

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

УДК 613.4:004.896:004.9

«ПОГОДЖЕНО»

«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»

Декан факультету
інформаційних технологій

Завідувач кафедри комп'ютерних наук

Глазунова О.Г., д.п.н., професор

Голуб Б.Л., к.т.н., доцент

_____ 2023 р.

_____ 2023 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему Розробка комп'ютерної інтегрованої системи керування процесу
вирощування рослин в гідропоніці

Спеціальність _____ 122, Комп'ютерні науки

(код і назва)

Освітня програма _____ Інформаційні управляючі системи та технології

(назва)

Орієнтація освітньої програми освітньо – професійна
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

кандидат економічних наук, доцент _____
(науковий ступінь та вчене звання) (підпис)

Густера Олег Михайлович
(ПІБ)

Керівник магістерської кваліфікаційної роботи

кандидат економічних наук, доцент _____
(науковий ступінь та вчене звання) (підпис)

Густера Олег Михайлович
(ПІБ)

Виконав

_____ (підпис)

Мороз Олександр Олександрович
(ПІБ студента)

КИЇВ-2023

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (ННІ) Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри
комп'ютерних наук

К.Т.Н., доцент _____ Голуб Б.Л.
(науковий ступінь, вчене звання) (підпис) (ПІБ)
“ _____ ” _____ 2023 року

З А В Д А Н Н Я

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Морозу Олександр Олександровичу

(прізвище, ім'я, по батькові)

Спеціальність 122 «Комп'ютерні науки»

(код і назва)

Освітня програма «Інформаційні управляючі системи та технології»

(назва)

Орієнтація освітньої програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Тема магістерської кваліфікаційної роботи: Розробка комп'ютерної інтегрованої системи керування процесу вирощування рослин в гідропоніці

затверджена наказом ректора НУБіП України від «29» серпня 2023р. №1384С

Термін подання завершеної роботи на кафедру 2 листопада 2023 року

(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи: для дослідження були використані дані з власної гідропоніки.

Перелік питань, що підлягають дослідженню:

1. Аналіз та моделювання предметної області.
2. Проектування системи.
3. Дослідження можливостей використання обраних технологій для вирішення аналітичних задач обліку складських технологій.

Перелік графічного матеріалу (за потреби) _____

Дата видачі завдання «01» квітня 2023 р.

Керівник магістерської кваліфікаційної роботи _____

(підпис)

Густера О.М.
(прізвище та ініціали)

Завдання прийняв до виконання _____

(підпис)

Мороз О.О.
(прізвище та ініціали студента)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	5
ВСТУП	6
1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1	10
1.1.1 Аналіз поточних методів керування в гідропоніці	10
1.1.2 Вплив факторів на вирощування рослин в гідропоніці	11
1.1.3 Важливість збору та зберігання даних в системах гідропоніки	12
1.1.4 Методи Data Mining для аналізу даних з гідропонічних систем	13
1.1.5 Застосування результатів аналізу для оптимізації керування процесом вирощування рослин	14
1.2	15
1.2.1 Аналіз існуючих систем гідропоніки та їхніх переваг і обмежень	15
1.2.2 Проблеми та можливості у гідропоніці та інновації в цій галузі	16
1.3	17
2 МОДЕЛЮВАННЯ СИСТЕМИ	19
2.1 Опис та аналіз методології системного аналізу	19
2.2 Загальні відомості про моделювання	20
2.2.1 Визначення UML	20
2.2.2 Об'єктно-орієнтоване проектування	21
2.3 Діаграма прецедентів	22
2.4 Діаграма послідовності	25

	4
2.5 Діаграма розгортання і компонентів	28
3 РОЗРОБКА СИСТЕМИ	31
3.1 Аналіз джерел інформації	31
3.1.1 Структура оперативної бази даних	31
3.1.2 Фундаментальні принципи OLAP-технологій	33
3.1.3 Організація сховища даних	36
3.1.4 Методи вилучення, обробки і передачі даних	38
3.2 Основи Data Mining	49
3.3 Аналіз інструментарію для виконання завдань Data Mining	49
3.4 Дані для проведення аналізу	50
4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ	52
4.1 Дослідження використання задач класифікації	52
4.1.1 Використання 1-Rule для класифікації	52
4.2 Дослідження використання методу асоціативних правил	55
4.3 Дослідження використання алгоритмів кластеризації	59
ВИСНОВКИ	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	67
ДОДАТОК А	69
ДОДАТОК Б	73

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

DM – Data Mining.

БД – база даних.

СД – сховище даних.

СУБД – система управління базами даних.

SSAS – SQL Server Integration Services.

SSIS – SQL Server Integration Services.

SSRS – SQL Server Reporting Services.

OLAP – On-Line Analytical Processing

DSV - Data Source View.

DF – Data Flow.

ВСТУП

Введення в гідропоніку та її актуальність

Гідропоніка - це сучасний та інноваційний підхід до вирощування рослин, який відкриває нові можливості для сільського господарства і сільськогосподарських культур. Вона основана на принципі культивування рослин без використання традиційного ґрунту, замість нього коренева система рослин знаходиться в спеціальних субстратах або підтримується у водному середовищі. У цьому методі всі необхідні поживні речовини подаються рослинам через воду, яка збагачена добривами та іншими необхідними елементами[1].

Актуальність гідропоніки надзвичайно важлива в сучасному світі, оскільки ми стикаємося зі зростаючим населенням і дефіцитом земельних ресурсів. Земельні ділянки, доступні для сільського господарства, стають обмеженими, і традиційні методи вирощування рослин не завжди можуть забезпечити потреби в продуктах харчування. Гідропоніка може бути відповіддю на цей виклик, допомагаючи підвищити продуктивність рослин і зменшити використання обмежених земельних площ та інших ресурсів.

Запропонований підхід до вирощування рослин у гідропоніці також відкриває нові можливості для використання високобудівельних структур у сільському господарстві. Високі будівлі можуть стати ідеальними місцями для вирощування рослин, забезпечуючи високу врожайність при обмеженому земельному просторі. Такий підхід має потенціал забезпечити продуктами харчування міські області та зменшити відстані транспортування продуктів.

Виклики сучасного господарства та гідропоніка як рішення

Сьогодні сільське господарство стає перед багатьма викликами, серед яких основними є зростаюча потреба в продуктах харчування та обмеженість доступного земельного простору. Збільшення населення планети, швидке міське зростання та зміни клімату роблять сільське господарство більш складним завданням, яке потребує інноваційних підходів[2].

Гідропоніка відкриває нові можливості для покращення продуктивності сільського господарства та розвитку стійкого способу вирощування рослин. Один із головних переваг цього методу полягає в можливості культивувати рослини в будь-якому регіоні, навіть у важкодоступних місцях, де земля не підходить для традиційного сільського господарства.

Гідропоніка дозволяє значно зменшити споживання води порівняно з традиційними методами, оскільки вода використовується в цій системі більш ефективно і рециркулюється. Це особливо важливо в умовах загостреного водного дефіциту в деяких регіонах світу.

Гідропоніка також може бути більш ефективною відповіддю на зростаючу потребу в продуктах харчування у міських областях. Високі будівлі можуть стати місцями для вирощування продуктів харчування, забезпечуючи містянам доступ до свіжих продуктів, які вирощуються локально, без необхідності далекого транспортування. Це може покращити якість харчування та зменшити вплив транспорту на навколишнє середовище.

Гідропоніка також надає можливість боротьби зі шкідниками та хворобами більш ефективно. Вирощування рослин в контрольованих умовах сприяє зменшенню ризику зараження, що може зменшити необхідність у хімічних обробках.

