

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

НУБІП України

Факультет інформаційних технологій

НУБІП України

УДК 004.9:57.07123/.082.12

«ПӨРОДЖЕНО»

«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»

Декан факультету

Завідувач кафедри комп'ютерних наук

інформаційних технологій

НУБІП України

Глазунова О.Г., д.п.н., професор

Голуб Б.Л., к.т.н., доцент

2021р.

30 листопада 2021 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

НУБІП України

на тему: Система аналізу екологічного відбитку людини

Спеціальність 122 Комп'ютерні науки

Освітня програма Комп'ютерний еколого-економічний моніторинг

Орієнтація освітньої програми освітньо-професійна

НУБІП України

Гарант освітньої програми

доц., к.т.н.
(науковий ступінь та вчене звання)

(підпис)

Басараб Р.М.
(ПІБ)

Керівник магістерської кваліфікаційної роботи

старший викладач
(науковий ступінь та вчене звання)

(підпис)

Яшук Дар'я Юріївна
(ПІБ)

НУБІП України

Консультант магістерської кваліфікаційної роботи

доц., к.т.н.
(науковий ступінь та вчене звання)

(підпис)

Голуб Белла Львівна
(ПІБ)

Виконала

(підпис)

Зима Анна Володимирівна

(ПІБ студента)

НУБІП України

КИЇВ-2021

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
Факультет інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц., к.т.н.

Голуб Б.Л.

(науковий ступінь, вчене звання) (підпис) (ІПБ)

“29” жовтня 2021 року

ЗАВДАННЯ

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Зимі Анні Володимирівні

(прізвище, ім'я, по батькові)

Спеціальність 122 Комп'ютерні науки

Освітня програма Комп'ютерний еколого-економічний моніторинг

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи «Система аналізу екологічного відбитку

людини»

затверджена наказом ректора НУБіП України від “29” 10 2020 р. № 1335”С”

Термін подання завершеної роботи на кафедру 30 листопада 2021 р.

Вихідні дані до магістерської кваліфікаційної роботи

1. Тестування на екологічний відбиток
2. Аналіз даних, накопичених в процесі тестування.

Перелік питань, що підлягають дослідженню:

№ з/п	Питання, що підлягає дослідженню	Строк виконання	Примітка
1.	Системний аналіз предметної області.	01.09.2021	
2.	Моделювання системи	10.09.2021	
3.	Розробка системи	17.09.2021	
4.	Результати дослідження	29.10.2021	
5.	Попередній захист	30.11.2021	
6.	Захист	16.12.2021	

Дата видачі завдання “29” жовтня 2020 р.

Керівник магістерської кваліфікаційної роботи

(підпис)

Яшук Д.Ю.

(прізвище та ініціали)

Консультант магістерської кваліфікаційної роботи

(підпис)

Голуб Б.Л.

(прізвище та ініціали)

Завдання прийнята до виконання

(підпис)

Зима А.В.

(прізвище та ініціали студента)

ЗМІСТ

НУБІП України

ВСТУП.....

5

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ..... 8

НУБІП України

1.1. Загальна характеристика предметної області..... 8

1.2. Постановка завдання..... 10

1.3. Діаграма прецедентів..... 11

1.4. Опис методів дослідження..... 14

НУБІП України

1.5. Аналіз наявних рішень..... 17

2 МОДЕЛЮВАННЯ СИСТЕМИ..... 19

2.1. Теоретичні відомості моделювання системи..... 19

НУБІП України

2.2. Діаграма послідовності..... 20

2.3. Діаграма діяльності..... 21

2.4. Діаграма пакетів..... 24

3 РОЗРОБКА СИСТЕМИ..... 26

НУБІП України

3.1. Архітектура системи..... 26

3.2. Середовище розробки..... 27

3.2.1. Реалізація клієнтської частини системи..... 27

3.3. Опис вузлів системи..... 28

НУБІП України

3.4. Структура сховища даних..... 31

3.5. Механізм вилучення, обробки і передачі даних..... 34

3.6. Розгортання OLAP куба..... 35

НУБІП України

3.7. Реалізація отримання даних за допомогою Data Flow..... 38

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ..... 40

4.2 Побудова звітності в середовищі BI..... 42

4.3 Розрахунок KPI.....	46
4.4 Пошук асоціативних правил з використанням Data Mining.....	49
4.5 Задача класифікації з використанням Data Mining.....	50

ВИСНОВКИ.....	53
---------------	----

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
---------------------------------	----

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

ВСТУП

Актуальність: У 21 столітті питання екологічних проблем постало найбільш гостро. Розвиток світового технічного прогресу, зростання населення та нераціональне використання ресурсів планети призвело до екологічної катастрофи, яка вимагає негайного вирішення від місцевого рівня до міжнародного. Екологічна ситуація стосується не тільки нашої країни, але і всієї планети. Це залежить не від життя окремого міста, а цілого людство в цілому.

Населення приділяє мало уваги навколишньому середовищу, нехтуючи всіма ресурсами, які воно дає. Головною проблемою людини є те, що вона не усвідомлює масштабів і серйозності цієї проблеми.

Однією з найбільш вразливих сфер життя суспільства є екологія. Ця сфера безпосередньо впливає на якість життя та стан здоров'я кожного з нас. Дана система направлена на усвідомлення про реальність зниження негативного впливу на навколишнє середовище та зменшення свого екологічного відбитку.

Об'єкт дослідження: процес аналізу екологічного відбитку.

Предмет дослідження: система аналізу екологічного відбитку.

Мета дослідження: дослідження шляхів використання сучасних технологій інтелектуального аналізу в роботі системи аналізу екологічного відбитку.

Завдання, які необхідно вирішити:

- провести системний аналіз об'єкта;
- сформулювати вимоги та структуру до системи аналізу;
- розробити структуру інформаційного забезпечення системи, сховище даних;
- розробити алгоритми аналізу даних, провести дослідження системи.

Методи дослідження: технологія OLAP - використовується для вилучення або видобутку даних, аналізу даних, звітності, для пошуку взаємозв'язків між елементами даних, розрахунку ключових показників ефективності. Технологія Data Mining використовується для вирішення задачі пошуку асоціативних правил та класифікації.

Наукова новизна: вперше було досліджено та проаналізовано систему аналізу обліку екологічного відбитку людини, яка направлена на підвищення екологічної обізнаності громадян.

Апробація результатів дослідження: В процесі розробки магістерської роботи на тему «Система аналізу екологічного відбитку людини» було здійснено наступні публікації:

1. Зима А.В. Програмне забезпечення системи аналізу екологічного відбитку людини. Збірник матеріалів XI Міжнародної науково-практичної конференції молодих вчених «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: ЕКОНОМІКА, ТЕХНІКА, ОСВІТА '2020», 10-11 листопада 2020 року, НУБіП України. Київ, 2020.

2. Зима А.В. Використання OLAP технологій в системі аналізу екологічного відбитку. IV Всеукраїнська науково-практична інтернет конференція студентів і аспірантів «ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ 2021», 29 квітня 2021 року, НУБіП України. Київ, 2021.

3. Зима А.В. Використання сучасних технологій інтелектуального аналізу в системі розрахунку екологічного відбитку. Збірник матеріалів XII Міжнародної науково-практичної конференції молодих вчених «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: ЕКОНОМІКА, ТЕХНІКА, ОСВІТА '2020», 11-12 листопада 2021 року, НУБіП України. Київ, 2021.

Структура роботи. Структурні елементи роботи розміщені в такій послідовності.

У *першому розділі* сформульовано постановку завдання щодо проведення магістерського дослідження, розглядається предметна область, описано методи дослідження, проведено аналіз існуючих рішень, порівняння її з досліджуваною системою.

Другий розділ присвячено моделюванню системи. Побудовано діаграми послідовності, діяльності, пакетів.

У *третьому розділі* описується архітектура спроектованої системи, розглядається інформаційне забезпечення системи, описано вузли системи, які поставляють дані по сховищу та представлено структуру сховища даних, описано побудову розгорнутого куба, інтеграцію отримання даних.

У *четвертому розділі* розглянуто результати дослідження, представлені апаратні та програмні вимоги до реалізації побудованої системи, побудову звітів розрахунок ключового показника ефективності та пошук асоціативних правил.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальна характеристика предметної області

Екологічні проблеми стають все більшою проблемою через вплив людської діяльності на навколишнє середовище. Аналіз екологічного відбитку є природним ресурсом, інструментом оцінки виснаження з високим рівнем точності, який вимірює вплив людської діяльності на навколишнє середовище.

З огляду на можливості екологічного відбитку (ЕВ), було прийнято рішення провести дослідження методів оцінки впливу на навколишнє середовище з використанням різних компонентів і параметрів.

Цілі дослідження - вивчити ефективні компоненти екологічного та запропонувати рекомендації щодо зменшення впливу людини на навколишнє середовище. П'ять компонентів, таких як електроенергія, будинок, побутові потреби, транспорт, харчування вимірюватимуться в ході проведення тестування.

Тиск людини на навколишнє середовище в умовах виснаження природних ресурсів, викидів парникових газів і виробництва відходів невідносно зростає [1].

Зараз екологічний слід людини на 60% вище, ніж здатність до оновлення екосистем. У 2020 році екологічний слід тимчасово зменшився через широкий вплив пандемії COVID-19 у глобальному масштабі, однак це ще далеко до того, що наша планета може поповнити [2].

Протягом останніх десятиліть було розроблено численні методи та інструменти оцінки міри екологічного відбитку. Вони згруповані на Рис.1.1.

Адаптована версія починається з базового підходу мислення життєвого циклу, за яким слідують такі методи, як вуглецевий або водний відбиток, оцінка життєвого циклу, ефективність використання ресурсів та оцінка сталості життєвого циклу

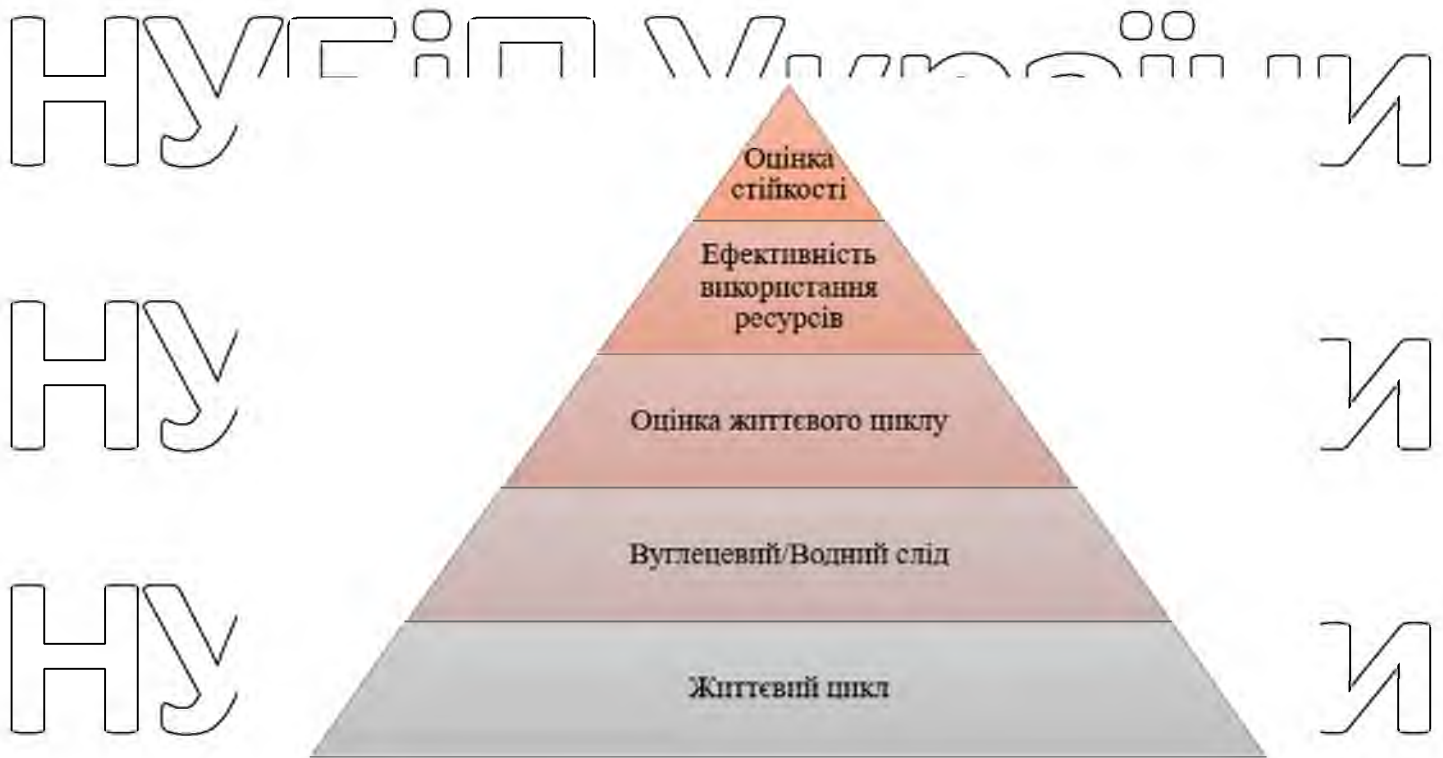


Рис.1.1 Піраміда потреб екологічного відбитку

Методологія розрахунку екологічного сліду розроблена міжнародною організацією Global Footprint Network. Ця організація щорічно перевизначає та коригує методологію розрахунку екологічного відбитку. Єдиної конкретної методології розрахунку ЕВ не існує, однак, існуючі методології досить схожі між собою.

Як правило, ЕВ розраховується за формулами, що наведено нижче.

$$a_i = \frac{c_i}{y_i} \times F \times E_f \quad (1.1)$$

$$E_p = \sum_{i=1}^n a_i \quad (1.2)$$

де a_i – екологічний слід кожного елемента;

c_i – річне споживання елемента;

y_i – продуктивність землі або вихід кожного елемента (кг/га);

F – коефіцієнт прибутковості;
 E_f – коефіцієнт еквівалентності,
 E_p – загальний екологічний слід населення.

Перша формула (1.1) використовується для обчислення кожного компонента ЕВ необхідно для розрахунку ЕВ. Іншими словами, E_f кожного компонента розраховується шляхом ділення річного споживання елемента на продуктивність землі (компонент продукції кг/га), і цей коефіцієнт множиться на коефіцієнт врожайності F та коефіцієнт еквівалентності E_f . Ці коефіцієнти використовуються при переводі землі в глобальні гектари.

Друга формула (1.2) – сума ЕВ/всіх обчислених компонентів, що показує загальний екологічний слід.

Основні сильні сторони екологічного сліду – це його здатність зводити ступінь людського тиску на різні види біопродуктивності в єдине число, можливість створити відчуття надмірного споживання та можливість повідомити результати на широку аудиторію.

Експерти вважають, що настав час зрозуміти екологічний слід і визнати його потужним інструментом для повідомлення про надмірне споживання людства, усвідомлюючи його обмеження в інших сферах [3].

1.2. Постановка завдання

Основною метою системи аналізу екологічного відбитку є зменшення екологічного відбитку людини. Для того щоб досягти дану ціль спочатку потрібно розрахувати ЕВ і надати рекомендації для його зменшення.

Екологічний відбиток (ЕВ), міра вимоги людини або групи людей до глобальних природних ресурсів. Він став одним із найбільш широко використовуваних показників впливу людства на навколишнє середовище і використовувався для того, щоб підкреслити як очевидну нестійкість поточної практики, так і нерівність у споживанні ресурсів між країнами та всередині них.

ВВ дозволяє порівняти потреби окремої людини у природному капіталі з тим обсягом ресурсів, що є у розпорядженні. Розрахунок екологічного відбитку відбувається шляхом проведення тестування, після проходження якого формується результат та рекомендації, направлені на покращення екологічної ситуації.

Сформовано такий перелік питань для аналізу даних.

1. В якій категорії показник екологічного відбитку найбільший?
2. Рівень споживання екологічних ресурсів в розрізі екологічного відбитку за певний період часу.

3. Середній екологічний відбиток по регіонах.

На ці запитання дасть змогу отримати відповіді аналіз проведений з використанням OLAP та Data Mining технологій.

1.3. Діаграма прецедентів

Діаграма прецедентів дозволяє відобразити відношення між акторами та прецедентами в системі. На Рис.1.2 представлено діаграму прецедентів системи аналізу екологічного відбитку людини.

Актори представлені на діаграмі:

- «Громадянин»;
- «Аналітик»;
- «Тестувальник»;
- «Екопровідник».

Громадянин включає в себе такі прецеденти:

- проходження тесту;
- перегляд результату і рекомендацій.

Аналітик включає в себе такі прецеденти:

- перегляд відповідей;
- перегляд рекомендацій.

НУБІП України

- формування рекомендацій;
- формування звітів.

Тестувальник включає в себе такі прецеденти:

- формування тесту;
- перегляд рекомендацій.

НУБІП України

Екопровідник включає в себе такі прецеденти:

- перегляд звітів;
- внесення змін.

