярним вибором, як механізм для сполучення двох класів, оскільки атрибути обох класів об'єднуються та відображаються разом у класі призначення. Наприклад, якби ми з'єднали атрибути класу Building з класом об'єктів Apartment, усі атрибути класу Building будуть присутні разом з атрибутами класу Apartment. Це працює, оскільки кожна квартира знаходиться в одному і лише в одному будинку.

Натисніть правою клавішею миші на клас об'єктів Apartment у Таблиці змісту та оберіть **З'єднання / З'єднання та Зв'язування (Joins / Joins and Relates)**. Уважно заповніть поля у діалоговому вікні необхідними параметрами та натисніть **OK**.


Опція	Значення	Примітка
Що ви хочете приєднати до шару? (<i>What do you want to join to this layer?</i>)	Приєднати атрибути з таблиці (<i>Join attributes from a table</i>)	Дозволяє вам обрати чи це буде атрибутивним з'єднанням, чи просторовим. Оскільки Apartment не є класом просторових об'єктів, то у цьому випадку немає опції виконати просторове з'єднання.
Виберіть поле шару, на якому буде базуватися з'єднання (<i>Choose the field in this layer that the join will be based on</i>):	B_SID	Це зовнішній ключ таблиці Apartment, що ідентифікує в якому будинку знаходиться кожна квартира.

Виберіть таблицю для приєднання до шару або завантажте її з диску (Choose the table to join to this layer, or load the table from disk)	Building	Це таблиця, яку ми хочемо приєднати до таблиці Apartment. Атрибути таблиці Building будуть доповнені в таблицю Apartment.
Оберіть поле в таблиці, на якому буде базуватися з'єднання (Choose the field in the table to base the join on)	B_SID	Це первинний ключ у таблиці, ми будемо виконувати з'єднання за принципом, що атрибут Building B_SID має дорівнювати атрибуту Apartment B_SID для успішного з'єднання.
Зберегти всі записи (Keep all records)	Вибрати	Вибір цієї опції свідчить, що ми хочемо бачити усі рядки таблиці Apartment незалежного від того, чи було знайдено відповідний рядок у таблиці Building.

Відкрийте таблицю Apartment. Першими будуть відображатися стовпчики таблиці Apartment, які були наявні, коли ми імпортували цей клас. Якщо ви продивитесь таблицю праворуч, то тепер ви побачите, що там є багато нових атрибутів – усі атрибути будинків, такі як номери будинку по вулиці (NUM) були доповнені для кожної квартири. Тепер ми можемо одразу бачити в якому будинку знаходиться кожна квартира.

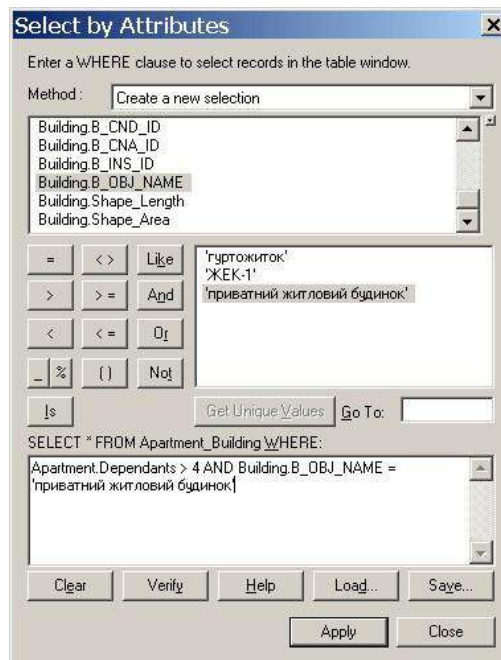
Тепер ми можемо формувати запити до цієї з'єднаної таблиці, як до будь-якої іншої, використовуючи інструмент *Вибрати за атрибутом (Select by Attributes)*.