Роль рН в гідропоніці та її вплив на врожайність

Контроль над рівнем кислотності (рН) грає важливу роль у гідропоніці та має значний вплив на врожайність та якість рослин. Рослини в гідропонічних системах набувають поживні речовини через воду, і їх коренева система дуже чутлива до змін рН у водному середовищі[3].

Важливою особливістю гідропоніки є можливість точного і автоматизованого контролю рівня рН у воді. За допомогою датчиків та систем автоматизації, можна підтримувати стає значення рН, оптимальне для кожного виду рослин, що вирощується. Це дозволяє забезпечити рослинам найкращі умови для зростання та розвитку.

Дослідження та розробка інтегрованих систем керування процесом вирощування рослин у гідропоніці, які включають в себе контроль та регулювання рН, є важливим кроком у забезпеченні стабільної та високоякісної продукції. Вона допомагає забезпечити оптимальні умови для рослин, що в свою чергу призводить до підвищення врожайності та зниження втрат через фізіологічні стреси, пов'язані з коливаннями рівня рН.

Майбутні перспективи гідропоніки

Гідропоніка відкриває перед собою широкі можливості для майбутнього сільського господарства. За роки свого розвитку вона довела свою ефективність у забезпеченні продуктами харчування та оптимізації вирощування рослин[4].

Однією з головних переваг гідропоніки є здатність вирощувати рослини в будь-якому кліматі і навіть у важкодоступних місцях, де земельні ресурси обмежені. Це може виявитися особливо важливим у майбутньому, коли земельні ділянки для традиційного сільського господарства можуть стати обмеженими.

Гідропоніка також допомагає зменшити споживання води порівняно з традиційними методами вирощування, завдяки більш ефективному використанню води і її рециркуляції в системі. Це робить її більш стійкою до водних дефіцитів, які можуть бути актуальними проблемами у майбутньому.

У міських областях гідропоніка може забезпечити можливість вирощування свіжих продуктів харчування на високих будівлях або у спеціально обладнаних контейнерах. Це дозволить містянам мати доступ до локальної та свіжої їжі без довгих транспортних маршрутів[5].

Загалом, гідропоніка має потенціал стати ключовим компонентом майбутнього сільського господарства та допомогти вирішувати проблеми, пов'язані з обмеженістю ресурсів і зростанням населення.

У структурі пояснювальної записки визначено перелік умовних позначень, вступ, і чотири розділи, складаючи загальний обсяг роботи на 66 сторінках і включаючи 55 рисунків. В інтродукції розглянуті актуальність, проблематика,

предмет, об'єкт, мета, завдання, методи дослідження та проведена апробація роботи.

Перший розділ присвячений предметній області, в якому розглядаються інформаційні джерела, оцінюються наявні рішення і формулюються основні завдання майбутньої роботи. Другий розділ охоплює теоретичну складову, зокрема, моделювання системи та використані технології. Тут також наведені діаграми, що описують розроблену інформаційно-аналітичну систему та розглянуто вузли, з яких здійснюється збір інформації. У третьому розділі надано огляд розробки системи, включаючи створення сховища даних, підготовку та обробку інформації, а також заповнення сховища даних. В четвертому розділі здійснюється дослідження кількох методів Data Mining для визначення їхньої ролі та важливості в контексті використання. У висновках наведено короткі підсумки кожного розділу, висвітлено результати дослідження та виконані завдання, і надані рекомендації.

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Data Mining для аналітики систем керування процесу вирощування рослин в гідропоніці

1.1.1 Аналіз поточних методів керування в гідропоніці. В гідропоніці, як і в будь-якій іншій галузі сільського господарства, ефективне керування процесом вирощування рослин є важливою складовою для досягнення високої врожайності та якості продукції. Аналіз поточних методів керування в гідропоніці відіграє ключову роль у визначенні ефективності та можливих покращень у цій сфері.

Для розпочатку аналізу поточних методів керування в гідропоніці слід розглянути різні аспекти, такі як системи живлення та поливу, контроль за водою та добривами, а також методи моніторингу та регулювання параметрів середовища для рослинного росту. Різні гідропонічні системи можуть використовувати різні підходи та технології в цих аспектах[6].

Аналіз також може включати огляд різних типів сенсорів та інструментів, що використовуються для моніторингу рівня рН, рівня добрив, температури та інших параметрів. Важливо визначити, які з цих параметрів мають найбільший вплив на рослинний ріст та як їхня динаміка впливає на врожайність.

Аналіз поточних методів також допоможе ідентифікувати можливі обмеження та проблеми, з якими стикаються гідропонічні системи. Це може включати питання стабільності рівня рН, ефективності систем поливу та добрив, а також оптимальних параметрів середовища для конкретних видів рослин[7].

Аналіз поточних методів керування в гідропоніці є важливим кроком у підготовці до впровадження інтегрованої системи керування та аналізу даних для оптимізації процесу вирощування рослин.

1.1.2 Вплив факторів на вирощування рослин в гідропоніці. Вирощування рослин в системах гідропоніки без використання ґрунту або землі є характерною рисою сучасного сільського господарства. У таких системах рослини ростуть у водних розчинах або на носіях, які не містять ґрунту, але забезпечують рослинам доступ до необхідних поживних речовин. Вирощування рослин у гідропоніці без ґрунту зумовлює вплив різних факторів на їхній ріст і розвиток[8].

Основні фактори, які впливають на вирощування рослин в гідропоніці без ґрунту, включають:

- Фактори середовища:
 1. Освітлення: Якість і тривалість освітлення мають великий вплив на фотосинтез та ріст рослин у гідропонічних системах.
 2. Температура: Регулювання температури важливе для створення оптимальних умов для росту рослин.
 3. Вологість повітря: Відносна вологість повітря повинна бути контрольованою, оскільки рослини можуть впливати на рівень вологості.
- Склад харчових розчинів: Гідропонічні системи без ґрунту постачають рослини поживними речовинами через воду. Оптимальний склад і концентрація харчових розчинів є ключовими для надання рослинам необхідних поживних речовин.
- Рівень рН: Контроль рівня рН води в гідропонічній системі без ґрунту допомагає узгодити доступність поживних речовин для рослин.
- Повітряна обмін: Рослини в гідропоніці без ґрунту також потребують доступу до кисню для кореневої респірації. Забезпечення належного повітряного обміну у системі є важливим аспектом для здоров'я рослин.

Розуміння впливу цих факторів на вирощування рослин у гідропонічних системах без ґрунту допомагає визначити оптимальні умови для їхнього росту та розвитку. На основі аналізу цих факторів можна розробити системи керування

та моніторингу, що покращують врожайність та якість продукції в гідропоніці[9].

1.1.3 Важливість збору та зберігання даних в системах гідропоніки. Збір та зберігання даних в системах гідропоніки є фундаментальним аспектом для досягнення ефективного та високоврожайного вирощування рослин[10]. У контексті гідропонічних систем ця важлива функція набуває ще більшої значущості через відсутність традиційного ґрунту та необхідність точного контролю середовища, в якому ростуть рослини[11].

Важливі аспекти збору та зберігання даних у системах гідропоніки включають:

- Моніторинг параметрів рослинного середовища: Це включає вимірювання та запис параметрів, таких як рівень рН води, електрична провідність розчинів, температура води та повітря, вологість ґрунту (або носія), та рівень CO₂. Зібрані дані надають інформацію про стан середовища, в якому ростуть рослини, і дозволяють вживати відповідних заходів для підтримки оптимальних умов.
- Спостереження за ростом рослин: Фіксування даних про ріст, розвиток та стан рослин є важливим етапом для визначення їхньої продуктивності та добробуту. Це включає в себе вимірювання висоти, кількості листя, структури кореневої системи та інших параметрів рослин.
- Моніторинг рівнів харчових розчинів: Склад та концентрація харчових розчинів, які постачаються до рослин, слід ретельно контролювати. Вимірювання рівнів макро- та мікроелементів, азоту, фосфору та калію допомагає забезпечити рослини необхідними поживними речовинами.
- Збір даних про врожайність та якість продукції: Важливо вимірювати та реєструвати врожайність, якість та кількість зібраної продукції. Ця інформація дозволяє оцінювати результати та ефективність системи гідропоніки та робити відповідні корективи.