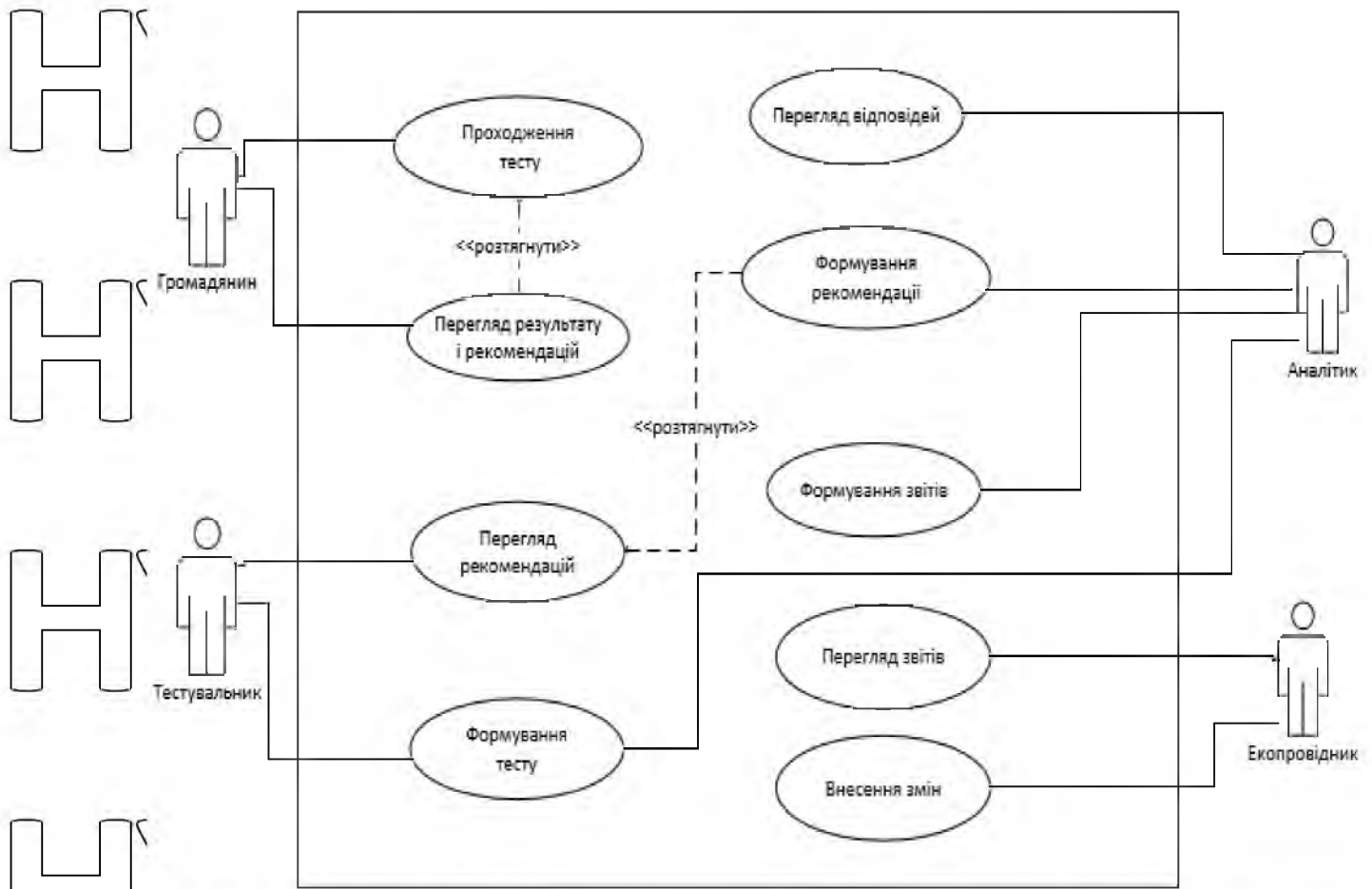


Рис. 1.2 Діаграма прецедентів

Для того, щоб детальніше зрозуміти роботу системи нижче представлено сценарій виконання прецеденту

Назва прецеденту: «Формування тесту»

НУБІП України

НУБІП України

Мета прецеденту: сформувавши тест, за допомогою якого визначатиметься екологічний відбиток.

Головний потік:

1. Формування категорій

2. Формування запитань

3. Формування варіантів відповідей

4. Вибір категорії тесту

//E1 – Відсутня необхідна категорія

5. Вибір запитань для даної категорії

//E2 – Відсутні необхідні питання

6. Відбір варіантів відповідей до конкретного запитання

//E3 – Відсутні необхідні варіанти відповідей

7. Формування запитання для тесту

8. Додавання запитання до тесту

9. Збереження сформованого тесту

//E4 – Кількість питань не достатньо для формування тесту

Альтернативні потоки:

//E1 – Відсутня необхідна категорія

1.1 Додати нову категорію

//E2 – Відсутні необхідні питання

2.1 Додати потрібні питання

//E3 – Відсутні необхідні варіанти відповідей

3.1 Додати необхідні варіанти відповідей

//E4: Кількість питань не достатньо для формування тесту

4.1 Додавати перелік запитань, доки кількість питань не

стане достатньою.

НУБІП України

1.4. Опис методів дослідження

Аналіз даних є невід'ємною частиною у більшості інформаційних систем, так як дозволяє побачити динаміку та результативність системи, робити припущення та гіпотези.

OLAP (Online Analytical Processing) - це потужна технологія для виявлення даних, що включає можливості для безмежного перегляду звітів, складних аналітичних розрахунків та прогнозного планування сценаріїв, бюджету, прогнозу. Вона дає можливість кінцевим користувачам виконувати спеціальний аналіз даних у кількох вимірах, надаючи тим самим уявлення та розуміння, які їм потрібні для кращого прийняття рішень[4].

Технологія OLAP використовується в дослідженні для інтелектуального аналізу даних та для пошуку раніше невідомих взаємозв'язків між елементами даних.

OLAP-системи можна розділити на 3 основні види:

1. **ROLAP** (Relational OLAP - реляційні системи OLAP) – цей вид системи OLAP працює з реляційними базами даних. Звертання до даних здійснюється напряму в реляційну базу даних. Дані зберігаються у вигляді реляційних таблиць. Користувачі мають можливість здійснювати багатомірний аналіз як у традиційних системах OLAP. Це досягається за рахунок застосування інструментів SQL і спеціальних запитів.

2. **MOLAP** (Multidimensional OLAP – багатомірні системи OLAP) - цей вид системи OLAP відноситься до традиційної системи. Відмінна традиційна система OLAP, від інших систем, полягає в попередній підготовці та оптимізації даних. Ці системи, як правило, використовують виділений сервер, на якому забезпечується переробна обробка даних. Дані формуються в багатомірні маси – куби OLAP.

Системи MOLAP є найбільш ефективними при обробці даних, так як вони дозволяють легко реорганізувати та структурувати дані під різними запитами користувачів. Аналітичні інструменти MOLAP дозволяють виконати

складні розрахунки. Іншими перевагами MOLAP є можливість швидкого формування запитів і отримання результатів. Це забезпечується за рахунок попереднього формування кубів.

3. **HOLAP** (Гібридний OLAP – гібридні OLAP системи). Гібридні системи OLAP представляють собою об'єднання систем ROLAP і MOLAP. У гібридних системах постаралися об'єднати переваги двох систем: використання багатомірних баз даних і управління реляційними базами даних. Система HOLAP дозволяє зберігати велику кількість даних у реляційних таблицях, а обробляються дані розміщуються в попередньо побудованих багатомірних кубах. Переваги цього виду системи включають в масштабі даних, швидку обробку даних і плавний доступ до джерелам даних [5].

Основні характеристики OLAP-систем:

Багатомірне концептуальне представлення - системи OLAP дозволяють бізнес-користувачам мати просторове та логічне представлення даних у зберіганні даних.

Багато користувальницька підтримка - оскільки методи OLAP є загальними, технологія повинна забезпечувати операції з базою даних, включаючи пошук, оновлення, контроль адекватності, цілісність та безпеку.

Доступність - OLAP діє як посередник між сховищами даних і клієнтською частиною. Операції OLAP повинні перебувати між джерелами даних (сховищами даних) та інтерфейсом OLAP.

Збереження результатів - результати OLAP зберігаються окремо від джерел даних.

Єдина система документування - збільшення кількості вимірювань або розміру бази даних не має істотного значення для визначення продуктивності звітності системи OLAP.

Дозволяє розрізняти нульові значення та пропущені значення, щоб агрегати розраховувались правильно.

Система OLAP повинна ігнорувати всі відсутність значень і враховувати правильні агреговані значення.

OLAP полегшує користувачам інтерактивний запит і комплексний аналіз.

Дозволяє користувачам виконувати деталі для отримання більш детальної інформації або відображати для агрегування показників за одним бізнес-вимірюванням або за декількома вимірами.

Дає можливість виконати складні обрахунки та порівняння.

OLAP надає результати різними способами, включаючи діаграми та графіки [6].

Основні переваги OLAP: багатовимірні достовірні дані та розрахунки, орієнтовані на бізнес та аналіз в реальному часі

Система OLAP включає в себе наступні компоненти, які використовуються для обробки аналітичних запитів:

Джерело даних – це може бути проміжна база даних або сховище з якого будуть отримуватись дані. У стандартному вигляді дані не оптимізовані для запитів, тому перед використанням необхідне перетворення.

Сховище даних – місце де зберігаються дані для аналізу.

Куб - інструмент подання багатовимірних даних для аналізу.

Data Mining – це процес виявлення закономірностей та іншої цінної інформації з великих наборів даних. Враховуючи еволюцію технологій сховищ даних і зростання великих даних, впровадження методів інтелекту даних швидко прискорилося за останні кілька десятиліть, допомагаючи перетворити вихідні дані в корисні знання.

Інтелектуальний аналіз даних покращив прийняття організаційних рішень завдяки глибокому аналізу даних. Методи аналізу даних, які лежать в основі цих аналізів, можна розділити на дві основні цілі - вони можуть або описати цільовий набір даних, або вони можуть передбачити результати за

допомогою алгоритмів машинного навчання. Ці методи використовуються для упорядкування та фільтрації даних, щоб отримати найцікавішу інформацію.

Інтелектуальний аналіз даних працює за допомогою різних алгоритмів і методів для перетворення великих обсягів даних в корисну інформацію. Один з найпоширеніших це правило асоціації [7].

Не дивлячись на те, що Data Mining та OLAP схожі, оскільки вони працюють з даними для отримання інтелектуальних даних, основна відмінність полягає в тому, як вони працюють з даними. Інструменти OLAP забезпечують багатовимірний аналіз даних та надають зведені дані, але, навпаки, інтелектуальний аналіз даних фокусується на співвідношеннях, шаблонах та впливах набору даних. Інша помітна відмінність полягає в тому, що хоча інструменти інтелектуального аналізу даних моделюють дані та повертають чинні правила, OLAP буде проводити методи порівняння та контрастування в рамках бізнес-вимірювання в режимі реального часу [8].

1.5. Аналіз наявних рішень

Під час огляду інформаційних джерел і аналізу існуючих рішень, було встановлено, що подібних систем орієнтованих на україномовних користувачів, та які відповідають встановленим вимоги до системи не знайдено, а системи, що орієнтовані зарубіжні країни є досить складними або реалізують поставлене завдання не в повному обсязі.

«Footprint wwf» у даному веб ресурсі можна розрахувати екологічний відбиток, але досить вагомим недоліком є відсутність української мови. Також недоліком цих систем є громоздкість. Для користувача, який вперше використовує систему, система може здатися переповненою функціоналом та не досить зрозумілим інтерфейсом.

На Рис. 13 представлена головна сторінка ресурсу

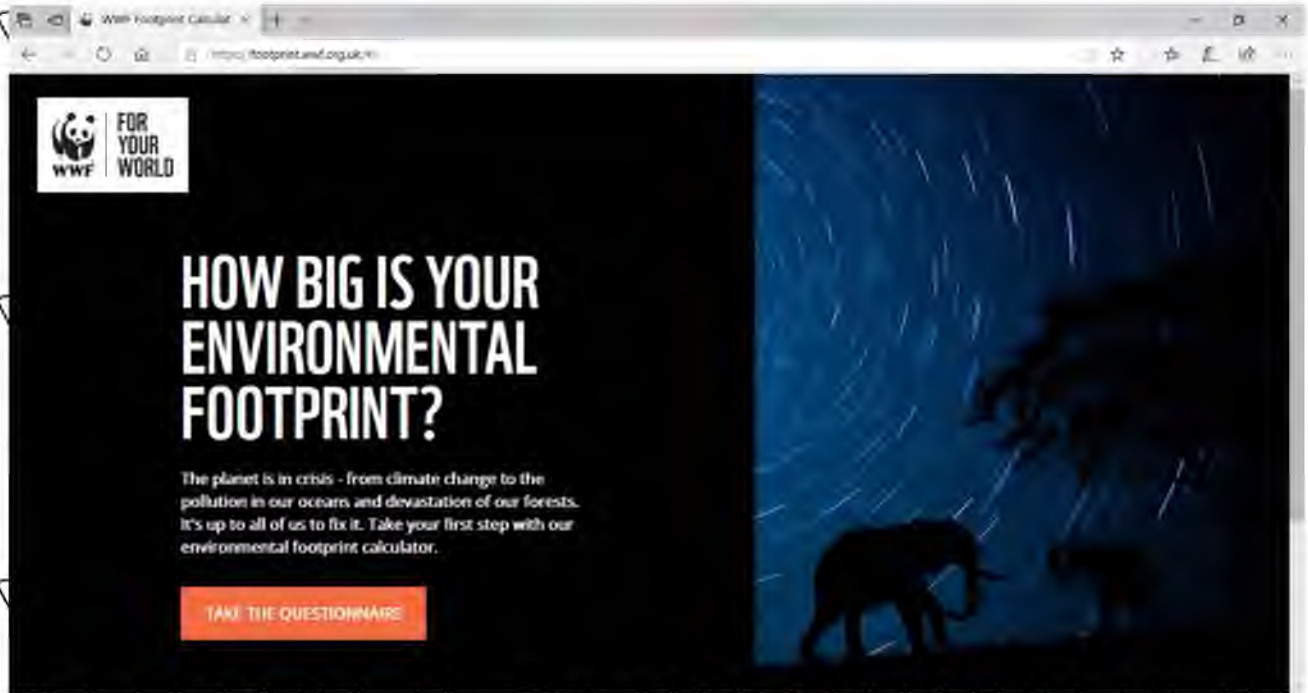


Рис. 1.3 Головна сторінка веб-ресурс «Footprint wwf»

Системи для розрахунку саме екологічного відбитку не зосереджені на українськомовних користувачів, що може стати перешкодою для багатьох громадян.

Також було опрацьовано статтю [9] яка вводить концепцію та метод аналізу екологічного сліду в логістику, щоб оцінити низьковуглецевий рівень логістики та проаналізувати те, що необхідно покращити.

Потім, використовуючи переваги теорії та застосування інформаційних технологій, вивчається системна структура системи аналізу логістичного вуглецевого сліду на основі теорії інформаційних систем. І проектується структура системи, аналізується її модульна функцію та викладаються ключові проблеми.

Дослідження цієї роботи може закласти основу для подальших досліджень щодо сприяння розвитку низьковуглецевої логістики шляхом інформування.

2 МОДЕЛЮВАННЯ СИСТЕМИ

2.1. Теоретичні відомості моделювання системи

Моделювання - процес дослідження реальних систем, який включає в себе побудову моделі, дослідження її властивостей та перенесення одержаної інформації на реальну систему. Моделювання як метод системного аналізу виникло у зв'язку з необхідністю вирішення завдань, які з тих чи інших причин не можуть бути вирішені безпосередньо [10].

Попереднє моделювання предметної області дозволяє скоротити час і строки проведення проектувальних робіт і отримати більш ефективний і якісний проект. Без проведення моделювання предметної області велика ймовірність допущення великої кількості помилок у вирішенні стратегічних

питань, що призводять до економічних втрат і високих витрат на подальше перепроектування системи. Внаслідок цього всі сучасні технології проектування ІС ґрунтуються на використанні методології моделювання предметної області.

Моделювання системи може бути реалізовано різними методиками, які відрізняються своїм підходом. Загалом методології прийнято ділити на функціональні та об'єктно-орієнтовані.

Об'єктно-орієнтоване моделювання - розглядають систему як набір взаємодіючих об'єктів. Об'єкт визначається як предмет або явище, що мають чітко визначене поведінку. Метою застосування даної методики є виділення об'єктів, складових організацію, і розподіл між ними відповідальності за виконувані дії.

Функціональне моделювання - розглядає систему як набір функцій, що перетворює потік інформації у вихідний потік. Процес перетворення інформації споживає певні ресурси. Основна відмінність від об'єктно-

орієнтоване моделювання полягає в чіткому відділенні методів обробки даних від самих даних [11].