Наприклад, можемо знайти усі квартири в приватних житлових будинках (атрибут B_OBJ_NAME з таблиці Building), що мають понад 4 дітей у віці до 18 років (атрибут Dependents у таблиці Apartment). Для побудови такого запиту ми зробимо наступне:

- відкриємо атрибутивну таблицю Apartment;
- натиснемо на кнопку *Вибрати за атрибутом (Select by Attributes)*  вверху вікна таблиці атрибутів;
- виберемо Метод “Створити нову вибірку” (**Create a New Selection**);
- натиснемо двічі на *Apartment.Dependants*;
- один раз натиснемо на знак понад (>);
- введемо значення 4;
- один раз натиснемо на кнопку *And* ;
- натиснемо двічі на *Building.B_OBJ_NAME*;
- один раз натиснемо на кнопку рівності (=);

- натиснемо на кнопку *Отримати значення (Get Unique Values)*;
- натиснемо двічі на 'приватний житловий будинок';
- натиснемо кнопку *Застосувати (Apply)*.

Тобто створимо наступний запит:



Питання для самостійного опрацювання:

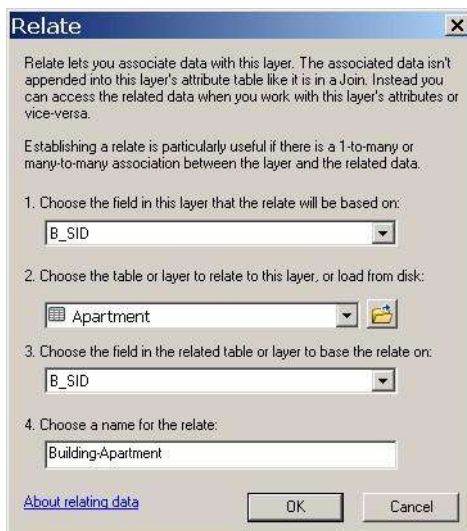
1. Використайте подібний запит, щоб знайти номери будинків по вулиці (атрибут NUM) для квартир, в яких мешкає понад 8 осіб (атрибут Residents > 8).
2. Знайдіть кількість квартир у приватних житлових будинках, в яких мешкає лише одна особа.

З'єднання є ефективним методом для асоціацій типів 1:1 та Б:1, де до таблиці додається один рядок з приєднаної таблиці. Наприклад, коли ми приєднали таблицю Building до таблиці Apartment, там був єдиний рядок для кожної квартири, тому один рядок Building міг бути приєднаний та доповнений в таблицю Apartment.

З'єднання не ефективні для асоціацій типу 1:Б та Б:Б, оскільки ArcMap не може приєднати багато рядків до таблиці з іншої таблиці, що приєднується. Наприклад, коли ми приєднали таблицю Apartment до таблиці Building, ми не могли бачити усі рядки таблиці Apartment, що були зв'язані з відповідним будинком. У таких випадках нам потрібно використовувати інший інструмент для керування подібними відношеннями.

Далі ми розглянемо **Зв'язування**.

Натисніть правою клавішею миші на клас просторових об'єктів Building у Таблиці змісту та оберіть *З'єднання та Зв'язування / Відношення (Joins and Relates / Relate)*. Оберіть B_SID як поле, на якому буде базуватися зв'язування, та дайте логічну назву цьому зв'язку. У наведеному нижче прикладі зв'язок має назву Building-Apartment:



Відкрийте атрибутивну таблицю для класу Building, зверніть увагу, що зараз не видно ніяких атрибутів класу Apartment. Зв'язок не об'єднує атрибути обох класів в єдиний вид, як це робить З'єднання. Механізм керування зв'язками між двома таблицями дещо відрізняється.

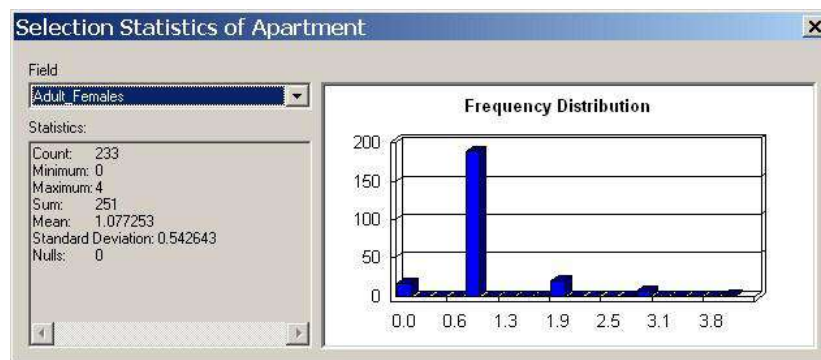
Розпочнемо з простого запиту до класу Building. Скористайтесь опцією *Вибрати за атрибутом (Select by Attributes)*, щоб знайти усі будинки, для яких атрибут "B_OBJ_NAME" = "ЖЕК-1".