Збір та зберігання даних в системах гідропоніки допомагають сільськогосподарським виробникам впроваджувати системи керування, які забезпечують оптимальні умови для росту рослин та максимізують врожайність. Дані також можуть використовуватись для аналізу та прогнозування, що сприяє покращенню сільського господарства в гідропоніці.

1.1.4 Методи Data Mining для аналізу даних з гідропонічних систем. Аналіз даних з гідропонічних систем вимагає використання спеціалізованих методів обробки та аналізу інформації, включаючи методи Data Mining. Data Mining, або видобуток знань із даних, є потужним інструментом для виділення прихованих зв'язків, закономірностей та важливої інформації із великих обсягів даних.

Нижче представлено деякі основні методи Data Mining, які можна застосовувати для аналізу даних з гідропонічних систем:

- **Кластерний аналіз (Cluster Analysis):** Цей метод дозволяє групувати рослини за спільними ознаками, такими як ріст, врожайність, реакція на певні параметри середовища тощо. Кластерний аналіз допомагає визначити схожі групи рослин та вивчити їхні особливості.
- **Кореляційний аналіз (Correlation Analysis):** Цей метод дозволяє визначити статистичні зв'язки між різними параметрами, наприклад, між рівнем рН та врожайністю. Відомі кореляції можуть служити основою для прогнозування та оптимізації умов вирощування рослин.
- **Класифікація та прогнозування (Classification and Prediction):** Ці методи дозволяють побудувати моделі, які передбачають ріст та врожайність рослин на основі історичних даних. Моделі класифікації та прогнозування можуть бути використані для прийняття рішень щодо оптимальних параметрів управління гідропонічною системою.
- **Асоціаційний аналіз (Association Analysis):** Цей метод дозволяє виявити асоціації між різними факторами та подіями. В контексті гідропоніки, це

може включати виявлення залежностей між змінами параметрів рослинного середовища та їхнім впливом на врожайність.

- Візуалізація даних (Data Visualization): Візуалізація даних допомагає представити інформацію у вигляді графіків, діаграм, карт та інших візуальних засобів. Це допомагає зрозуміти складні взаємозв'язки між різними параметрами.

Методи Data Mining в гідропоніці дозволяють зробити інформовані рішення щодо керування системами, покращити вирощування рослин та максимізувати врожайність. Вони допомагають виявляти та аналізувати складні взаємозв'язки між параметрами та оптимізувати процес гідропонічного вирощування рослин.

1.1.5 Застосування результатів аналізу для оптимізації керування процесом вирощування рослин. Результати аналізу даних з гідропонічних систем можуть бути використані для розробки та впровадження оптимізованих стратегій керування процесом вирощування рослин[14]. Застосування знань, отриманих з аналізу, дозволяє досягти наступних важливих цілей:

- Удосконалення ростових умов: На основі даних можна визначити оптимальні параметри для вирощування рослин, такі як рівень рН, концентрація харчових розчинів, температура та освітлення. Це допомагає підтримувати найкращі умови для росту та розвитку рослин.
- Ефективне використання ресурсів: Аналіз даних допомагає визначити, скільки води, добрив та енергії потрібно для вирощування рослин. Це дозволяє зменшити витрати ресурсів та збільшити ефективність системи.
- Прогнозування врожаю: Враховуючи історичні дані та залежності між параметрами, можна розробити моделі для прогнозування врожаю. Це допомагає планувати виробництво та запобігати можливим проблемам.
- Автоматизація керування: Результати аналізу можуть бути використані для створення систем автоматизованого керування, які регулюють параметри середовища та добрив, оптимізуючи їхні значення в реальному часі.

- Моніторинг та управління на віддалі: Засоби дистанційного моніторингу та керування можуть бути інтегровані на основі аналізу даних, щоб дозволити виробникам віддалено контролювати та оптимізувати процес вирощування рослин.

Застосування результатів аналізу даних для оптимізації керування процесом вирощування рослин є ключовим етапом для досягнення високої продуктивності та якості продукції в гідропоніці. Він дозволяє виробникам впроваджувати наукові підходи та технології для досягнення найкращих результатів у сільському господарстві без ґрунту.

1.2 Аналіз існуючих рішень та їх проблем

1.2.1 Аналіз існуючих систем гідропоніки та їхніх переваг і обмежень. Аналіз існуючих систем гідропоніки є важливим етапом у розробці кращих практик вирощування рослин у гідропонічних умовах. У цьому розділі розглядається огляд існуючих систем та їхніх характеристик, а також визначаються переваги і обмеження кожної з них[12].

Переваги існуючих систем гідропоніки:

- Висока продуктивність: Деякі системи гідропоніки дозволяють досягати значно вищих врожаїв порівняно з традиційними методами вирощування рослин.
- Збереження ресурсів: Гідропоніка зазвичай вимагає менше води та добрив у порівнянні з вирощуванням у ґрунті.
- Контроль над умовами: Гідропоніка дозволяє точно контролювати параметри росту рослин, такі як рівень рН, концентрація добрив, освітлення та температура.
- Зменшення ризику захворювань: Відсутність ґрунту може зменшити ризик зараження рослин шкідниками та хворобами.

Обмеження існуючих систем гідропоніки:

- Високі витрати на обладнання: Наявність спеціалізованого обладнання для гідропоніки може бути дорогим фактором.
- Складність у догляді: Деякі системи вимагають великої уваги та навичок для їхнього ведення та обслуговування.
- Електроенергія: Відсутність стабільного живлення може створювати проблеми у функціонуванні систем гідропоніки.
- Ризик системних відмов: Помилки або відмови в системах автоматизації можуть мати серйозний вплив на врожайність.
- Залежність від технології: У разі відмови технології або програмного забезпечення може виникнути загроза втрати врожаю.

Аналіз існуючих систем гідропоніки та їхніх переваг і обмежень є важливим для вибору оптимальної системи та розробки стратегій управління гідропонічними установками. Враховуючи ці переваги і обмеження, можна планувати та оптимізувати процес вирощування рослин у гідропонічних системах.

1.2.2 Проблеми та можливості у гідропоніці та інновації в цій галузі. Цей розділ включає аналіз проблем, з якими стикаються системи гідропоніки, та можливостей для інновацій у цій галузі.

Проблеми у гідропоніці:

- Системи контролю та моніторингу: Відсутність точних і надійних систем контролю може призвести до змін параметрів росту рослин, що негативно впливає на врожайність.
- Енергоспоживання: Використання штучного освітлення та автоматизованих систем може призводити до високого споживання електроенергії, що збільшує витрати.
- Наявність інфекцій та хвороб: Гідропоніка не гарантує захисту рослин від інфекцій та хвороб, і може знадобитися додаткова обробка та профілактика.

Можливості для інновацій в гідропоніці:

- Розвиток ефективних систем контролю: Розробка більш точних та автоматизованих систем контролю параметрів росту рослин для забезпечення стабільних умов вирощування.
 - Використання відновлювальних джерел енергії: Перехід до використання відновлювальних джерел енергії, таких як сонячні батареї, для зменшення енергоспоживання.
 - Біологічний захист: Розробка біологічних методів захисту рослин від хвороб та шкідників для зменшення використання хімічних препаратів.
 - Інтеграція сенсорів та штучного інтелекту: Використання сенсорів та аналізу даних за допомогою штучного інтелекту для автоматизації та оптимізації процесів у гідропоніці.
 - Розробка нових сортів рослин: Селекція та генетична модифікація рослин для підвищення їхньої стійкості та продуктивності в гідропонічних умовах.
- Аналіз проблем та можливостей у гідропоніці та інновації в цій галузі вказує на те, що існують великі можливості для вдосконалення систем гідропоніки та підвищення ефективності вирощування рослин в цих умовах.