Для моделювання предметної області системи аналізу екологічного відбитку людини було використано мову об'єктно орієнтований підхід, а саме уніфіковану мову моделювання UML, так як він надає можливість побудувати більш стійку до змін систему.

UML- уніфікована мова об'єктно-орієнтованого моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем [12].

UML є дуже важливою частиною розробки об'єктно-орієнтованого програмного забезпечення та процесу розробки програмного забезпечення.

2.2. Діаграма послідовності

Діаграма послідовності – один із способів формалізації сценаріїв використання. Її перевага заключається в тому, що на ранніх стадіях опису сценаріїв можна визначити взаємодію компонентів та описати потік повідомлень між цими компонентами. Ці компоненти та потоки повідомлень в подальшому будуть трансформовані в конкретні класи (об'єкти), методи цих об'єктів [13].

Діаграма послідовності складається з об'єктів, що взаємодіють між собою у рамках сценарію та повідомленнями якими вони обмінюються. На діаграмі пунктирною вертикальною лінією показано лінію життя, яка являє собою період життєвого циклу об'єкту під час взаємодії, а також активації, які показують період протягом якого об'єкт виконує операцію.

Діаграму послідовності системи аналізу, яка відображає взаємодію об'єктів впорядкованих за часом наведено на Рис.2. 1.

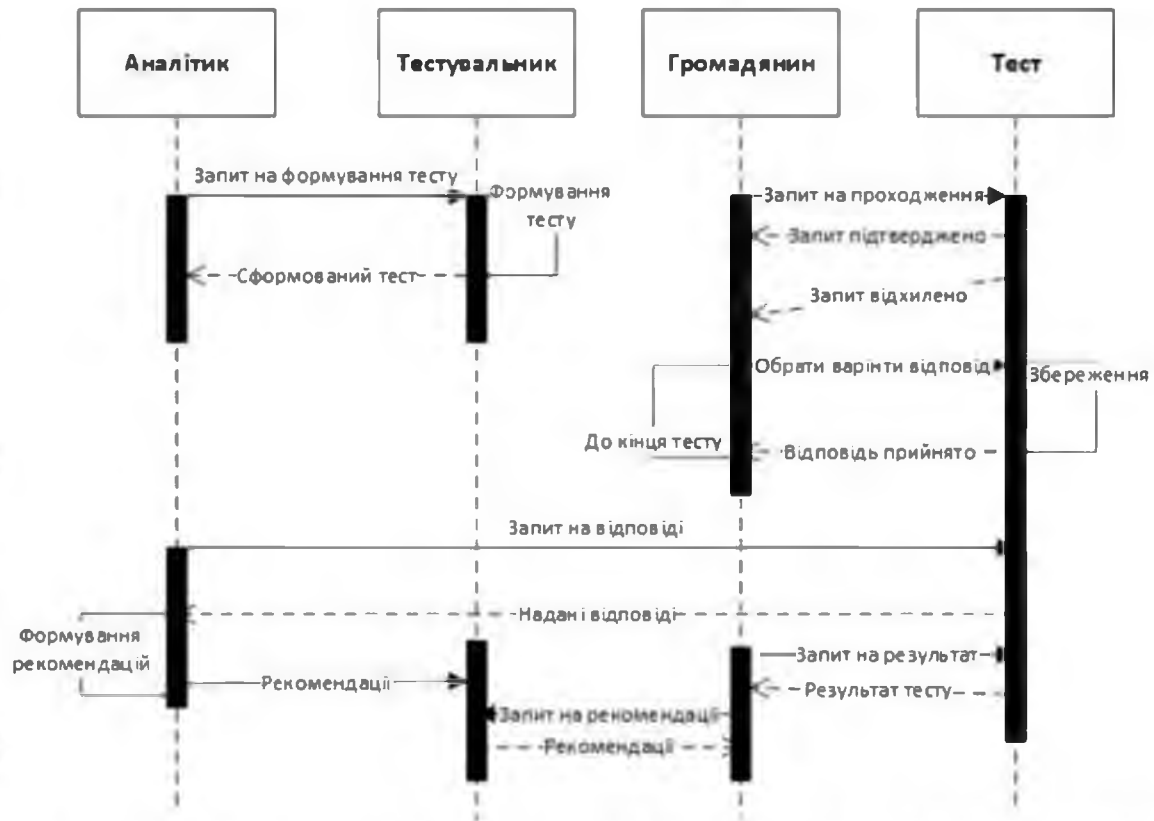


Рис.2.1 Діаграма послідовності

- Аналітик – користувач, який подає завдання та запит на створення тесту.
- Тестувальник – користувач, що формує тест по запиту
- Громадянин – це користувач, що користується системою.
- Тест – система, що виконує процес тестування.

2.3. Діаграма діяльності

Діаграма діяльності - важлива діаграма в UML для опису динамічних аспектів системи. Діаграма діяльності – це в основному блок-схема, яка представляє перехід від однієї діяльності до іншої. Діяльність можна описати як операцію системи. Потік управління переходить від однієї операції до іншої. Цей потік може бути послідовним, розгалуженим або одночасним

Діаграми діяльності розглядають усі типи керування потоком за допомогою різних елементів, таких як розгалуження, з'єднання тощо [14].

На Рис.2.2 та Рис.2.3 зображено діаграми діяльності прецедентів «Формування тесту» та «Проходження тесту» відповідно.

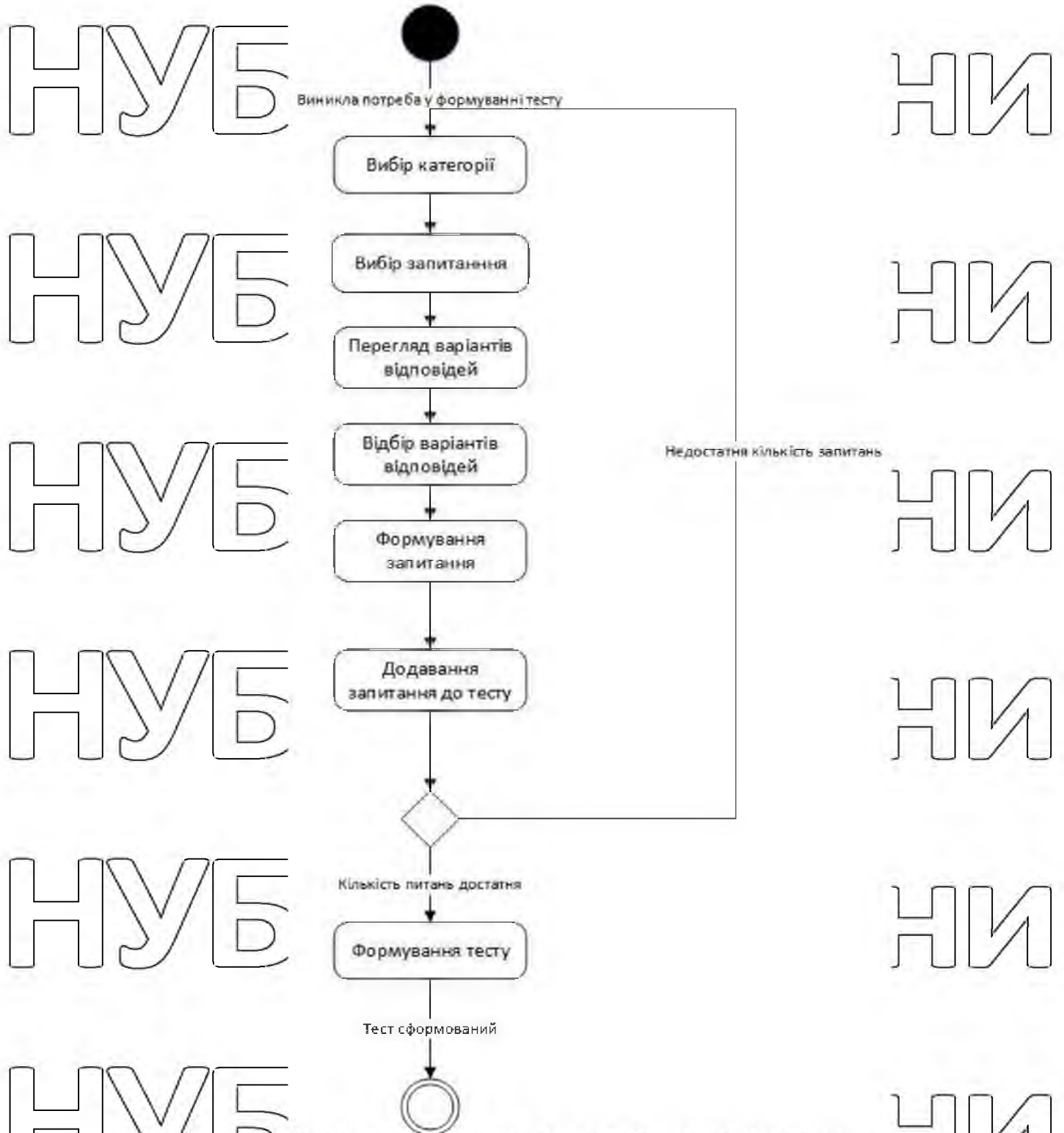


Рис. 2.2 Діаграма діяльності прецеденту «Формування тесту»

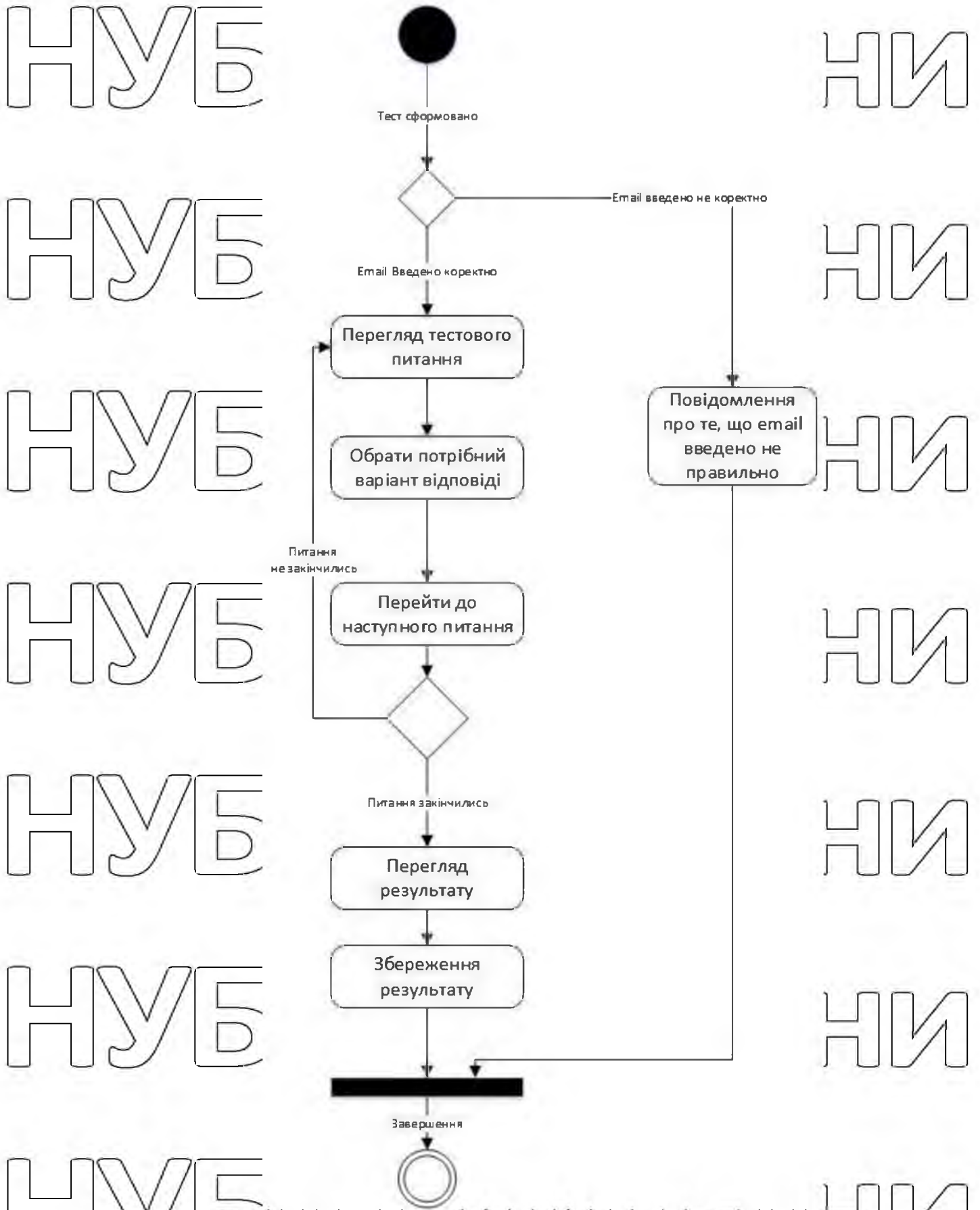


Рис. 2.3 Діаграма діяльності прецеденту «Проходження тесту»

2.4. Діаграма пакетів

Діаграма пакетів – використовуються для структурування елементів систем високого рівня. Пакети використовуються для організації великої системи, яка містить діаграми, документи та інші ключові результати.

Пакет - це набір логічно пов'язаних елементів UML.

Пакети зображуються як папки файлів і можуть використовуватися на будь-якій діаграмі UML [15].

Діаграму пакетів системи аналізу екологічного відбитку людини

зображено на Рис. 2.4.

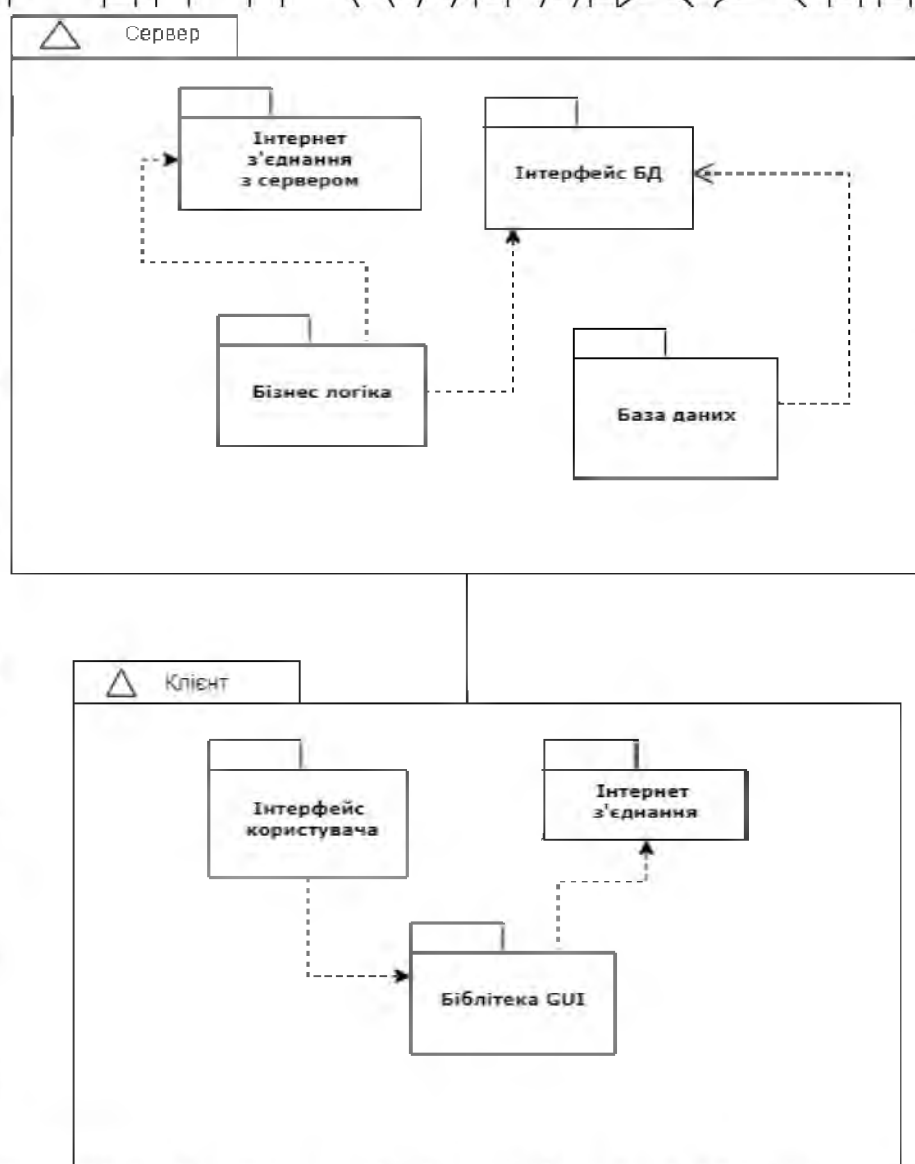


Рис. 2.4 Діаграма пакетів розробленої системи

Пакет «клієнт» містить в собі такі пакети:

- інтерфейс користувача в якому знаходиться опис користувацького інтерфейсу;
- інтернет з'єднання – мережеве з'єднання через інтернет-браузер

з сервером;

- бібліотека GUI – графічна бібліотека.