Деякі з цих будинків мають зв'язані дані з класу Apartment. Щоб визначити, які квартири пов'язані з вибраними будинками, відкрийте атрибутивну таблицю класу Building та натисніть спадаючий перелік *Зв'язані таблиці (Related Tables)* близько верхнього лівого кута вікна атрибутивної таблиці. Оскільки зараз ми маємо лише один Зв'язок, у вас має бути лише одна опція: зв'язок Building-Apartment, який виконає вибірку у таблиці Apartment. Натисніть *Building-Apartment: Apartment*.

OBJECTID	Shape	B_SID	B_KOATUU	HUM	STR_NAME	POPULATION	APARTMENT	B_CHD_ID
1	Polygon	1	4610300000			0	0	-1
2	Polygon	2	4610300000			0	0	-1
3	Polygon	3	4610300000			0	0	-1
4	Polygon	4	4610300000	10	Шевченка просп.	100	02	-1
5	Polygon	5	4610300000	24	Шевченка просп.	170	63	-1
6	Polygon	6	4610300000	26a	Шевченка просп.	196	70	-1
7	Polygon	7	4610300000	22a	Шевченка просп.	155	70	-1
8	Polygon	8	4610300000	20a	Шевченка просп.	285	100	-1
9	Polygon	9	4610300000	16/1	Шевченка просп.	0	0	-1
10	Polygon	10	4610300000	28a	Шевченка просп.	234	100	-1

Тепер відкриється таблиця класу Apartment, показуючи вам вибрані там записи. Ця вибірка 233 квартир була побудована на основі виконаної нами первинної вибірки у класі Building. Функціонально ArcGIS “поширив” вибірку з таблиці Building у таблицю Apartment на основі створеного нами Зв’язку.

Ми також можемо одержати просту узагальнюючу статистику для цієї вибірки. Наприклад, щоб визначити скільки дорослих осіб жіночої статі мешкає у будинках ЖЕК-1, натисніть правою клавішею миші на сірій заголовок стовпчика атрибуту Adult_Females у таблиці Apartment та оберіть *Статистика (Statistics)*. Це вікно дає нам деякі прості статистичні параметри щодо дорослих осіб жіночої статі у цих квартирах:



Ми також можемо поширити вибірку з таблиці Apartment в зворотному напрямку. Виберіть будь-який рядок у таблиці Apartment, або натиснувши на рядок, або побудувавши запит *Вибрати за атрибутом*. Щоб знайти в якому будинку знаходяться ці квартири, просто оберіть *Зв’язані таблиці (Related Tables) / BuildingApartment: Building*. Відкриється атрибутивна таблиця Building з підсвіченими відповідними будинками.

Питання для самостійного опрацювання:

1. Використайте цю методику для визначення наступного: у будинках з яким атрибутом *B_OBJ_NAME* мешкає більше дітей – ЖЕК-1 чи приватний житловий будинок?
2. Скільки будинків містять квартири, в яких мешкає виключно одна доросла жінка?

Виконавши завдання, вилучіть цей Зв’язок з таблиці Building, натиснувши правою клавішею миші на клас просторових об’єктів Building у Таблиці змісту та обравши *З’єднання та Зв’язки / Вилучити зв’язки / Вилучити всі зв’язки (Joins and Relates / Remove Relates / Remove All Relates)*.