1.3 Постановка завдання для магістерської роботи

Метою магістерської роботи є дослідження та аналіз впливу рівня рН на вирощування рослин у системі гідропоніки з використанням даних, зібраних з датчика рН, а також визначення найоптимальнішого значення рН для отримання максимального врожаю. Для досягнення цієї мети були визначені наступні завдання:

- Збір та збереження даних: Провести збір та реєстрацію даних з датчика рН, а також параметрів росту рослин (наприклад, врожайність) протягом тривалого періоду.
- Перенесення даних до бази даних: Організувати процес передачі зібраних даних до бази даних для подальшого аналізу та обробки.

- Аналіз даних: Провести статистичний аналіз отриманих даних з метою встановлення кореляції між рівнем рН і врожайністю рослин.
- Визначення оптимального рН: Встановити значення рН, при якому досягається максимальний врожай та оптимальні умови для росту рослин.
- Порівняння результатів: Порівняти результати експерименту з різними рівнями рН для визначення найефективніших умов вирощування.

2 МОДЕЛЮВАННЯ СИСТЕМИ

2.1 Опис та аналіз методології системного аналізу

Сучасний аграрний сектор і сільське господарство стикаються зі складними викликами, пов'язаними з необхідністю забезпечення стабільного і високоякісного вирощування рослин в умовах обмеженого земельного простору та зменшення впливу на навколишнє середовище[13]. Гідропоніка, яка базується на вирощуванні рослин у водних розчинах з наданням всіх необхідних поживних речовин, води і світла, є одним з інноваційних напрямків в сільському господарстві, що може вирішити ці проблеми.

У контексті гідропоніки і систем керування процесом вирощування рослин, системний аналіз грає важливу роль у розробці та оптимізації цих систем. Для досягнення максимальної ефективності і врожайності необхідно вивчати складну взаємодію всіх компонентів системи, таких як датчики, насоси, фітолампи, контроль рівня води та рН.

Вибір методології моделювання є ключовим етапом у розробці системи керування гідропонікою. У нашій роботі ми використовуємо об'єктно-орієнтований підхід до моделювання системи. Цей підхід дозволяє нам створювати моделі, які відображають об'єкти та їх взаємодію в системі, що важливо для розуміння та оптимізації гідропонічного процесу. Об'єктно-орієнтований підхід дозволяє легше управляти і змінювати компоненти системи, а також розробляти більш ефективні алгоритми керування рослинами та датчиками.

Розробка моделей є важливим етапом в системному аналізі гідропонічних систем. Моделі дозволяють нам відобразити функціонування системи, провести аналіз її роботи та виявити можливі шляхи вдосконалення. У нашому проекті ми створюємо різні типи моделей.

Фізичні моделі дозволяють нам проводити експерименти в реальних умовах і збирати дані щодо фізичних параметрів системи, таких як вологість,

освітленість, рН тощо. Ці дані використовуються для аналізу та налагодження роботи гідропонічної системи.

Математичні моделі базуються на математичних рівняннях і дозволяють нам прогнозувати поведінку системи при різних умовах. Наприклад, ми можемо розробити модель для прогнозування росту рослин в залежності від рівня рН та інших параметрів води.

Комп'ютерні моделі є найбільш гнучкими і дозволяють нам проводити віртуальні експерименти та аналізувати великий обсяг даних. Моделі грають важливу роль у визначенні оптимальних умов для вирощування рослин, управління системою та прийнятті обґрунтованих рішень.

Використання різних типів моделей дозволяє нам зрозуміти, як працює система гідропоніки в різних умовах і вдосконалити її функціонування для досягнення максимальної врожайності та ефективності. Моделі грають важливу роль у визначенні оптимальних умов для вирощування рослин, управління системою та прийнятті обґрунтованих рішень.

2.2 Загальні відомості про моделювання

2.2.1 Визначення UML. У нашому дослідженні важливим інструментом для моделювання системи гідропоніки є UML (Unified Modeling Language) — це стандартизована мова моделювання, розроблена для забезпечення ефективного проектування, аналізу та документування програмних та інших систем. UML ставить перед собою завдання надати зручні та стандартизовані засоби для візуалізації структури та поведінки системи, що є критичним для розуміння та оптимізації складних процесів, які відбуваються у системі гідропоніки.

Походження UML пов'язане з об'єднанням зусиль багатьох висококваліфікованих фахівців з областей програмування, інженерії програмного забезпечення, системного аналізу та моделювання. Завдяки цій спільній ініціативі, UML стала однією з найпопулярніших та найважливіших мов моделювання, і її стандарт підтримується впливовим об'єднанням OMG (Object Management Group).

UML має широкий спектр можливостей, які допомагають моделювати різноманітні аспекти систем. Вона включає різні види діаграм, які можна розділити на дві основні категорії: діаграми структури та діаграми поведінки. Діаграми структури дозволяють описати компоненти системи, такі як класи, об'єкти, компоненти, а також відносини між ними. Діаграми поведінки служать для моделювання взаємодії компонентів та системи в цілому, включаючи сценарії виконання.

Використання UML в моделюванні системи гідропоніки дозволяє команді розробників здійснювати більш точну та структуровану специфікацію системи, полегшує взаєморозуміння між учасниками проекту, сприяє виявленню та виправленню можливих недоліків у дизайні. UML може бути успішно використана для моделювання не лише програмних систем, але й будь-яких систем різного призначення, включаючи бізнес-процеси та апаратні рішення.

2.2.2 Об'єктно-орієнтоване проектування. Об'єктно-орієнтоване проектування (ООП) є ключовим аспектом розробки системи гідропоніки. Для візуалізації та моделювання архітектури системи та взаємодії між її компонентами використовується Уніфікована мова моделювання (UML). UML - це стандартний набір нотацій та діаграм, який дозволяє інженерам зобразити та розуміти структуру та поведінку системи.

Основні концепції UML включають:

- Діаграми класів: Діаграми класів дозволяють представити структуру системи, вказуючи класи, їх атрибути та методи. У контексті гідропоніки, це може бути корисним для моделювання класів, таких як "Рослина", "Датчик", "Насос" тощо.
- Діаграми послідовності: Діаграми послідовності демонструють, як об'єкти взаємодіють між собою у певній послідовності для виконання певних операцій або сценаріїв. Наприклад, вони можуть вказувати, як датчики вимірюють рівень води та передають цю інформацію системі керування.

- **Діаграми активності:** Діаграми активності моделюють послідовність дій у конкретному процесі або функції. Вони можуть бути корисними для розуміння алгоритмів управління рослинами та іншими компонентами системи.
- **Діаграми взаємодії:** Діаграми взаємодії розглядають взаємодію між об'єктами у певному контексті, зокрема сценаріями використання. Вони допомагають зрозуміти, як об'єкти спілкуються один з одним.
- **Діаграми станів:** Діаграми станів моделюють різні стани об'єкта та переходи між ними. Вони можуть бути корисними для моделювання станів рослин та системи водопостачання.

Застосування UML в ООП дозволяє команді розробників створювати чіткі та докладні моделі системи гідропоніки, що полегшує розробку та розуміння системи перед реалізацією. Він також допомагає виявляти можливі проблеми та оптимізувати архітектуру до введення системи в дію.

2.3 Діаграма прецедентів

Діаграма такого виду показує зв'язок між акторами та прецедентами системи, яка встановлює обмеження. Основними елементами даної діаграми є актори та прецеденти.