Пакет «сервер» містить в собі такі пакети.

- бізнес-логіка – реалізація бізнес логіки серверу;
- інтернет-з'єднання – мережеве з'єднання з клієнтом;

- інтерфейс БД – відповідає за інтерфейс з базою даних

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

3 РОЗРОБКА СИСТЕМИ

3.1. Архітектура системи

Архітектура системи є концептуальною моделлю, яка описує структуру і поведінку системи. Система, як правило, складається з декількох компонентів і підсистем. Ці компоненти разом зосереджені на впровадженні всієї системи. Опис архітектури системи - це формальний опис, який пояснює структуру та поведінку всієї системи [16].

Архітектура системи, яка представляє топологію та загальну конфігурацію програмної системи представлена на Рис.3.1.

Було виокремлено такі вузли:

- Робоча станція громадянина з компонентом “Модуль внесення даних”
- Робоча станція тестувальника з компонентом “Модуль внесення даних”
- Сервер бази даних з компонентом База даних
- Сервер сховища даних з компонентом Сховище даних, яке включає в себе зовнішні джерела
- Робоча станція з компонентом “Модуль аналітики”

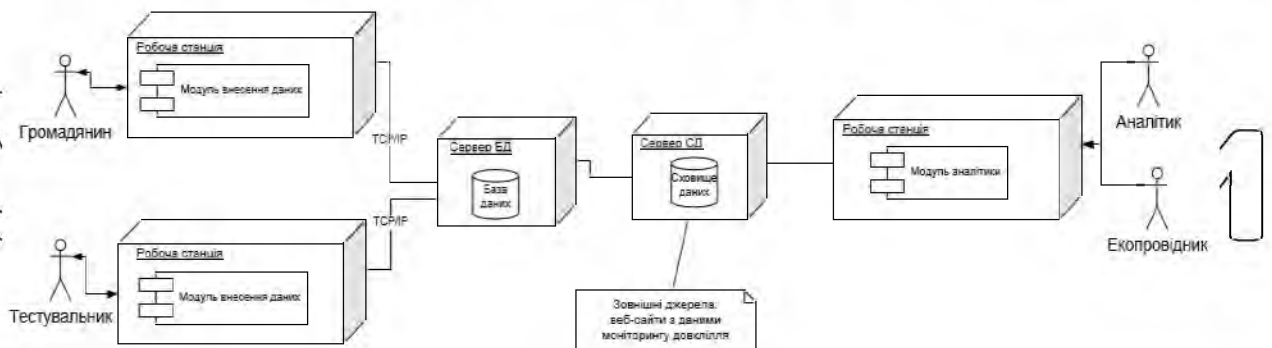


Рис.3.1 Архітектура системи

3.2. Середовище розробки

Для побудови структури сховища даних та розгортання OLAP кубу використовувалось середовище Microsoft Visual Studio з надбудовою служби Analysis Services.

Інтегроване середовище розробки Visual Studio – це середовище для написання, налагодження та складання коду, а також подальшої публікації додатків. Visual Studio включає компілятори, засоби автозавершення коду, графічні конструктори і багато інших функцій для поліпшення процесу розробки [17].

SQL Server Analysis Services – це засіб аналітичних даних, який використовується у службі підтримки прийняття рішень та бізнес-аналітики, він надає можливості моделі семантичних даних корпоративного рівня для бізнес-аналітики (BI), аналізу даних та створення звітів, таких як Power BI, Excel Reporting Services та інші засоби візуалізації даних [18].

Microsoft SQL Server Служби Integration Services (SSIS) є платформою для створення високопродуктивних рішень інтеграції даних та для зберігання даних. Служби SSIS містять графічні інструменти та майстри для побудови та налагодження пакетів; завдання для виконання функцій робочого процесу [19].

3.2.1. Реалізація клієнтської частини системи. Для реалізації клієнтського рівня використовується наступний набір технологій.

HTML (Hyper Text Markup Language) – мова гіпертекстової розмітки, що використовується для створення та візуального відображення веб-сторінок. HTML сторінка містить в собі теги та атрибути які можна переглядати з будь-якого веб-браузера [20].

Саме HTML дозволяє наповнити сторінки певним змістом та є засобом логічної розмітки сторінки. Браузер читає сторінку та відображає його вміст для перегляду користувачем.

CSS – це формальна мова, яка слугує для опису оформлення зовнішнього вигляду документа, створеного з використанням мови розмітки

(HTML, XHTML, XML). Назва походить від англійської Cascading Style Sheets, що означає "каскадні таблиці стилів".

Призначення CSS — відокремлювати те, що визначає зовнішній вигляд сторінки, від її змісту. Якщо документ створений лише з використанням HTML, то в ньому визначається не тільки кожен елемент, а й спосіб відображення (колір, шрифт, положення блоку і т. д.). Якщо ж підключені каскадні таблиці стилів, HTML описує лише черговість об'єктів. А за всі їхні властивості відповідає CSS. У HTML достатньо прописувати клас, не перераховуючи всі стилі щоразу [21].

JavaScript — це текстова мова програмування, яка використовується як на стороні клієнта, так і на стороні сервера, що дозволяє зробити веб-сторінки інтерактивними. Якщо HTML і CSS є мовами, які надають структуру та стиль веб-сторінкам, JavaScript надає веб-сторінкам інтерактивні елементи, які залучають користувача.

Включення JavaScript покращує роботу веб-сторінки, перетворюючи її зі статичної сторінки в інтерактивну. Також JavaScript додає поведінку веб-сторінкам [22].

3.3. Опис вузлів системи

Оперативна база даних складається з таких таблиць:

- «Запитання»;
- «Відповідь»;
- «Категорія»;
- «Громадянин»;
- «Рекомендація»;
- «Результат».

Структура таблиці «Запитання», зберігає в собі інформацію про питання, представлена таким чином:

- id — первинний ключ таблиці

Структура таблиці «**Запитання**», яка містить в собі інформацію про запитання, представлена таким чином:

- category_id – зовнішній ключ, який посилається на таблицю «category», що визначає категорію запитання.
- Текст запитання – символічне поле, в якому міститься текст запитання.

Структура таблиці «**Відповідь**», яка містить в собі інформацію про відповіді, представлена таким чином:

- id – первинний ключ таблиці
- id_question – зовнішній ключ, який посилається на таблицю «question», що визначає запитання.

- Текст відповіді – символічне поле, в якому міститься текст варіантів відповідей.
- Критерій оцінювання – числове значення, яке показує критерій оцінювання варіанту відповіді.

Структура таблиці «**category**», яка зберігає в собі інформацію про категорії запитань, представлена таким чином:

- id – первинний ключ таблиці
- Назва категорії – символічне поле, в якому міститься назва категорії запитання.

Структура таблиці «**Громадянин**», яка містить в собі інформацію користувача системи, представлена таким чином:

- id – первинний ключ таблиці
- email – символічне поле, в якому міститься електронна пошта користувача.

- ПІБ – символічне поле, яке містить в собі прізвище, ім'я, по-батькові.

Структура таблиці «**Рекомендація**», яка зберігає в собі результати з рекомендаціями для користувача, представлена таким чином:

- id – первинний ключ таблиці

• Текст рекомендації – символічне поле, яке містить в собі рекомендації для користувачів.

Структура таблиці «Результат», яка зберігає в собі результати з рекомендаціями для користувача, представлена таким чином:

- id – первинний ключ таблиці;
- Кількість балів – числове поле, яке містить в собі кількість набраних балів;
- recommendation_id – зовнішній ключ, який посиляється на таблицю «Рекомендація», що визначає рекомендацію;

- answer_id – зовнішній ключ, який посиляється на таблицю «Відповідь», що визначає варіанти відповідей.

Оперативна база даних представлена на Рис. 3.2.

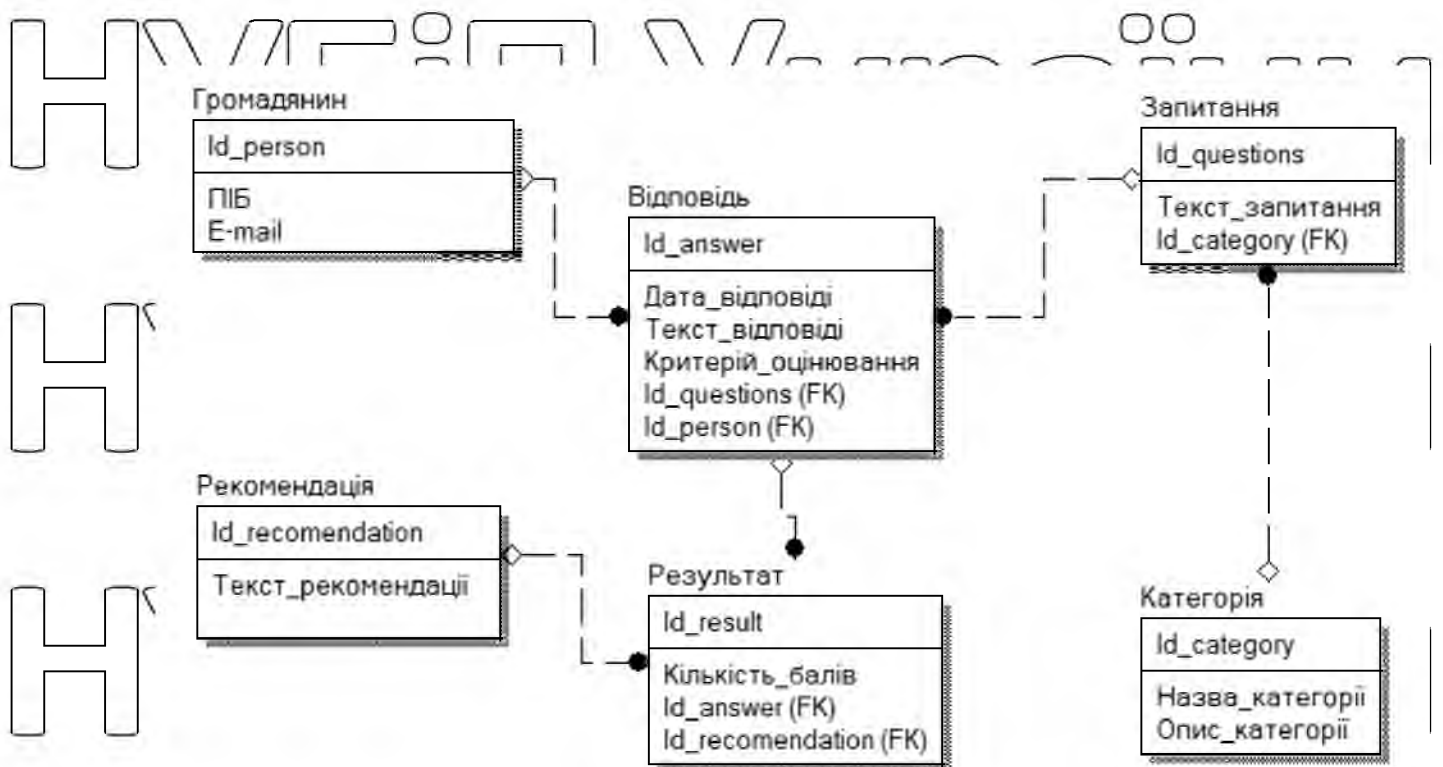


Рис.3.2 Оперативна база даних

НУБІП України

3.4. Структура сховища даних

Сховище даних (англ. data warehouse) — це структурована інформація з різнорідних джерел, що використовується для аналізу даних з метою прийняття стратегічно важливих рішень.

Сховище будується на основі архітектури клієнт-сервер, СУБД та утиліт підтримки прийняття рішень. Дані, отримані в сховищі, стають доступними лише для читання.

Властивості сховища даних:

- предметна спрямованість (інформація організована відповідно до основних аспектів діяльності);
- інтеграція даних (дані в репозиторії надходять з різних джерел і відповідно агрегуються);
- стабільність, незмінність у часі (записи в DW ніколи не змінюються, що представляють собою відбитки даних, зроблені в певний час);
- мінімізація втрат інформації (перед завантаженням у сховище дані фільтруються, зберігаються в певній послідовності та формується деяка кінцева інформація) [23].

Сховище даних проектованої системи складається з таблиці фактів та трьох таблиць вимірів.

Таблиця фактів - є основною таблицею зберігання даних. Як правило, вона містить відомості про об'єкти або події, сукупність яких буде в подальшому аналізуватися.

Таблиці вимірів містять незмінні або рідко змінювані дані. У більшості випадків ці дані представляють собою один запис для кожного члена нижнього рівня ієрархії в вимірі. Таблиці вимірів також містять як мінімум одне поле виміру і, як правило, ключове поле для однозначної ідентифікації члена виміру [24].

Структура сховища представлена на Рис.3.3. Сховище даних складається з таких таблиць фактів та вимірів:

НУБІП України

- Таблиця фактів «**footprint fact**» - основна таблиця сховища даних. Вона містить в собі такі поля, ключові атрибути таблиці вимірів, та такі числові значення як максимальне, мінімальне та середнє значення екологічного відбитку, на підставі яких в

НУБІП України

подальшому будуть отримані агрегатні дані, тобто самі фактичні дані.

- Таблиця вимірів «**category dim**» - таблиця, яка містить в собі ключове поле та назву категорії, та наповнює таблицю фактів змістом.

НУБІП України

- Таблиця вимірів «**date dim**» - часовий вимір, який включає в себе день, місяць та рік проходження тестування та наповнює таблицю фактів змістом.

- Таблиця вимірів «**region dim**» - таблиця, яка містить в собі ключове поле назву регіону та кліматичну зону, та наповнює таблицю фактів змістом.

НУБІП України

Таблиці вимірів і таблиця фактів з'єднуються ідентифікуючими зв'язками. Первинні ключі батьківських таблиць є зовнішніми ключами дочірньої. Тобто, первинний ключ таблиці виміру є зовнішнім ключем таблиці фактів.

НУБІП України

НУБІП України

НУБІП України

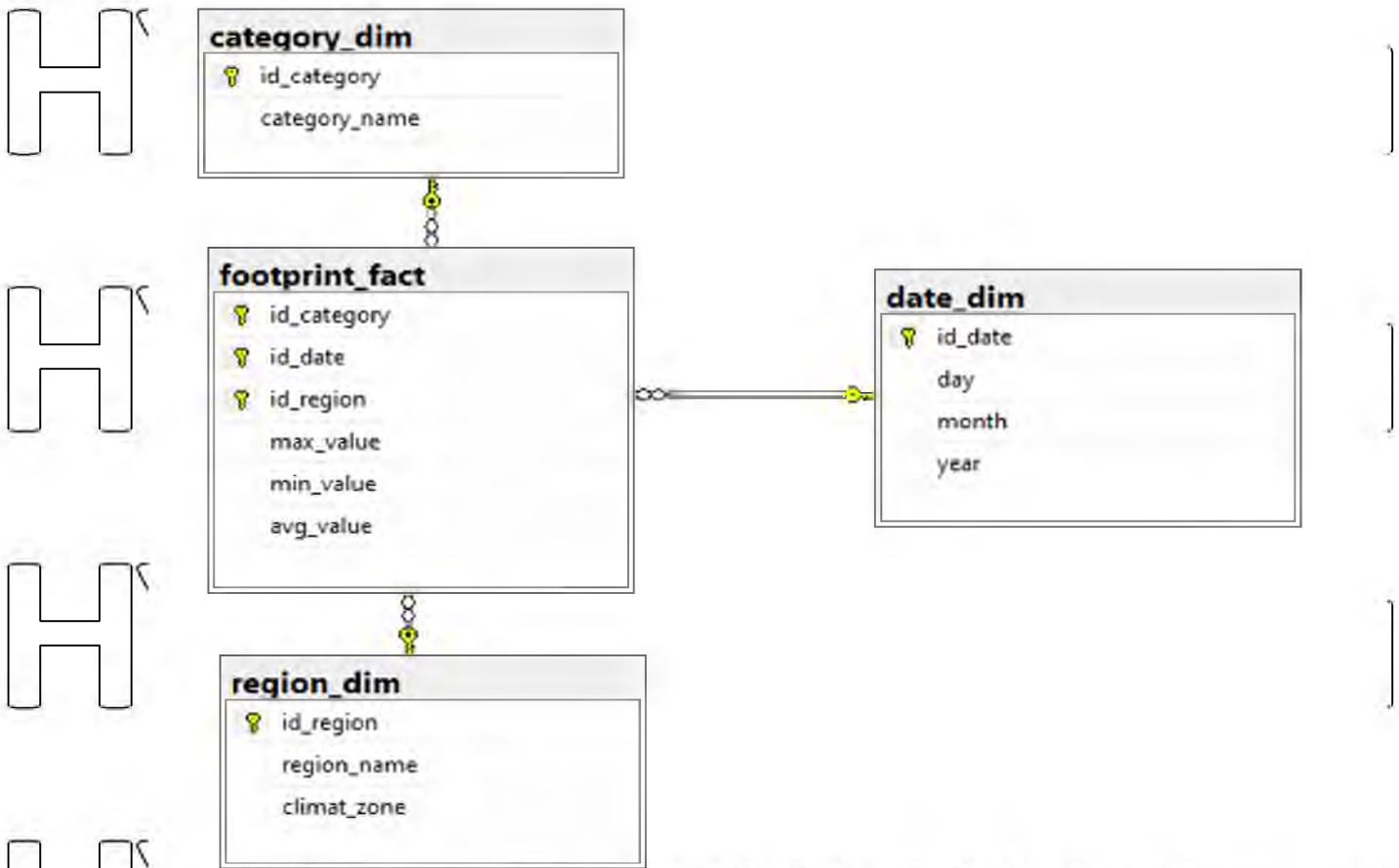


Рис.3.3 Сховище даних системи аналізу екологічного відбитку людини

Процес наповнення сховища даними реалізований з використанням SQL запитів, які беруть дані з відповідних джерел та приводяться до потрібного вигляду вже у сховищі даних.