Зв’язки є ефективним шляхом керування асоціаціями будь-якої кратності 1:1, 1:Б, Б:1 та Б:Б. Однак, ці зв’язки зберігаються у файлі MXD і не мають постійного статусу у самих файлах даних; зв’язок **не зберігається** у базі геоданих. Тому коли

інший користувач завантажує класи Building або Apartment в новий документ ArcMap, зв'язок буде відсутній. Встановити **постійну асоціацію** можна, використовуючи **Класи відношень**.

У вікні ArcCatalog натисніть правою клавішею миші на назву вашої бази геоданих та оберіть *Новий / Клас відношень (New / Relationship Class)*, щоб запустити Майстер. Назвіть цей клас відношень Building_Apartment, встановіть Building як *Таблицю/клас просторових об'єктів – Джерело (Origin table/feature class)* та Apartment як *Таблицю/клас просторових об'єктів – Адресат (Destination table/feature class)* Натисніть *Далі (Next)*.

Оберіть *Складне відношення (Composite Relationship)*. Це означає, що квартири є залежними об'єктами, тобто не можуть існувати без будинку, в якому вони розміщені. Коли будинок вилучається з класу Building, усі пов'язані квартири також мають бути вилучені.

Задайте два підписи для асоціації, коли ви “поширюєте” вибірку в одному або протилежному напрямку. Ви можете обрати назви, як ContainsApartment для підпису, що описує пряме відношення (*origin to destination label*) та IsWithinBuilding для підпису, що описує зворотне відношення (*destination to origin label*), однак ви можете вибрати інші підписи на ваш розсуд. Зазначте, що повідомлення мають передаватися *Вперед (від об'єкту-джерела до об'єкта-адресата) (Forward (Origin to Destination))*. Ці підписи дозволяють ArcMap вилучити квартири, коли вилучаються пов'язані будинки.

Оберіть 1:Б як *Кардинальність (Cardinality)* (кратність) для нового зв'язкового класу та натисніть *Далі (Next)*. Оберіть *Ні, я не хочу додавати атрибути у цей клас зв'язків (No, I do not want to add attributes to this relationship class)* та натисніть *Далі (Next)*.

Оберіть B_SID і як *Первинний ключ (Primary Key)*, і як *Зовнішній ключ (Foreign Key)*, та натисніть *Далі (Next)*. Підтвердьте результуючу інформацію та натисніть *Готово (Finish)*.

Цей клас відношень збережеться в базі геоданих та буде доступний для подальшого використання, працюватиме подібно до тимчасового Зв'язування, а механізми/процедури для переміщення наборів вибірок між двома таблицями є ідентичними. По завершенню роботи база геоданих має виглядати, як показано на рисунку нижче.



РЕКОМЕНДОВАНІ ДЖЕРЕЛА

1. Системи керування базами даних для інфраструктури просторових даних. Навчальний посібник / Д. Кейк, А.А. Лященко, В.В. Путренко Ю. Хмелевський, К.С. Дорошенко, М. Говоров. – Планета-Прінт, 2017. – 456 с.
2. Довідка Access [Електронний ресурс]. – Режим доступу: <https://support.office.com/uk-ua/access>
3. Справка ArcMAP. Управление данными [Електронний ресурс]. – Режим доступу: <https://desktop.arcgis.com/ru/arcmap/latest/manage-data/main/what-is-geodata.htm>
4. ДСТУ ISO 19101:2009 "Географічна інформація/геоматика: Географічна інформація – еталонна модель. – К.: Держспоживстандарт України, 2011. – 37с.
5. Даценко Л.М. Остроух В.І. Основи геоінформаційних систем і технологій: навчальний посібник / Л.М. Даценко, В.І. Остроух. – К.: ДНВП «Картографія», 2013. – 184 с.
6. В.І. Остроух, Н.О. Полякова. ГІС, бази даних та цифрова картографія (навчально-методичний посібник для студентів-картографів). – К.: КиївЦНТЕІ, 2009.– 44 с.

При укладанні методичних вказівок було використано зокрема матеріали курсів Спільного проекту ВОУ (Канада), КНУ імені Тараса Шевченка та НТУУ “КПІ” “Закладення основи інфраструктури просторових даних: забезпечення бази в українському уряді для підтримки стабільного економічного зростання”