Актор у діаграмі прецедентів – це будь-який користувач чи об'єкт, який взаємодіє з системою.

Прецедент – це певна дія, яка виконується конкретним актором у системі. Прецедент може бути напряму пов'язаний з актором, з яким взаємодіє, або з іншим прецедентом, від виконання якого залежить виконання даного прецеденту.

У діаграмі прецедентів можуть бути такі варіанти взаємодії:

- використання;
- включення;
- розширення;
- вимога;

- схожість;
- рівнозначність.

Актор може бути пов'язаний з прецедентом напряму або через інший прецедент. Якщо немає прямого зв'язку між актором та прецедентом, то такий прецедент називається абстрактним.

Між прецедентами можуть бути такі зв'язки:

- Включення, або include – це взаємозв'язок, коли залежний прецедент обов'язково виконується на основі базового.
- Розширення, або extend – це різновид взаємозв'язку, коли базовий прецедент не обов'язково використовує дочірній.

Для побудови діаграми прецедентів розроблюваної системи перш за все необхідно визначити акторів, які взаємодіють з нею. На таблиці 2.1 показано перелік акторів, які взаємодіють з системою.

Актор	Опис
1	2
Оператор	Спостерігає за значеннями показників температури й вологості в гідропоніці, за необхідності регулює вручну, опрацьовує виміряні значення, вносить дані про врожай
Аналітик	Проводить аналіз роботи підприємства та формує аналітичну звітність
Директор	Приймає рішення щодо управління господарством
Блок керування	Зчитує показники з датчиків та керує роботою виконавчих механізмів
Датчик кислотності	Вимірює значення кислотності в суміші
Насос	При необхідності корегує значення кислотності в суміші

Таблиця 2.1 Актори, які діють у системі

Визначившись з акторами, які діють у системі, можна описати прецеденти. На основі всіх обов'язків акторів побудована діаграма прецедентів, з якою можна ознайомитись на рис. 2.1.

У зв'язку з тим, що мета розроблюваної системи зосереджена на аналізі зафіксованих даних головним актором виступає аналітик. Саме він відповідає за ведення аналізу всіх показників та взаємозв'язків між ними. На основі аналітичної звітності директор запроваджує необхідні зміни в зазначених бізнес-процесах. Також у системі акторами виступають оператор, блок керування, датчик кислотності та насос. Блок керування, датчик та насос, апаратне забезпечення системи, виконують роль моніторингу речовин в живильній трубі. Працівник підприємства бере участь у ручному управлінні та регулярному внесенні необхідних даних.

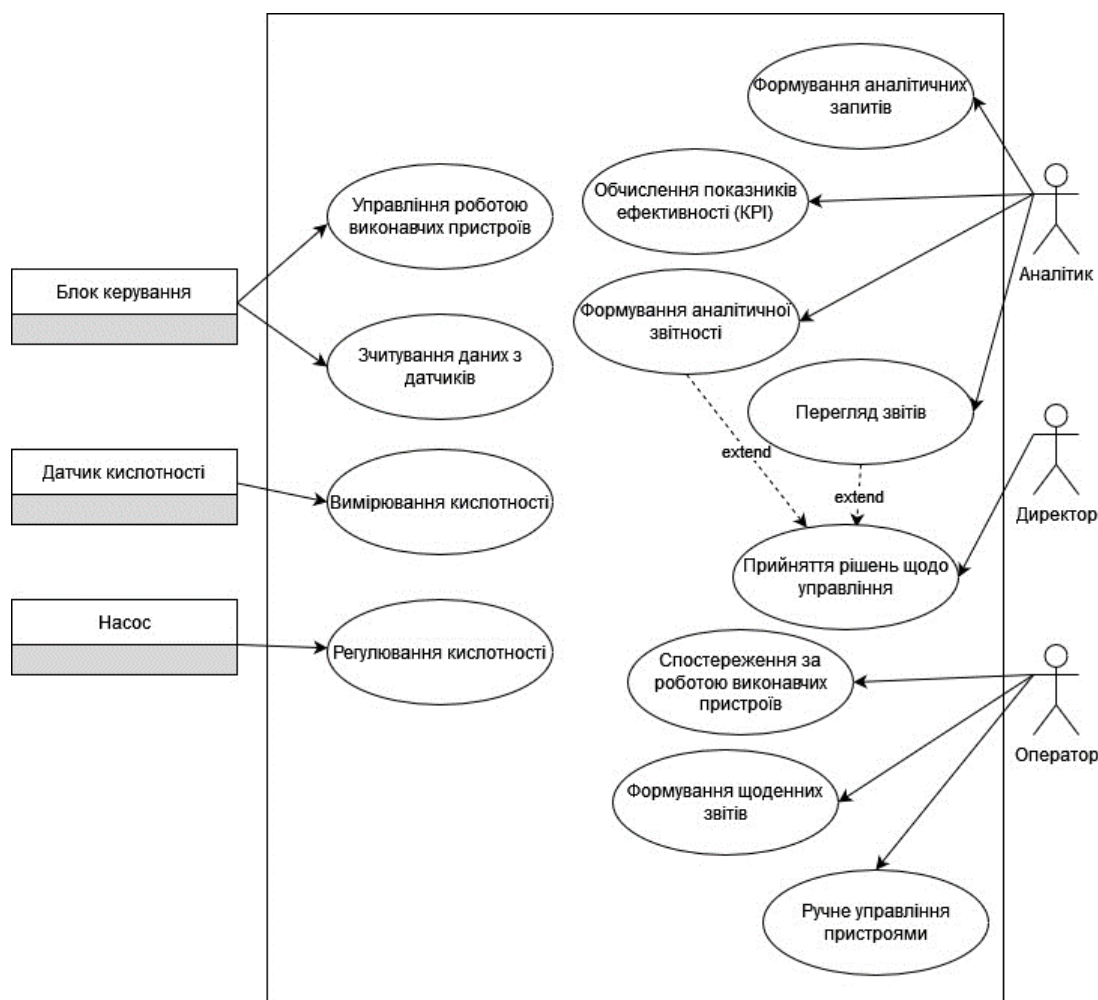


Рисунок 2.1 Діаграма прецедентів

2.4 Діаграма послідовності

Діаграма послідовності - це важливий інструмент у системному аналізі та проектуванні, який допомагає візуалізувати послідовність взаємодій між об'єктами або компонентами системи. Ця діаграма дозволяє детально

проаналізувати порядок виконання операцій, функцій чи процесів в системі та визначити взаємозв'язки між ними. Основними елементами діаграми послідовності є об'єкти, лінії життєвого циклу та повідомлення.

- **Об'єкти:** Об'єкти на діаграмі послідовності представляють суб'єкти, які беруть участь у взаємодії. Це можуть бути користувачі, компоненти системи, зовнішні системи, або будь-які інші учасники. Кожен об'єкт визначається ім'ям і може мати свій життєвий цикл, під час якого він виконує певні дії або операції.
- **Лінії життєвого циклу:** Лінії життєвого циклу з'єднують об'єкти і показують, коли і як об'єкти взаємодіють між собою. Вони можуть мати спеціальні засоби для позначення умов або ітерацій взаємодії. Лінії життєвого циклу допомагають визначити порядок виконання дій та послідовність подій.
- **Повідомлення:** Повідомлення на діаграмі послідовності показують взаємодію між об'єктами. Вони описують, які дії виконуються та яка інформація передається від одного об'єкта до іншого. Повідомлення можуть бути супроводжені параметрами, що вказують на передачу даних між об'єктами.

Діаграми послідовності часто використовуються для аналізу та оптимізації бізнес-процесів, проектування систем керування, визначення послідовності виконання операцій у програмному забезпеченні та багатьох інших областях. Вони сприяють розумінню та візуалізації важливих аспектів взаємодії між об'єктами системи, що допомагає удосконалити проект та забезпечити його відповідність функціональним вимогам.