Алгоритм наповнення сховища даними наведено на Рис.3.4.

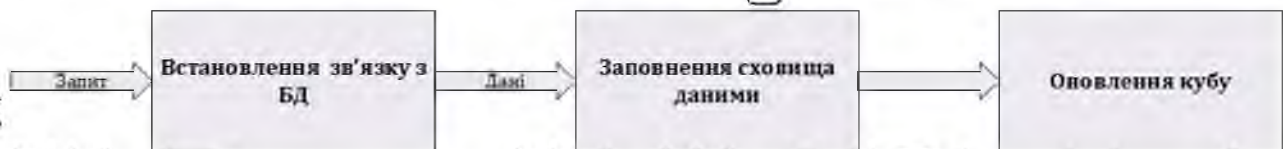


Рис.3.4 Алгоритм заповнення сховища даних

Після оновлення кубу система матиме доступ до аналізу актуальних даних. Лістинг з запитамі наповнення сховища даних наведено у Додатку А.

НУБІП України

3.5. Механізм вилучення, обробки і передачі даних

Служби аналізу Microsoft SQL Server (SSAS) – це технологія зі стека Microsoft Business Intelligence для розробки рішень онлайн-аналітичної обробки (OLAP). Простіше кажучи, ви можете використовувати SSAS для створення кубів, використовуючи дані з вітрин даних / сховища даних для більш глибокого та швидкого аналізу даних.

Куби — це багатовимірні джерела даних, які мають розміри та факти (також відомі як міри) як основні складові. З точки зору реляцій, виміри можна розглядати як основні таблиці, а факти - як вимірні деталі. Ці відомості зазвичай зберігаються в попередньо агрегованому власному форматі, і користувачі можуть аналізувати величезні обсяги даних і дуже легко розділяти ці дані за розмірами. Багатовимірний вираз (MDX) — це мова запитів, яка використовується для запитів до куба, подібно до того, як T-SQL використовується для запиту таблиці в SQL Server [25].

Головною метою використання сховища даних та OLAP є аналіз даних та подання результатів цього аналізу у формі зручній для сприйняття та прийняття рішень.

Використовується підхід, заснований на використанні серверних OLAP-інструментів як проміжна ланка між сховищем даних у вигляді реляційної бази даних та клієнтським додатком. У цьому випадку OLAP-сервер повинен перетворити дані з реляційного сховища на форму, зручну для створення аналітичних звітів - на OLAP-куби.

Як вже зазначалось, кінцевою метою використання сховища даних і OLAP є аналіз даних і представлення результатів цього аналізу в зручному для сприйняття і прийняття рішень вигляді. Використовується підхід, заснований на застосуванні серверних OLAP-засобів в якості проміжної ланки між сховищем даних у вигляді реляційної СУБД і клієнтським додатком. В цьому випадку OLAP-сервер повинен перетворювати дані з реляційного сховища в форму, більш зручну для створення аналітичних звітів - в OLAP-куби.

3.6. Розгорнення OLAP куба

Куб OLAP є структурою даних, яка забезпечує можливість швидкого аналізу даних за рамками обмежень реляційних баз даних. Куби здатні відображати і підсумовувати великі обсяги даних, також надаючи користувачам доступ до будь-яких точках даних з можливістю пошуку. Таким чином, дані можуть бути зведені, фрагментовані і оброблені в міру необхідності для вирішення найбільш широкого спектра питань, що відносяться тих, що цікавить користувача [26].

OLAP куб має структуру, схожу на електронну таблицю. Отриманий досвід аналізу OLAP-куби в основному є багатовимірними базами даних. Вони зберігають дані для аналізу, і багато класичних продуктів BI покладаються на куби OLAP для доступу до інформації компанії для звітів, бюджетів або інформаційних панелей.

Двома ключовими елементами архітектури куба OLAP є:

- Вони агрегують дані.
- Різні набори даних вимагають окремого куба OLAP, оскільки всі дані в одному кубі мають бути пов'язані, щоб їх можна було агрегувати.

Побудова куба для даного проекту відбувається наступним чином:

На початку створюється проект Cube за допомогою служби Analysis Services, після цього створюється джерело даних.

Джерело даних є джерелом всіх даних, що містяться в кубі OLAP. Куб

OLAP підключається до джерела даних для читання і обробки необроблених даних шляхом виконання агрегування і обчислень пов'язаних з ними заходів.

Процес створення джерела даних наведено на Рис.3.4.

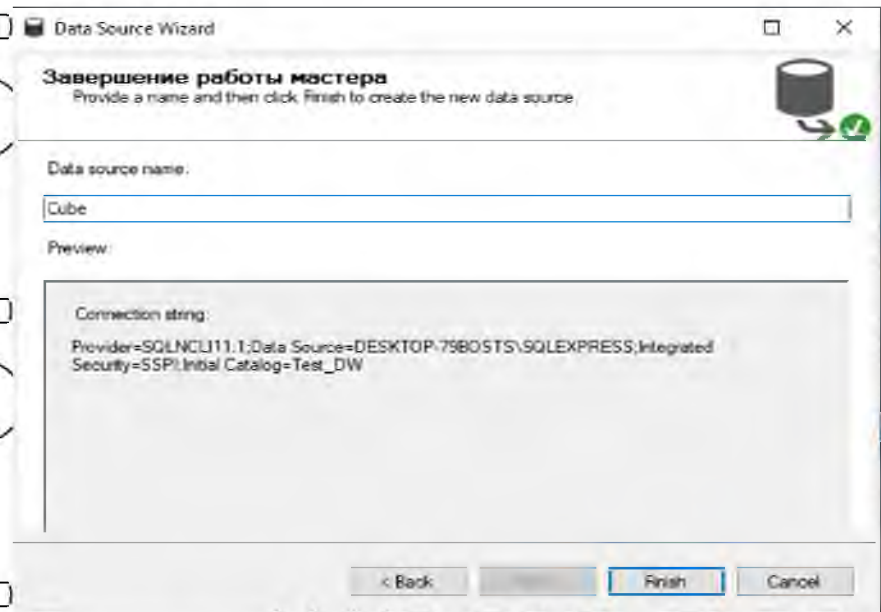


Рис.3.4 Процес створення джерела даних

Подання джерела даних (DSV) - це колекція уявлень, що представляють таблиці вимірювань, фактів і допоміжних вимірів з джерела даних. Подання джерела даних відображає всі відносини між таблицями, в тому числі первинні і зовнішні ключі.

Процес створення представлення джерела наведено на Рис.3.5-3.6.

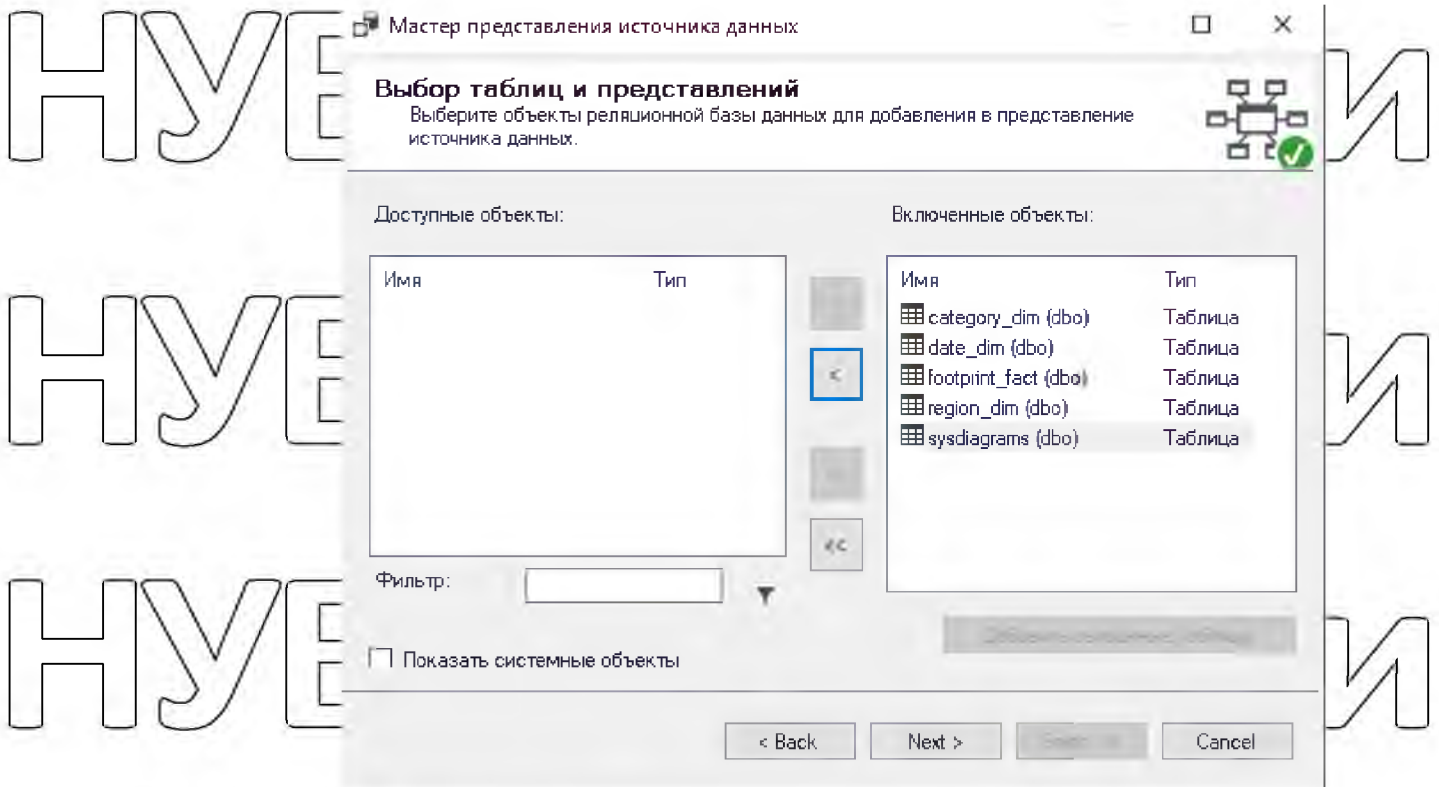


Рис. 3.5 Процесс створення представлення джерела даних

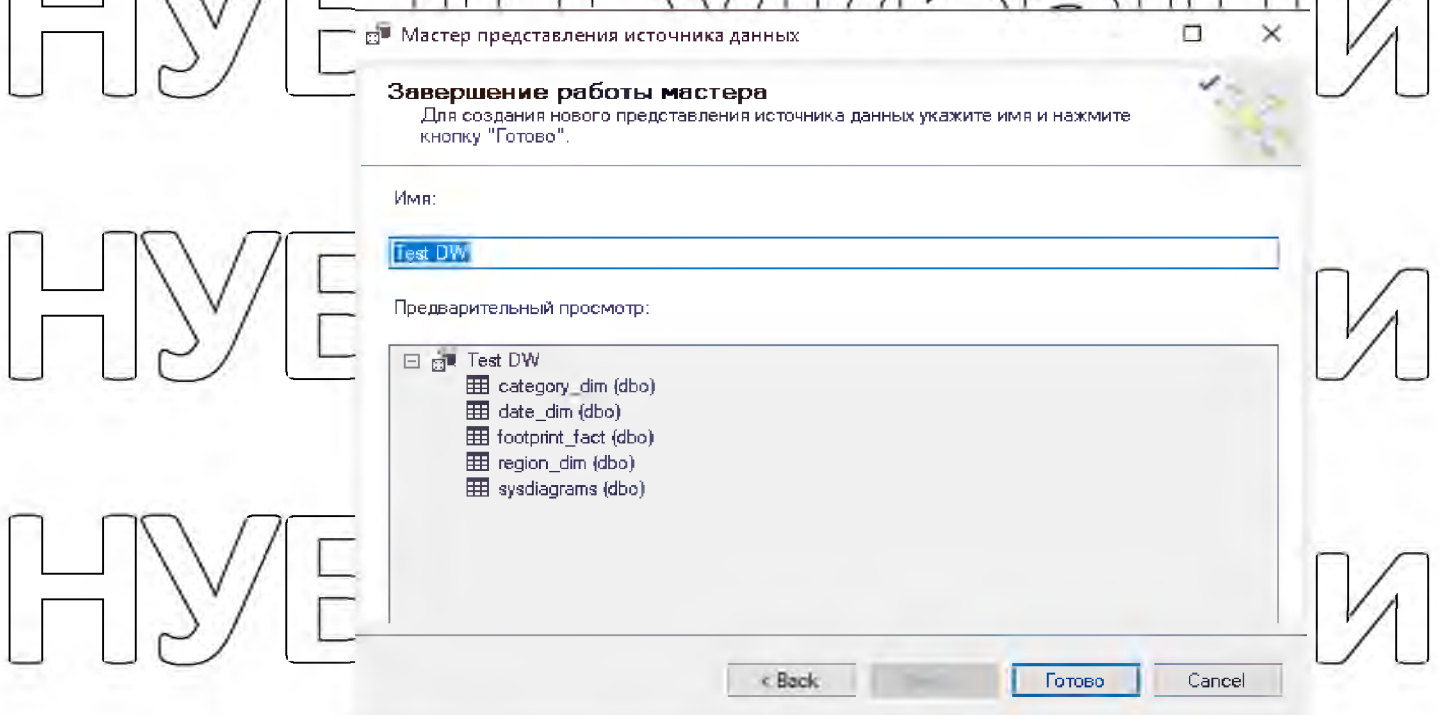


Рис. 3.6 Процесс створення представлення джерела даних

Після цього розгорнемо куб, який наведено на Рис. 3.7.

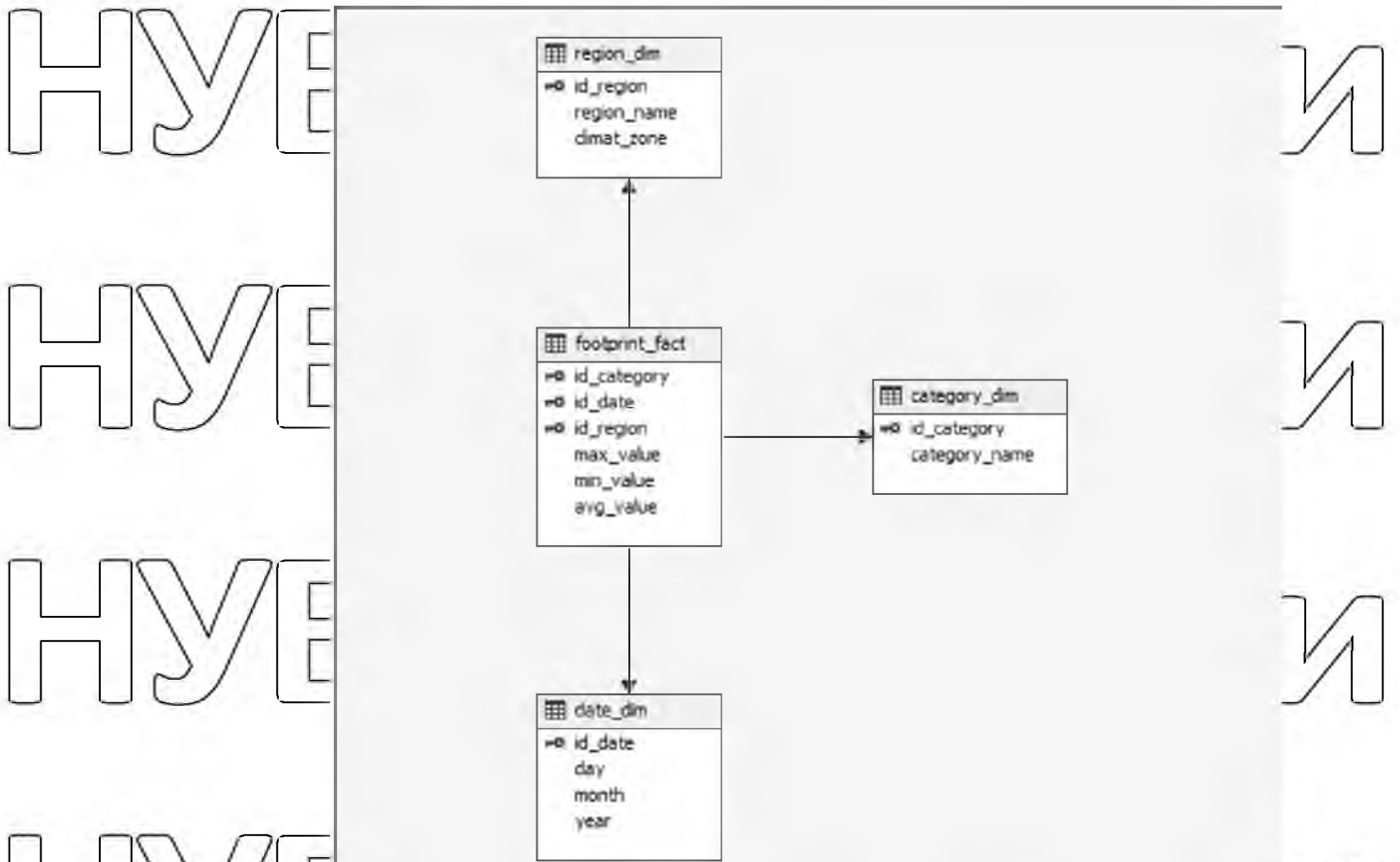


Рис.3.7 Розгорнутий куб

3.7. Реалізація отримання даних за допомогою Data Flow

Створімо проєкт за допомогою служби Business Intelligence. Додаємо елемент Data Flow Task та викликаємо редактор з'єднання та виконаємо налаштування. На Рис.3.8 наведено потік вимірів, який успішно виконаний.