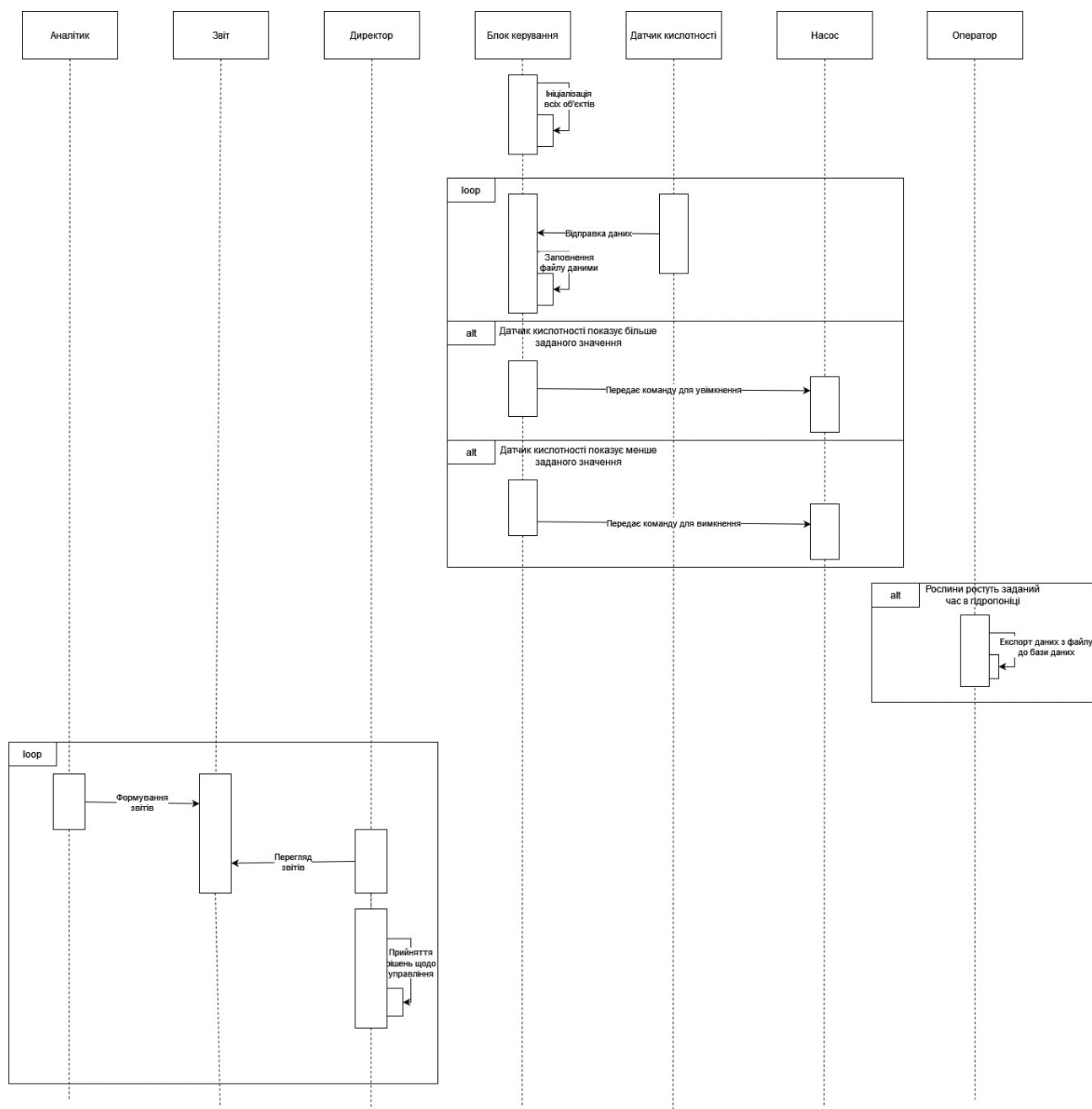


Рисунок 2.2 Діаграма послідовності

В діаграмі послідовності на рис. 2.2 зображено різні сценарії взаємодії між акторами та сутностями в контексті управління гідропонікою та аналітичної діяльності. Перший сценарій включає у себе спільну роботу блока управління, насоса та датчика кислотності, де блок управління вмикається, тим самим запускаючи всі елементи системи, після чого в циклі він отримує дані від датчика кислотності та записує їх в файл, потім у нас знаходиться умова, яка означає що після відправки не відповідних даних до блока управління, а саме кислотність більша заданого рівня, вмикається насос, який викачує добрива з однієї ємності

в ємність гідропоніки, щоб понизити рівень кислотності до заданого, після виходу з умови у ми знову йдемо до циклу, отримуємо задані значення і потім через іншу умову передаємо сигнал до насосу і вимикаємо його.

Після певного терміну в роботу підключається оператор, який повинен зібрати всі дані з датчиків і перенести їх до бази даних. Аналітик формує звіти після отримання даних, після чого директор має змогу подивитись ті звіти і зробити висновки по прийняттю рішень в майбутніх посадках.

Все це показує як працює гідропоніка між акторами та сутностями у контексті управління підприємством та аналізу даних.

Діаграма послідовності є важливим інструментом у системному аналізі та проектуванні. Вона дозволяє візуалізувати послідовність взаємодій між об'єктами або компонентами системи, а також детально аналізувати порядок виконання операцій, функцій чи процесів. Діаграма послідовності допомагає визначити взаємозв'язки між об'єктами, розуміти порядок виконання дій та послідовність подій.

Цей інструмент дуже корисний для аналізу та оптимізації бізнес-процесів, проектування систем керування та програмного забезпечення. Він дозволяє краще розуміти та візуалізувати важливі аспекти взаємодії між об'єктами системи, що сприяє удосконаленню проекту та забезпеченню відповідності функціональним вимогам. Діаграма послідовності є потужним інструментом для покращення якості та ефективності розроблених систем і процесів.

2.5 Діаграма розгортання і компонентів

Діаграма розгортання та компонентів є ключовими інструментами для подальшого аналізу та проектування системи гідропоніки. Ці інструменти дозволяють нам детально вивчати архітектуру системи та взаємозв'язки між її складовими частинами.

Діаграма розгортання визначає, як система фізично розгортається на апаратному рівні. Вона відображає апаратне забезпечення, на якому працює наша система, такі як сервери, сенсори, мережеве обладнання тощо. Діаграма

розгортання допомагає нам зрозуміти, де розташовані різні компоненти системи і як вони взаємодіють один з одним.

Діаграма компонентів, у свою чергу, фокусується на логічній архітектурі системи. Вона визначає компоненти програмного забезпечення, які входять до складу системи, та їх взаємозв'язки. Ця діаграма важлива для розуміння функціональності різних частин системи та допомагає налагодити правильну взаємодію між ними.

Аналіз і створення діаграм розгортання та компонентів допомагає нам краще розуміти архітектурні аспекти системи гідропоніки. Це важливий крок у підготовці до наступних етапів проєктування та розробки, які полягатимуть в розробці та імплементації компонентів, які відповідають за функціональність системи, і їх розгортанні на апаратному забезпеченні.