Рис.3.8 Потік вимірів

Створюємо потоки даних для таблиці фактів. На Рис.3.9 наведено потік фактів, який успішно виконаний.

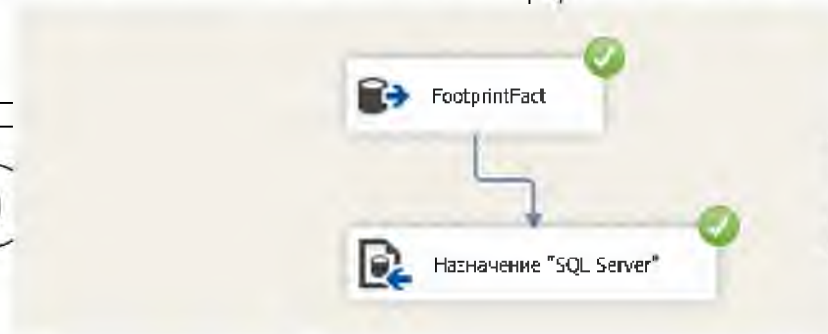


Рис.3.9 Потік фактів

Джерелом даних є заповнене сховище а приймачем даних є пуста сховище, яке наведено на Рис.3.10.

DESKTOP-79BOSTS\S...dbo.category_dim		
	id_category	category_name
▶	NULL	NULL

Рис.3.10 Таблиця сховища даних до інтеграції

Після виконання інтеграції, сховище даних заповнене, його наведено на Рис.3.11.

DESKTOP-79BOSTS\S...dbo.category_dim		
	id_category	category_name
▶	1	Харчування ...
	2	Транспорт ...
	3	Будинок ...
	4	Побутові потр...
	5	Електроенергі...
•	NULL	NULL

Рис.3.11 Таблиця сховища даних після інтеграції

4. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

4.1 Вимоги до апаратного та програмного забезпечення

Діаграма розгортання – це діаграма, яка показує конфігурацію вузлів обробки під час виконання та компонентів, які на них працюють. Діаграми розгортання – це різновид структурної діаграми, яка використовується при моделюванні фізичних аспектів об'єктно-орієнтованої системи. Вони часто використовуються для моделювання статичного вигляду розгортання системи (топології апаратного забезпечення) [27].

Діаграму розгортання системи аналізу екологічного відбитку людини представлено на Рис.4.1.

На діаграмі виділено такі вузли:

- робоча станція користувача;
- робоча станція працівника;
- веб-сервер;
- сервер бази даних.

Вузлу «Робоча станція користувача» належить компонент «Управління тестуванням».

Вузлу «Робоча станція працівника» належить компонент «Браузер».

Вузлу «Веб-сервер» належать такі компоненти:

- «Бізнес-логіка»;
- «Інтерфейс користувача»;
- «Інтерфейс для роботи з БД».

Вузлу «Сервер БД» належать такі компоненти:

- «СУБД MySQL»;
- «База даних».

Користувач взаємодіє з системою та надає власну інформацію через браузер на робочій станції.

Робоча станція працівника містить в собі програму управління тестуванням, за допомогою якої відбувається редагування інформації, редагування запитань та відповідей, а також видалення і додавання нових даних.

На веб сервері відбувається обробка даних. Сервер надає користувацький інтерфейс та використовує інтерфейс бази даних. Сервер отримує та обробляє запити.

Веб сервер взаємодіє з системою управління базою даних, та власне самою базою даних, також відправляє запити та отримує інформацію з бази даних.

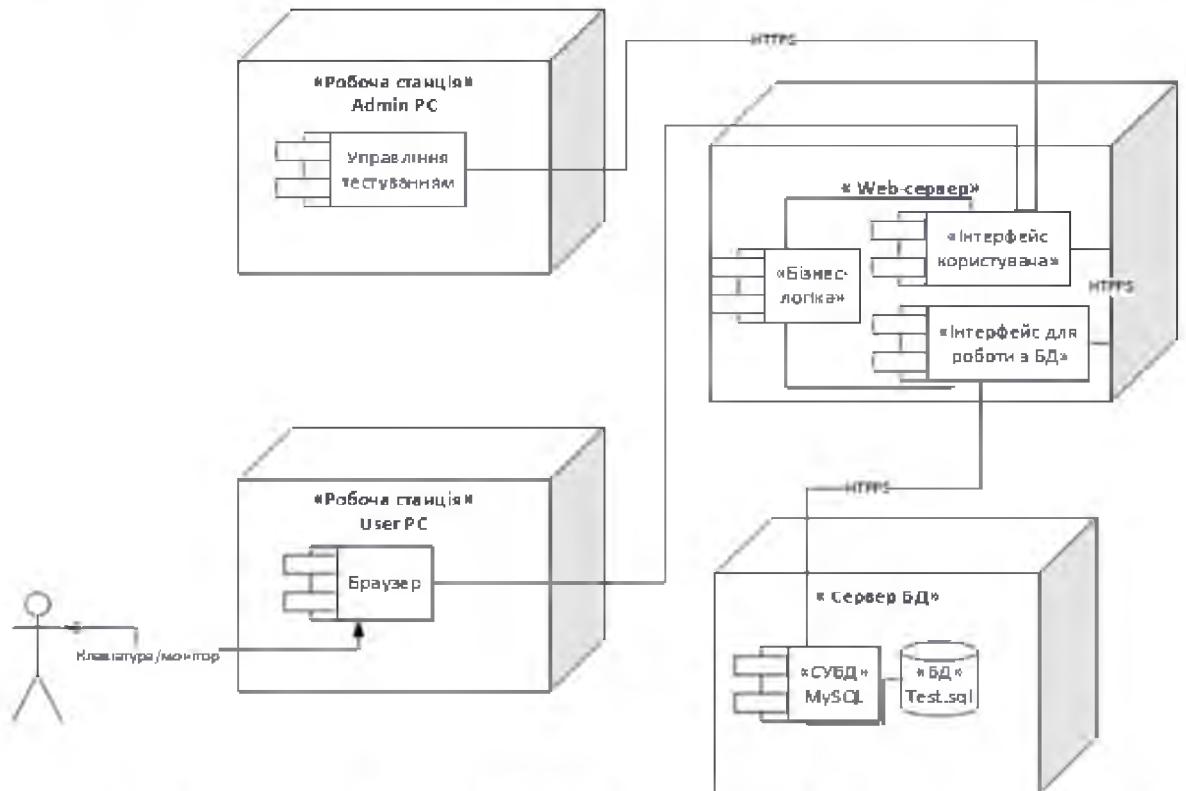


Рис. 4.1. Діаграма розгортання

4.2 Побудова звітності в середовищі BI

Як вже було зазначено раніше перед початком дослідження було сформовано перелік питань, на які повинні дати відповіді проведений аналіз, а саме:

1. В якій категорії показник екологічного відбитку найбільший?
2. Рівень споживання екологічних ресурсів в розрізі екологічного відбитку за певний період часу.
3. Середній екологічний відбиток по регіонах.

За допомогою служби Reporting створено проект та обрано раніше створене джерело даних з використанням майстра звітів. Процес вибору джерела і рядка з'єднання наведено на Рис.4.2.

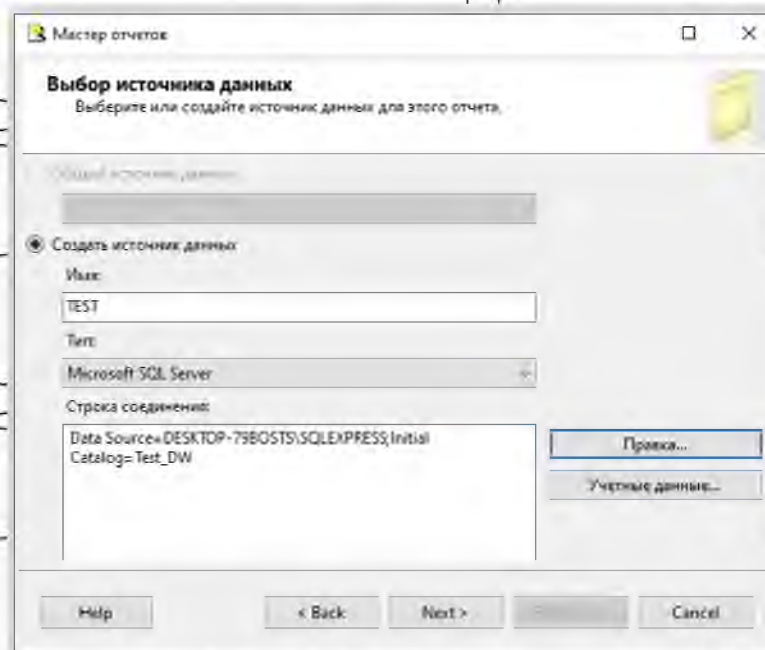


Рис.4.2 Вибір джерела даних для звіту

На наступному етапі було створено запит з використанням операторів мови SQL на створення звіту, який наведено на Рис.4.3.

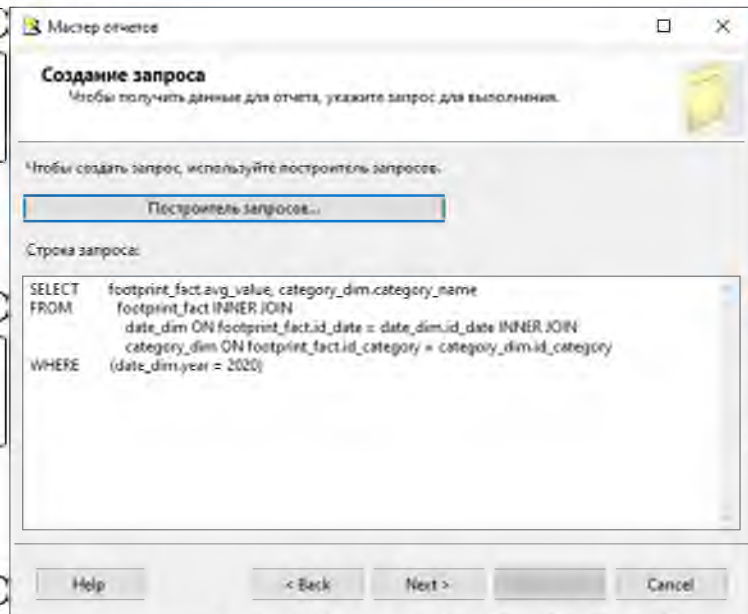


Рис.4.3 Створення запиту

У результаті виконання запиту отримано звіт, який наведено на Рис.4.4.

Даний звіт дозволяє порівняти середній екологічний відбиток по різних категоріях за 2020 рік

Середній екологічний відбиток по категоріях за 2020 рік	
Середнє значення	Категорія
4,8	Харчування
5,4	Транспорт
3,2	Будинок
6,2	Побутові потреби
4,2	Електроенергія

Рис.4.4 Звіт з середнього екологічного відбитку по категоріях за 2020 рік

На основі створеного звіту будемо кругову діаграму, що наведено на Рис.4.5, яка графічно демонструє середній екологічний відбиток по категоріях в межах 2020 року.

НУ

Середній екологічний відбиток по категоріях за 2020 рік

НУ



НУ

Рис.4.5 Кругова діаграма середнього екологічного відбитку по категоріях за 2020 рік

Аналізуючи отриманий звіт можна відмітити, що найбільше значення екологічного відбитку відповідає категорії Побутові потреби, а найменше в категорії Будинок.

Дана діаграма дозволяє зрозуміти на рекомендації з яких категорій потрібно загострити увагу, сформулювати більшу кількість рекомендацій та акцентувати увагу.

НУ

На наступному етапі формується запит на створення звіту який демонструє середні значення екологічного відбитку по регіонах, запит наведено на Рис.4.6.

НУ

НУ

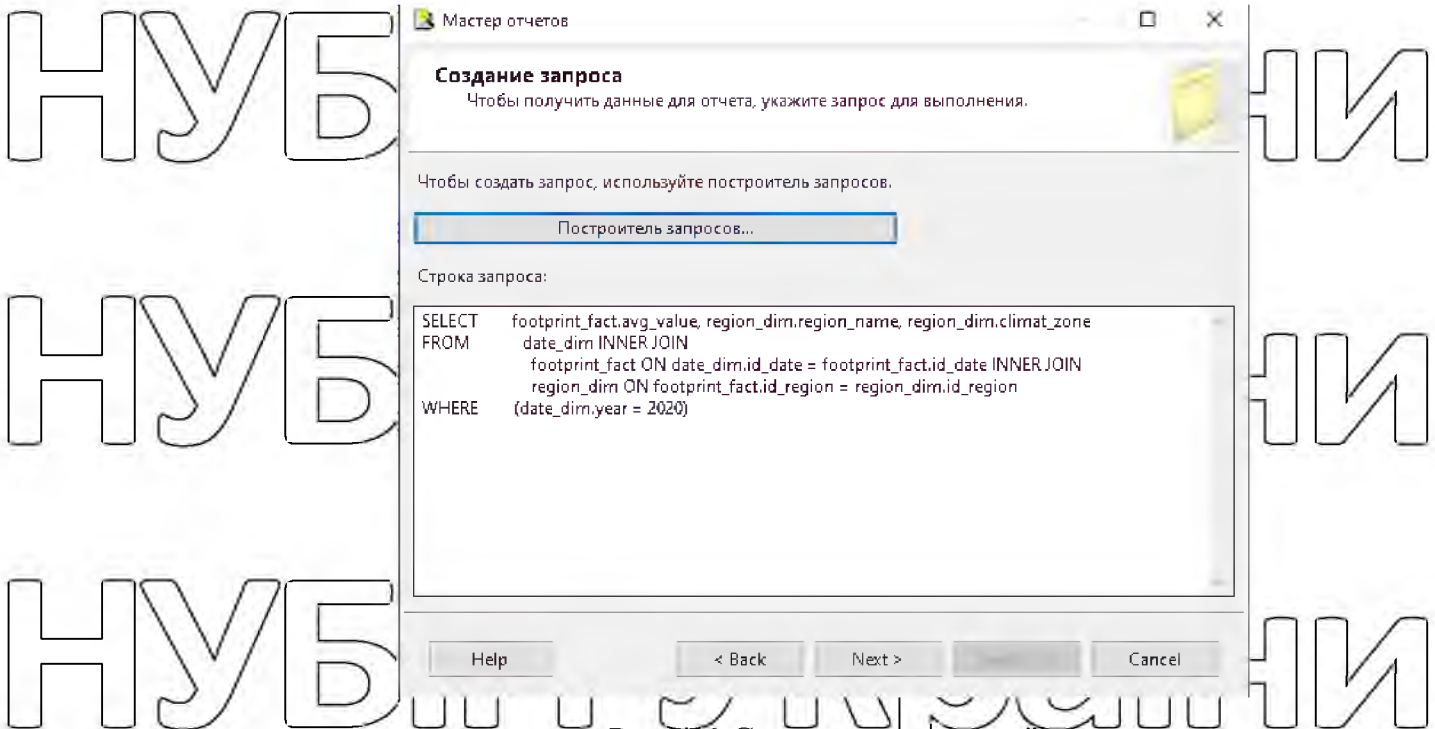


Рис.4.6 Створення запиту "

Після виконання даного запиту одержано звіт, що наведено на Рис.4.7,

який дозволяє порівняти середній екологічний відбиток в межах регіонів та кліматичної зони країни за 2020 рік

Середній екологічний відбиток по регіонах за 2020 рік

Кліматична зона	Назва регіону	Середнє значення
Лісостеп	Вінницька обл.	5,1
	Київська обл.	5,6
	Полтавська обл.	6,2
	Сумська обл.	4,2
	Тернопільська обл.	5,1
	Харківська обл.	3,9
	Хмельницька обл.	5,4
	Черкаська обл.	3,2
	Чернівецька обл.	4,8
	Полісся	Волинська обл.
Житомирська обл.		5,9
Закарпатська обл.		4,2
Львівська обл.		4,7
Рівненська обл.		4,4
Чернігівська обл.		6,9
Степ	АР Крим	5,7
	Донецька обл.	3,7
	Запорізька обл.	4,1
	Кіровоградська обл.	3,1
	Луганська обл.	4,3
	Миколаївська обл.	6,9
	Одеська обл.	3,4
	Харківська обл.	6,5

Рис.4.7 Звіт по середньому екологічному відбитку по регіонах за 2020 рік

На основі створеного звіту побудовано стовпчасту діаграму, що наведено на Рис.4.8, яка графічно демонструє середній екологічний відбиток по регіонах в межах 2020 року.