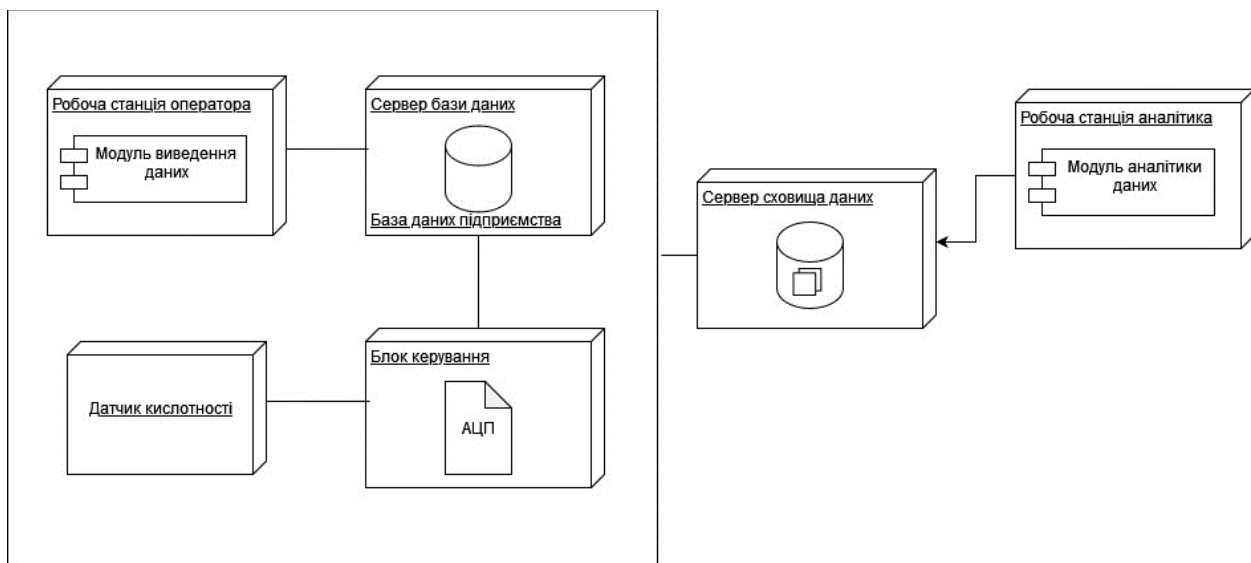


Рисунок 2.3 Діаграма розгортання і компонентів

На рис. 2.3 зображено архітектуру розроблюваної системи. Основними фізичними вузлами є:

- робоча станція підприємства;
 - модуль внесення даних в бд;
 - оперативна бд гідропоніки;
 - блок керування;
 - датчик кислотності;

- насос;
- сервер сховища даних;
- робоча станція аналітика.

3 РОЗРОБКА СИСТЕМИ

3.1 Аналіз джерел інформації

3.1.1 Структура оперативної бази даних. Ця база даних є важливою частиною проекту з гідропоніки і розробки системи управління гідропонічними об'єктами.

Мета і призначення бази даних:

- Система управління гідропонічними об'єктами: База даних створена з метою збереження та управління інформацією, пов'язаною з гідропонічними системами. Вона дозволяє детально відстежувати та керувати гідропонічними об'єктами, такими як грядки, рослини, датчики та обладнання.
- Відстеження параметрів росту рослин: База даних дозволяє записувати параметри росту рослин, такі як кислотність, вага тощо. Ця інформація допомагає вам вести докладну статистику та аналіз росту рослин.
- Слідкування за врожаєм: За допомогою цієї бази даних ви можете записувати і відстежувати дані про збори, включаючи вагу та дату. Це дозволяє вам ефективно контролювати виробництво та врожаї.
- Управління обладнанням і датчиками: База даних також містить інформацію про обладнання та датчики, що використовуються в гідропоніці. Ви можете вести журнали обслуговування та стану обладнання, а також моніторити показники датчиків в режимі реального часу.

Застосування:

- Моніторинг гідропонічних систем: Використання бази даних для моніторингу та керування гідропонічними системами, включаючи віддалене керування параметрами середовища та обладнанням.
- Аналіз та оптимізація виробництва: Збирання та аналіз даних про ріст рослин та врожаї дозволяє оптимізувати процеси гідропонічного виробництва та підвищити врожайність.

- Звіти та аналізи: База даних надає можливість створювати різноманітні звіти та аналізи на основі накопичених даних. Можна створювати звіти про стан рослин, якість врожаю, ефективність датчиків та багато іншого. Ці аналізи допомагають приймати обґрунтовані рішення щодо виробництва та оптимізації ресурсів.
- Підвищення продуктивності: Завдяки можливостям цієї бази даних є можливість підвищити продуктивність гідропонічного господарства, зменшити ризики та витрати, а також забезпечити кращий контроль над рослинами та урожаєм.
- Підтримка прийняття рішень: База даних надає доступ до даних в реальному часі, що сприяє швидкому реагуванню на зміни у гідропонічних системах та навколишньому середовищі. Ви можете приймати рішення на основі актуальних даних.
- Автоматизація процесів: Інформація, збережена в базі даних, може бути використана для автоматизації процесів поливу, регулювання середовища та управління обладнанням.
- Віддалений моніторинг: База даних може бути доступна віддалено, що дозволяє вам відстежувати та керувати гідропонічними системами з будь-якого місця, де є доступ до Інтернету.

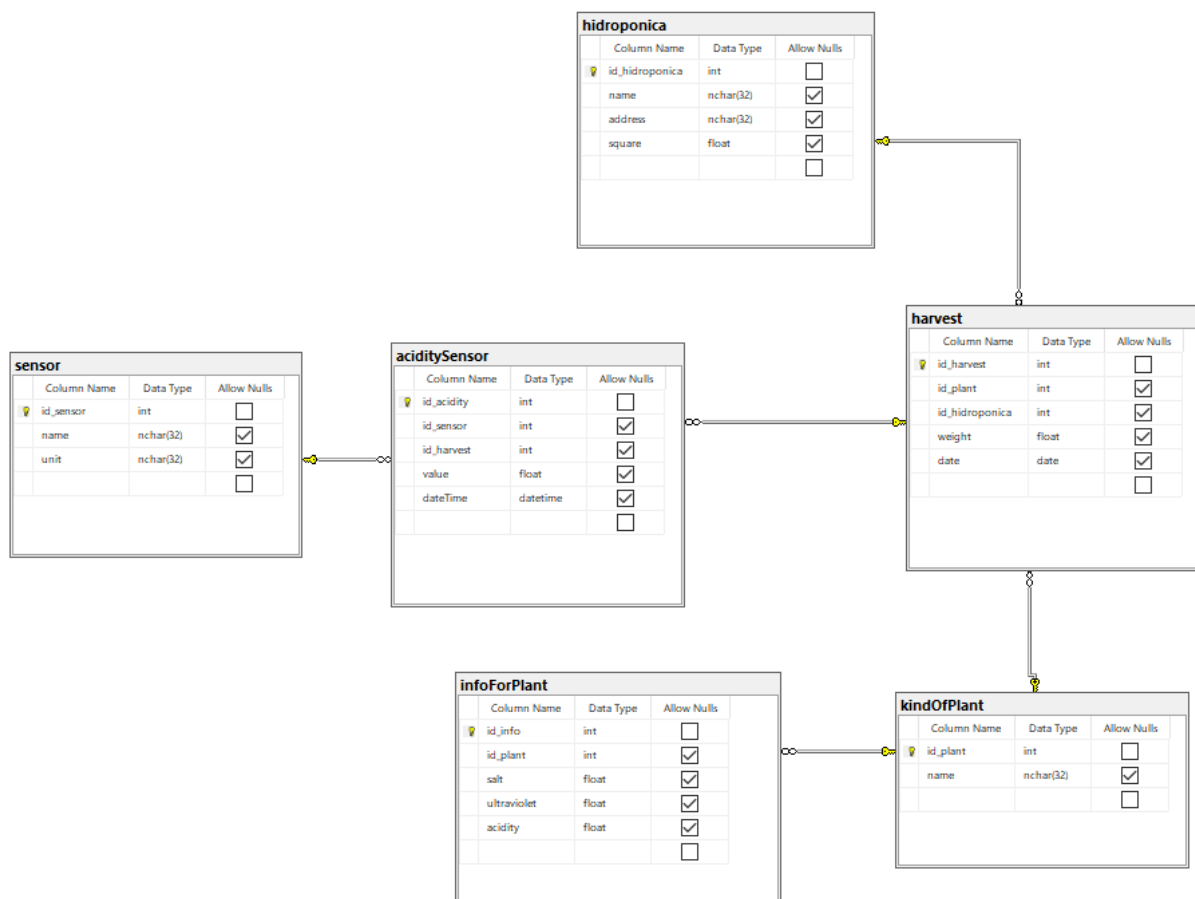


Рисунок 3.1 Структура оперативної бази даних

База даних, яка зображена на рис. 3.1, є надзвичайно важливим інструментом для ефективного управління та моніторингу гідропонічних систем. Вона дозволяє вести докладний облік рослин, урожаїв та параметрів росту, а також допомагає вам приймати обґрунтовані рішення та оптимізувати виробництво. За допомогою цієї бази даних ви можете досягти високої продуктивності та якості у галузі гідропоніки.