Рис.4.8 Кругова діаграма середнього екологічного відбитку по регіону за 2020 рік

Аналізуючи отриманий звіт можна відмітити, що найбільше значення екологічного відбитку в Миколаївській та Чернігівській областях, а найменше в Кіровоградській та Черкаській області.

Дані звіту дають змогу перевірити сформувані гіпотези (підтвердити або спростувати) та зробити висновки на основі графічних звітів.

4.3 Розрахунок KPI

Ключовий індикатор продуктивності застосовується для обробки інформації про задану мету, записаної в кубі фактичної формули продуктивності і результати вимірювань, які показують тренд і стан продуктивності.

KPI має чотири важливі властивості значення, ціль, статус та тренд

Значення: фактичне значення KPI. Це буде числове значення. Це може не бути включено до таблиці фактів, тому доведеться обчислити або стримати цей стовпець.

Мета: кожна організація має на меті цю цінність. Наприклад, організація може прагнути до досягнення п'ятивідсоткової маржі прибутку. Іноді вони можуть мати різні значення для різних сфер бізнесу. Наприклад, залежно від категорії продукту або території продажу цільова маржа продажів буде відрізнятися.

Статус: Залежно від значення KPI та мети KPI можна визначити статус. Наприклад, ми можемо сказати, що якщо значення KPI більше мети, це чудово, якщо воно не більше мети, але все ж таки більше нуля, це добре, а якщо менше нуля або працює зі збитками, це погано. Це "Відмінно", "Добре" або "Погано" може бути відображено користувачеві за допомогою графічного уявлення, такого як стрілка, світлофор або індикатор.

Тренд: тренд - це необов'язковий параметр при визначенні KPI, але досить важлива функція в KPI [28].

Використовуючи створений куб і службу SSAS було налаштовано обчислення KPI для кількості проходження тестування.

На Рис.4.9-4.12 відповідно наведено формули для обчислення реального значення, цільового значення, стану та тренду.

Обчислення реального значення визначається як кількість проходження тестування за 2020 рік.

Выражение значения

```
((Measures].[Число Footprint Fact],[Date Dim 2].[Year].&[2020])
```

Рис.4.9" Формула для обчислення реального значення"

Обчислення цільового значення визначається як кількість проходження тестування за 2019 рік помножено на два.

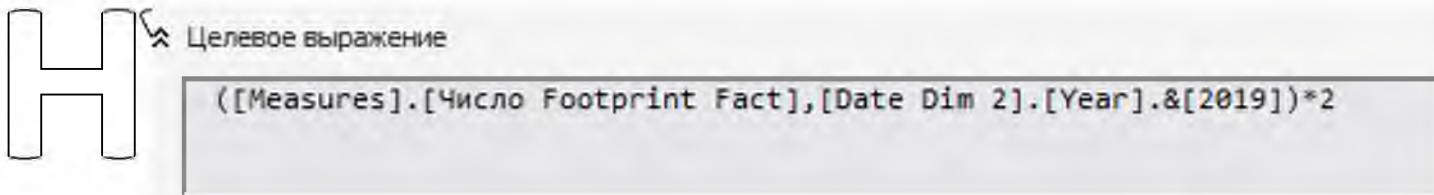


Рис.4.10" Формула для обчислення цільового значення"



Рис.4.11" Формула для обчислення стану"

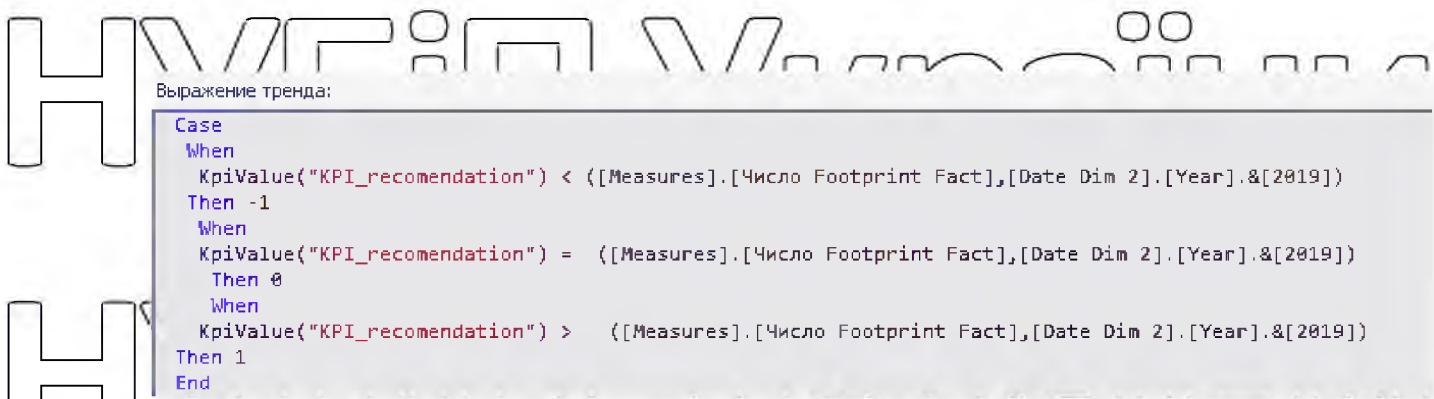


Рис.4.12" Формула для обчислення тренду"

На Рис.4.13 наведено результат обчислення КРІ, який показує, що реальне значення, перевищує цільове, це означає що ціль не тільки досягнена а і перевищено, та стрілку тренду направлено вгору, що означає поліпшення фактичного значення, щодо цільового значення.

Отобразить структуру	Значение	Цель	Состояние	Тренд	Вес
KPI_recomendation	31	24			

Рис.4.12 Результат обчислення КРІ

4.4 Пошук асоціативних правил з використанням Data Mining

У великих наборах даних досить часто можна знайти залежності, які на перший погляд не є очевидними. Для такої задачі використовується технологія Data Mining.

На основі інформації, збереженої у сховищі даних, реалізовано задачу пошуку асоціативних правил. Для цього використано створений раніше куб та середовище MS SQL Server Data Tools.

На Рис.4.13 зображено структуру моделі інтелектуального аналізу даних. Де середнє значення екологічного відбитку є прогнозом, а назва категорії вхідним значенням.

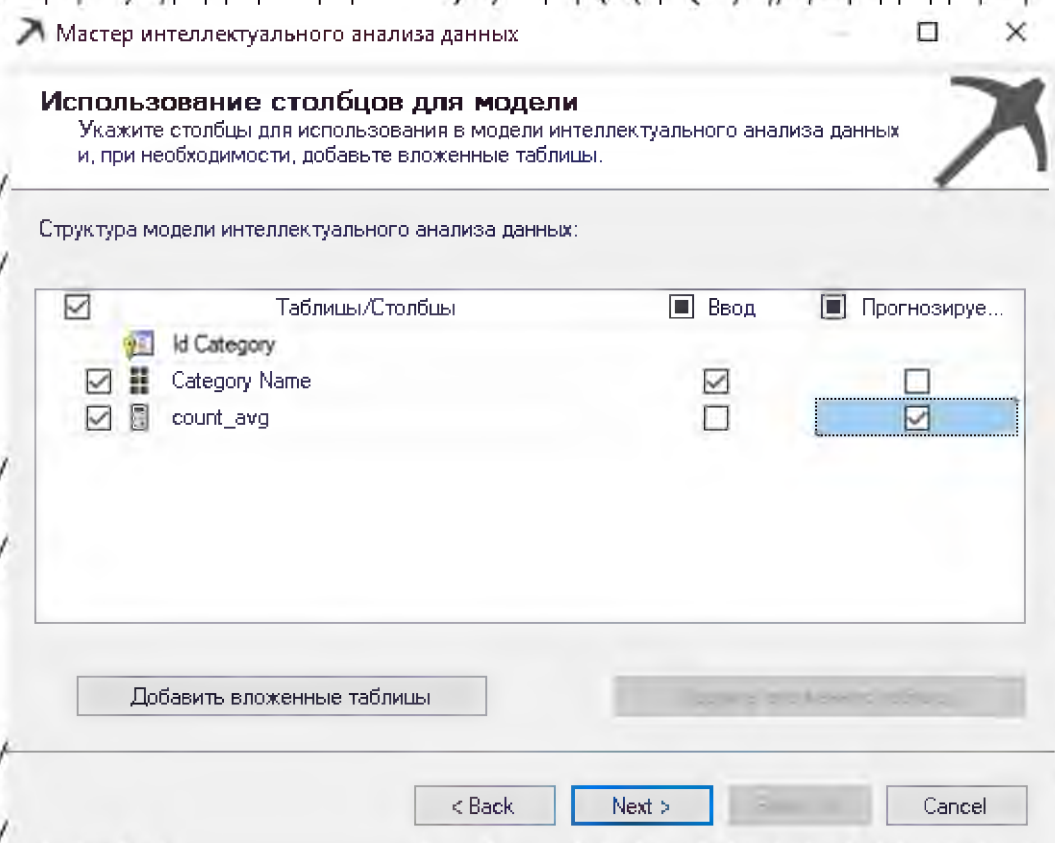


Рис.4.13 Структура моделі інтелектуального аналізу даних

Після обробки отримано результат, що наведено на Рис.4.14, які вказують наступні правила:

- В категорії Пробутові потреби та Будинки – середнє значення екологічного відбитку складає – 5,007

НУБІП України

- В категорії Харчування – 4,84-4,96
- В категорії Транспорт – 4,84
- В категорії Електроенергія – 4,96-5,007

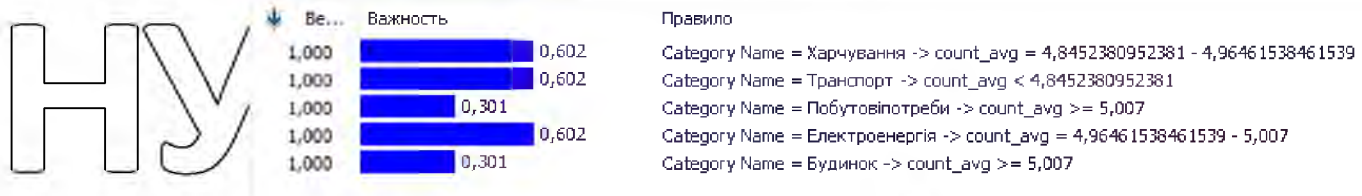


Рис.4.14 Результат пошуку асоціативних правил

Отримані результати дають змогу припустити яким буде середній екологічний відбиток в різних категоріях.

4.5 Задача класифікації з використанням Data Mining

Для рішення задачі класифікації використано IR-алгоритм. Фрагменти розробленого коду наведено у Додатку Б

Класи:

- Високий екологічний відбиток
- Низький екологічний відбиток

Залежна змінна: клас

Незалежні змінні: категорія, кліматична зона

В результаті класифікації отримано дані, що наведено на Рис.4.15-4.17.

НУБІП України

НУБІП України

Н

AVGGRADE	CategoryName	Climatzone	CLASSNAME
5.2725	Побутові потреби	Лісостеп	H
5.16	Транспорт	Полісся	H
5.09	Харчування	Полісся	H
5.715	Будинок	Степ	H
5.6425	Побутові потреби	Степ	H
5.31	Будинок	Полісся	H
6.075	Електроенергія	Полісся	H
5.134286	Харчування	Степ	H
4.325	Побутові потреби	Полісся	L
4.664	Транспорт	Степ	L
4.521	Транспорт	Лісостеп	L
4.484445	Харчування	Лісостеп	L
4.67	Будинок	Лісостеп	L
4.762727	Електроенергія	Лісостеп	L

Рис.4.15 Загальна класифікація

За результатами класифікації даних по типу кліматичної зони,

можна зробити висновок, що:

Якщо тип «Полісся», то він відноситься до класу «Високий рівень екологічного відбитку» (з ймовірністю 61%)

Якщо тип «Степ», то він відноситься до класу «Високий рівень екологічного відбитку» (з ймовірністю 66%)

Якщо тип «Лісостеп», то він відноситься до класу «Низький рівень екологічного відбитку» (з ймовірністю 69%)

Climatzone	CountMoreThanAvarage	CountLessThanAvarage	CountTotal	ClassName	Probability
Полісся	8	5	13	H	61.5384615384615
Степ	12	6	18	H	66.6666666666667
Лісостеп	18	26	44	L	59.0909090909091

Рис.4.16 Класифікація по типу кліматичної зони

Класифікація по категорії є не достатньо точною, в порівнянні з класифікацією за кліматичною зоною. Найбільш точними є припущення, що:

Якщо категорія «Харчування», то він відноситься до класу «Високий рівень екологічного відбитку» (з ймовірністю 57%)

Якщо категорія «Транспорт», то він відноситься до класу «Високий рівень екологічного відбитку» (з ймовірністю 58%)

Якщо категорія «Будинок», то він відноситься до класу «Низький рівень екологічного відбитку» (з ймовірністю 70%)

CategoryName	CountMoreThanAvarage	CountLessThanAvarage	CountTotal	ClassName	Probability
Харчування	9	12	21	L	57.1428571428571
Транспорт	7	10	17	L	58.8235294117647
Будинок	7	3	10	H	70
Побутові потреби	9	5	14	H	64.2857142857143
Електроенергія	6	7	13	L	53.8461538461538

Рис.4.17 Класифікація за категорією

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

ВИСНОВКИ

В процесі виконання даної роботи було досліджено шляхи використання сучасних OLAP-технологій в аналізі роботи системи аналізу екологічного відбитку, яка направлена на підвищення екологічної обізнаності громадян.

У першому розділі описано загальну характеристику предметної області, сформовано постановку завдання, представлено модель даних предметної області у вигляді діаграми прецедентів, описано методи дослідження використані під час роботи, а також проведено аналіз існуючих рішень та.

У другому розділі було описано моделювання системи, для цього було обрано об'єктно-орієнтований підхід. Показано взаємодію компонентів системи в часі за допомогою діаграми послідовності, побудовано діаграму діяльності для опису динамічних аспектів системи, а також діаграму пакетів для структурування елементів систем високого рівня.

У третьому розділі представлено архітектуру системи, яка представляє топологію та загальну конфігурацію програмної системи, аргументовано вибір середовища розробки, описано реалізацію клієнтської частини системи, наведено перелік інструментів, що були використані під час дослідження системи, описано вузли, які постачають дані по системі, побудовано сховище даних, а також розгорнутий куб за допомогою використання служби SSAS, та виконаю реалізацію отримання даних за допомогою Data Flow.

В останньому розділі було побудовано звіти, гістограми та розраховано ключовий показник ефективності. За допомогою отриманих звітів та гістограм надано відповіді на питання аналізу даних. Розрахунок KPI показав, що реальне значення перевищує цільове, це означає що ціль не тільки досягнуто, а і перевищено. Сформовані звіти дають змогу виявити в яких категоріях та регіонах показник найбільший. Також було вирішено задачу пошуку

асоціативних правил та задачу класифікації з використанням технологій Data Mining.

Отже, використання OLAP-технологій дало ефективні результати в процесі дослідження даної системи. Використання даних технологій дає змогу надавати своєчасну інформацію, сховища даних дають змогу оптимізувати дані для їх аналізу.

При наявності OLAP системи завжди є можливість простежити джерело інформації і визначити логічний зв'язок між отриманими результатами і вхідними даними. Знижується суб'єктивність результатів аналізу.

Використання технологій Data Mining дозволило виявити комбінації значень, що зустрічаються найчастіше та прогнозувати значення екологічного відбитку по категорії з певною ймовірністю.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

НУБІТТ України

1. Experiences in the use of Ecological Footprint as a sustainability indicator [Електронний ресурс]. – 2011. – Режим доступу до ресурсу:

https://rua.ua.es/dspace/bitstream/10045/19566/1/cuadbiod37_02.pdf.

2. Blum, O. Strategies for one-planet prosperity [Електронний ресурс] / Blum, O., Wackernagel, M. // Global Footprint Network (GFN) and Schneider Electric. – 2020. – Режим доступу до ресурсу:

<https://roscongress.org/upload/medialibrary/6ea/Earth-Overshoot-Day--Final.pdf>.

3. Ruzevicius J. Ecological footprint: Evaluation methodology and international benchmarking [Електронний ресурс] / Juozas Ruzevicius. – 2010. –

Режим доступу до ресурсу:

https://www.researchgate.net/publication/265820809_Ecological_footprint_Evaluation_methodology_and_international_benchmarking.

4. What is the definition of OLAP? [Електронний ресурс] – Режим доступу до ресурсу: <https://olap.com/olap-definition/>.

5. OLAP системи [Електронний ресурс] – Режим доступу до ресурсу:

https://www.kpms.ru/Automatization/OLAP_system.htm.

6. Characteristics of OLAP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javatpoint.com/characteristics-of-olap>.

7. Data Mining [Електронний ресурс] – Режим доступу до ресурсу:

<https://www.ibm.com/cloud/learn/data-mining>.

8. Разница между Data Mining и OLAP [Електронний ресурс] – Режим доступу до ресурсу:

<https://ru.sawakinome.com/articles/technology/difference-between-data-mining-and-olap-2.html>.

9. Research on Logistics Carbon Footprint Analysis System [Електронний ресурс] // IEEE – Режим доступу до ресурсу:

<https://ieeexplore.ieee.org/abstract/document/5687994>.

10. Кустовська О. В. Методологія системного підходу та наукових досліджень: Курс лекцій. – Тернопіль: Економічна думка, 2005. – 124 с.

11. Проектування інформаційних систем [Електронний ресурс]

Режим доступу до ресурсу: <https://uadoc.zavantag.com/text/1719/index-9.html>.

12. Уніфікована мова моделювання UML [Електронний ресурс] –

Режим доступу до ресурсу: <http://www.znannya.org/?view=uml>.

13. Застосування UML (Частина 2). Діаграма послідовності

[Електронний ресурс] – Режим доступу до ресурсу:

[http://www.dut.edu.ua/ua/news-1-626-7897-zastosuvannya-uml-chastina-2-](http://www.dut.edu.ua/ua/news-1-626-7897-zastosuvannya-uml-chastina-2-diagrama-poslidovnosti---sequence-diagram-kafedra-kompyuternih-nauk-ta-informaciynih-tehnologiy)

[diagrama-poslidovnosti---sequence-diagram kafedra-kompyuternih-nauk-ta-informaciynih-tehnologiy.](http://www.dut.edu.ua/ua/news-1-626-7897-zastosuvannya-uml-chastina-2-diagrama-poslidovnosti---sequence-diagram-kafedra-kompyuternih-nauk-ta-informaciynih-tehnologiy)

14. UML - Activity Diagrams [Електронний ресурс] – Режим доступу

до ресурсу: https://www.tutorialspoint.com/uml/uml_activity_diagram.htm.

15. What is Package Diagram [Електронний ресурс] – Режим доступу

до ресурсу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-package-diagram/>.

16. У чому різниця між архітектурою системи та архітектурою ПЗ

[Електронний ресурс] – Режим доступу до ресурсу:

[https://uk.strephonsays.com/what-is-the-difference-between-system-architecture-and-software-architecture.](https://uk.strephonsays.com/what-is-the-difference-between-system-architecture-and-software-architecture)

17. Интегрированная среда разработки Visual Studio [Електронний

ресурс] – Режим доступу до ресурсу: [https://docs.microsoft.com/ru-](https://docs.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2019)

[ru/visualstudio/get-started/visual-studio-ide?view=vs-2019.](https://docs.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2019)

18. Основные сведения об Analysis Services [Електронний ресурс] –

Режим доступу до ресурсу: [https://docs.microsoft.com/ru-ru/analysis-services/analysis-services-overview?view=asallproducts-allversions.](https://docs.microsoft.com/ru-ru/analysis-services/analysis-services-overview?view=asallproducts-allversions)

19. Службы SSIS: создание пакета ETL [Електронний ресурс] – Режим

доступу до ресурсу: [https://docs.microsoft.com/ru-ru/sql/integration-services/ssis-how-to-create-an-etl-package?view=sql-server-ver15.](https://docs.microsoft.com/ru-ru/sql/integration-services/ssis-how-to-create-an-etl-package?view=sql-server-ver15)

20. HTML basics [Електронний ресурс] – Режим доступу до ресурсу:
https://developer.mozilla.org/enUS/docs/Learn/Getting_started_with_the_web/HTML_basics?retiredLocale=uk.

21. CSS [Електронний ресурс] – Режим доступу до ресурсу:
<https://wiki.rookee.ru/css/>.

22. What is JavaScript used for? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hackreactor.com/blog/what-is-javascript-used-for>.

23. Створення сховищ даних. Технології OLAP та Data Mining [Електронний ресурс] – Режим доступу до ресурсу:
https://pidru4niki.com/16120414/informatika/stvorenniya_shovisch_daniv_tehnologiyi_olap_data_mining.

24. Введение в OLAP: часть 2. Хранилища данных [Електронний ресурс] – Режим доступу до ресурсу: http://www.olap.ru/basic/olap_intro2.asp.

25. SQL Server Analysis Services (SSAS) Tutorial [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.mssqltips.com/sqlservertutorial/2000/sql-server-analysis-services-ssas-tutorial/>.

26. Общие сведения о кубах OLAP в Service Manager для расширенной аналитики [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.microsoft.com/ru-ru/system-center/scsm/olap-cubes-overview?view=sc-sm-2019>.

27. What is Deployment Diagram? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/>.

28. Create Key Performance Indicators KPI in a SQL Server Analysis Service Cube [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.mssqltips.com/sqlservertip/6033/create-key-performance-indicators-kpi-in-a-sql-server-analysis-service-ssas-cube/>.

29. OLAP-технології та звітність [Електронний ресурс] – Режим доступу до ресурсу: https://stud.com.ua/62441/menedzhment/olap_tehnologiyi_zvitnist.

30. Барсегян, А. Методы и модели анализа данных: OLAP и Data Mining. БХВ-Петербург, 2004.

31. Горбач, Ирина Владимировна. Microsoft SQL Server 2005 Analysis Services. OLAP и многомерный анализ данных. БХВ-Петербург, 2007.

32. Кудрявцев, Ю. "OLAP-технологии: обзор решаемых задач и исследований." Бизнес-информатика 1 (2008).

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

ЗАПОВНЕННЯ СХОВИЩА ДАНИХ

```
USE [test_dw]
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'1',
N'Волнинська обл.', N'Полісся')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'2',
N'Рівненська обл.', N'Полісся')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'3',
N'Житомирська обл.', N'Полісся')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'4',
N'Чернігівська обл.', N'Полісся')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'5',
N'Львівська обл.', N'Полісся')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'6',
N'Івано-Франківська обл.', N'Полісся')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'7',
N'Закарпатська обл.', N'Полісся')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'8',
N'Тернопільська обл.', N'Лісостеп')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'9',
N'Хмельницька обл.', N'Лісостеп')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'10',
N'Вінницька обл.', N'Лісостеп')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'11',
N'Чернівецька обл.', N'Лісостеп')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'12',
N'Київська обл.', N'Лісостеп')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'13',
N'Черкаська обл.', N'Лісостеп')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'14',
N'Полтавська обл.', N'Лісостеп')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'15',
N'Сумська обл.', N'Лісостеп')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'16',
N'Харківська обл.', N'Лісостеп')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'17',
N'Кіровоградська обл.', N'Лісостеп')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'18',
N'Дніпропетровська обл.', N'Лісостеп')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'19',
N'Миколаївська обл.', N'Лісостеп')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'20',
N'Дніпропетровська обл.', N'Лісостеп')
GO
```

```
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'21',
N'Херсонська обл.', N'Степ')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'22',
N'Запорізька обл.', N'Степ')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'23',
N'Донецька обл.', N'Степ')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'24',
N'Луганська обл.', N'Степ')
GO
INSERT [dbo].[region_dim] ([id_region], [region_name], [climat_zone]) VALUES (N'25',
N'АР Крим.', N'Степ')

USE [Test_DW]
GO
INSERT [dbo].[category_dim] ([id_category], [category_name]) VALUES (N'1', N'Харчування')
GO
INSERT [dbo].[category_dim] ([id_category], [category_name]) VALUES (N'2', N'Транспорт')
GO
INSERT [dbo].[category_dim] ([id_category], [category_name]) VALUES (N'3', N'Будинок')
GO
INSERT [dbo].[category_dim] ([id_category], [category_name]) VALUES (N'4', N'Побутові
потреби')
GO
INSERT [dbo].[category_dim] ([id_category], [category_name]) VALUES (N'5',
N'Електроенергія')

USE [Test_DW]
GO
INSERT [dbo].[date_dim] ([id_date], [day], [month], [year]) VALUES (N'1', N'12', N'10',
N'2019')
GO
INSERT [dbo].[date_dim] ([id_date], [day], [month], [year]) VALUES (N'2', N'03', N'11',
N'2019')
GO
INSERT [dbo].[date_dim] ([id_date], [day], [month], [year]) VALUES (N'3', N'11', N'10',
N'2019')
GO
INSERT [dbo].[date_dim] ([id_date], [day], [month], [year]) VALUES (N'4', N'15', N'01',
N'2019')
GO
INSERT [dbo].[date_dim] ([id_date], [day], [month], [year]) VALUES (N'5', N'26', N'02',
N'2019')
GO
INSERT [dbo].[date_dim] ([id_date], [day], [month], [year]) VALUES (N'6', N'12', N'04',
N'2019')
GO
INSERT [dbo].[date_dim] ([id_date], [day], [month], [year]) VALUES (N'7', N'18', N'09',
N'2020')
GO
INSERT [dbo].[date_dim] ([id_date], [day], [month], [year]) VALUES (N'8', N'10', N'10',
N'2020')
GO
INSERT [dbo].[date_dim] ([id_date], [day], [month], [year]) VALUES (N'9', N'10', N'12',
N'2020')
GO
INSERT [dbo].[date_dim] ([id_date], [day], [month], [year]) VALUES (N'10', N'15', N'04',
N'2020')
GO
```

```
INSERT [dbo].[date_dim] ([id_date], [day], [month], [year]) VALUES (N'11', N'11', N'05',
N'2020')
GO
INSERT [dbo].[date_dim] ([id_date], [day], [month], [year]) VALUES (N'12', N'05', N'05',
N'2020')

USE [Test_DW]
GO
INSERT [dbo].[footprint_fact] ([id_category],[id_date],[id_region], [max_value],
[min_value], [avg_value]) VALUES (N'1', N'2', N'10', N'7.2', N'2.4', N'5.1')
GO
INSERT [dbo].[footprint_fact] ([id_category],[id_date],[id_region], [max_value],
[min_value], [avg_value]) VALUES (N'2', N'8', N'17', N'8.2', N'2.8', N'5.4')
GO
INSERT [dbo].[footprint_fact] ([id_category],[id_date],[id_region], [max_value],
[min_value], [avg_value]) VALUES (N'3', N'3', N'20', N'6.8', N'3.4', N'5.5')
GO
INSERT [dbo].[footprint_fact] ([id_category],[id_date],[id_region], [max_value],
[min_value], [avg_value]) VALUES (N'4', N'6', N'12', N'8.1', N'4.4', N'6.2')
GO
INSERT [dbo].[footprint_fact] ([id_category],[id_date],[id_region], [max_value],
[min_value], [avg_value]) VALUES (N'5', N'1', N'16', N'4.2', N'5.2', N'4.2')
GO
INSERT [dbo].[footprint_fact] ([id_category],[id_date],[id_region], [max_value],
[min_value], [avg_value]) VALUES (N'1', N'10', N'18', N'3.2', N'3.0', N'3.2')
GO
INSERT [dbo].[footprint_fact] ([id_category],[id_date],[id_region], [max_value],
[min_value], [avg_value]) VALUES (N'2', N'7', N'13', N'5.4', N'2.6', N'4.8')
GO
INSERT [dbo].[footprint_fact] ([id_category],[id_date],[id_region], [max_value],
[min_value], [avg_value]) VALUES (N'3', N'3', N'24', N'7.9', N'2.4', N'5.5')
```

КОД ПРОГРАМИ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ
КЛАСИФІКАЦІЇ

Загальна класифікація:

```

public static IEnumerable<MainClasses> MainClasses()
{
    var sqlQuery = @"declare @totalAvg float

                    set @totalAvg = (SELECT AVG(AVG_VALUE) FROM Footprint_fact
                    );
                    WITH MAIN_CLASSES as
                    (
                    SELECT  category_dim.category_name as CategoryName,
region_dim.climat_zone as Climatzone, AVG(AVG_VALUE) as AVG_GRADE
                    FROM    Footprint_fact
                    LEFT JOIN Category_dim ON Category_dim.id_category = Footprint_fact.id_category
                    LEFT JOIN Region_dim ON Region_dim.id_region =
Footprint_fact.id_region

                    GROUP BY category_dim.category_name, region_dim.climat_zone
                    )
                    select *, case
                        when AVG_GRADE >= @totalAvg then 'H'
                        ELSE 'L'
                        END AS CLASS_NAME
                    from MAIN_CLASSES
                    Order BY CLASS_NAME";
    List<MainClasses> mainClasses = new List<MainClasses>();
    using (IDbConnection db = new SqlConnection(connectionString))
    {
        mainClasses = db.Query<MainClasses>(sqlQuery).ToList();
    }
    return mainClasses;
}
}

```

Класифікація по кліматичній зоні:

```

private List<ClasificationByClimatzone> getDataClimatzone()
{
    var allRecord = FactDataRepository.Retrieve();
    var dict = new Dictionary<string, ClasificationByClimatzone>();
    foreach(var rec in allRecord)
    {
        ClasificationByClimatzone Climatzone;
        if (dict.TryGetValue(rec.Climatzone, out Climatzone))
        {
            Climatzone.CountTotal++;
            if(rec.Grade >= FactDataRepository.TotalResourcesAvgGrade)
            {
                Climatzone.CountMoreThanAvarage++;
            }
            else
            {
                Climatzone.CountLessThanAvarage++;
            }
        }
        else
        {
            dict[rec.Climatzone] = new ClasificationByClimatzone()
            {
                ClimatzoneName = rec.Climatzone
            };
            dict[rec.Climatzone].CountTotal++;
        }
    }
}

```



```

if (rec.Grade >= FactDataRepository.TotalResourcesAvgGrade)
    {
        dict[rec.Climatzone].CountMoreThanAvarage++;
    }
else
    {
        dict[rec.Climatzone].CountLessThanAvarage++;
    }
}

foreach(var resType in dict.Values)
{
    if ((resType.CountMoreThanAvarage == 0 && resType.CountLessThanAvarage ==
0) || resType.CountTotal == 0)
    {
        resType.ClassName = "L";
        resType.Probability = 0;
    }
else if (resType.CountMoreThanAvarage > resType.CountLessThanAvarage)
    {
        resType.ClassName = "H";
        resType.Probability = (double)resType.CountMoreThanAvarage /
(double)resType.CountTotal * 100;
    }
else
    {
        resType.ClassName = "L";
        resType.Probability = (double)resType.CountLessThanAvarage /
(double)resType.CountTotal * 100;
    }
}
return dict.Values.ToList();
}

```

Класифікація по категорії:

```

private List<ClasificationByCourse> getDataByCategoryName()
{
    var allRecord = FactDataRepository.Retrieve();
    var dict = new Dictionary<string, ClasificationByCourse>();
    foreach (var rec in allRecord)
    {
        ClasificationByCourse course;
        if (dict.TryGetValue(rec.CategoryName, out course))
        {
            course.CountTotal++;
            if (rec.Grade >= FactDataRepository.TotalResourcesAvgGrade)
            {
                course.CountMoreThanAvarage++;
            }
            else
            {
                course.CountLessThanAvarage++;
            }
        }
        else
        {
            dict[rec.CategoryName] = new ClasificationByCourse()
            {
                CategoryName = rec.CategoryName
            }
        }
    }
}

```

```
};

    dict[rec.CategoryName].CountTotal++;
    if (rec.Grade >= FactDataRepository.TotalResourcesAvgGrade)
    {
        dict[rec.CategoryName].CountMoreThanAvarage++;
    }
    else
    {
        dict[rec.CategoryName].CountLessThanAvarage++;
    }
}

foreach (var resType in dict.Values)
{
    if ((resType.CountMoreThanAvarage == 0 && resType.CountLessThanAvarage ==
0) || resType.CountTotal == 0)
    {
        resType.ClassName = "L";
        resType.Probability = 0;
    }
    else if (resType.CountMoreThanAvarage > resType.CountLessThanAvarage)
    {
        resType.ClassName = "H";
        resType.Probability = (double)resType.CountMoreThanAvarage /
(double)resType.CountTotal * 100;
    }
    else
    {
        resType.ClassName = "L";
        resType.Probability = (double)resType.CountLessThanAvarage /
(double)resType.CountTotal * 100;
    }
}
return dict.Values.ToList();
}
```