3.1.2 Фундаментальні принципи OLAP-технологій. Оперативна обробка даних є важливим компонентом OLAP-технологій і включає в себе кілька ключових аспектів:

- Трансформація даних: Однією з основних задач оперативної обробки даних є їхнє перетворення для підготовки до аналізу. Це може включати в себе вибір потрібних даних, агрегацію, фільтрацію і сортування.

Трансформація даних допомагає підготувати дані для створення багатовимірних кубів, які використовуються для аналізу та візуалізації.

- Завантаження даних: Оперативна обробка включає в себе завантаження даних з різних джерел. Це можуть бути операційні бази даних, зовнішні джерела даних або файли. Під час завантаження даних важливо враховувати їхню структуру та якість, і забезпечувати зв'язок між різними джерелами і таблицями для забезпечення консистентності та інтеграції даних.
- Підготовка до аналізу: Оперативна обробка включає в себе підготовку даних для подальшого аналізу. Це може включати в себе обчислення нових показників або агрегатів, які зручні для користувачів. Також ця підготовка включає валідацію та очистку даних, щоб забезпечити їхню якість та достовірність.

Оперативна обробка даних в OLAP-технологіях спрямована на створення підґрунтя для подальшого аналізу, візуалізації і прийняття рішень. Вона дозволяє організаціям ефективно керувати та аналізувати великі обсяги даних, враховуючи їхню структуру та значущість для процесів прийняття рішень. Трансформація та підготовка даних є важливим кроком у забезпеченні, що дані готові для використання в OLAP-системі, і вони відображають реальний стан справ суб'єкта аналізу.

Таким чином, оперативна обробка даних у OLAP-технологіях дозволяє ефективно використовувати дані для розуміння та вдосконалення діяльності підприємства, а також сприяє покращенню процесів прийняття рішень на основі фактичних даних.

Механізми аналізу є важливою складовою OLAP-технологій і дозволяють користувачам виконувати глибокий аналіз та отримувати цінні інсайти зі структурованих даних. Вони розширюються від простого обрання конкретних значень до складних аналітичних операцій та візуалізації результатів. Розглянемо докладніше деякі механізми аналізу в OLAP-технологіях:

- Вибірка даних: Один із перших етапів аналізу - вибірка конкретних даних для подальшого розгляду. Користувачі можуть вибирати певні значення, дати або діапазони даних, які цікавлять їх найбільше. Вибірка може бути як простою фільтрацією, так і складним пошуком за різними параметрами.
- Групування і агрегація: Для отримання узагальнених даних користувачі можуть використовувати групування і агрегацію. Вони можуть об'єднувати дані в окремі категорії та обчислювати суми, середні значення, мінімальні або максимальні значення. Це дозволяє отримувати загальні висновки з великих обсягів даних.
- Розрізання і деталізація: Механізми аналізу включають в себе здатність розглядати дані на різних рівнях деталізації. Користувачі можуть переходити від загальних даних до більш детальних і обернено. Це дозволяє вивчати дані на різних рівнях абстракції та отримувати глибокі інсайти.
- Аналітичні операції: Механізми аналізу включають в себе широкий спектр аналітичних операцій. Користувачі можуть виконувати різноманітні обчислення, включаючи порівняння, прогнозування, рейтинги, виявлення аномалій і багато інших операцій. Це дозволяє користувачам вивчати дані в глибину та в ширину.
- Візуалізація результатів: Важливою частиною механізмів аналізу є можливість візуалізації результатів. Користувачі можуть створювати графіки, діаграми, теплові карти, географічні карти та інші візуалізації для наглядного представлення даних.

Механізми аналізу в OLAP-технологіях допомагають користувачам максимально використовувати дані для прийняття обґрунтованих рішень, виявлення тенденцій та вдосконалення діяльності підприємства. Вони роблять аналіз даних більш доступним і ефективним, сприяючи досягненню бізнес-цілей.

3.1.3 Організація сховища даних. Однією з фундаментальних складових OLAP-технологій є організація сховища даних, яке грає ключову роль у забезпеченні ефективності та доступності даних для аналізу. Організація сховища даних передбачає структуровану і ефективну систему збереження і управління даними, яка спрощує їх витягування, обробку і аналіз.

Ключові аспекти організації сховища даних в рамках OLAP-технологій:

- **Многомірна модель даних:** Однією з основних характеристик сховища даних OLAP є використання багатовимірних моделей даних. Вони дозволяють структурувати дані у вигляді кубів, які мають багато вимірів. Ця структура спрощує аналіз і навігацію даними, оскільки користувачі можуть вибирати виміри та переглядати дані на різних рівнях деталізації.
- **ETL процеси:** Оперативна організація сховища даних починається з процесів ETL (вилучення, трансформація і завантаження). ETL включає в себе вилучення даних з джерел, їхню трансформацію у відповідні дані для OLAP, і завантаження в сховище даних. Цей процес дозволяє підготувати дані для подальшого аналізу та забезпечити їхню цілісність.
- **Методи збереження даних:** Сховище даних використовує різні методи збереження даних, зокрема індексацію, компресію і розподілення. Індиксація допомагає знизити час доступу до даних, компресія - зменшити вимоги до простору, а розподілення - підтримувати високу доступність та надійність.
- **Оптимізація запитів:** Системи OLAP володіють механізмами оптимізації запитів, які дозволяють ефективно витягувати дані для аналізу. Це включає в себе використання попередньо побудованих агрегатів, кешування запитів та інші техніки, які покращують продуктивність.
- **Безпека даних:** Оскільки дані в сховищі містять чутливу інформацію, важливо забезпечити високий рівень безпеки. Це включає в себе контроль доступу до даних, шифрування та аудит дій користувачів.

- Спеціалізовані сховища: У більших організаціях зазвичай використовують спеціалізовані сховища даних для підтримки великого обсягу даних та швидкого аналізу. Це можуть бути колоночні бази даних, NoSQL сховища або розподілені сховища.
- Підтримка різних джерел даних: OLAP-системи зазвичай підтримують інтеграцію з різними джерелами даних, включаючи реляційні бази даних, документи, веб-сервіси і багато інших. Це дозволяє користувачам об'єднувати дані з різних джерел для глибшого аналізу.

Важливість організації сховища даних в OLAP-технологіях:

Організація сховища даних грає важливу роль у забезпеченні ефективного та продуктивного аналізу даних. Вірно спроектована та оптимізована інфраструктура дозволяє користувачам швидко та легко вибирати дані для аналізу, виконувати складні запити та генерувати цінні інсайти. Ретельний вибір методів збереження, оптимізація запитів і забезпечення безпеки гарантують успішний аналіз та використання даних для підтримки прийняття рішень.

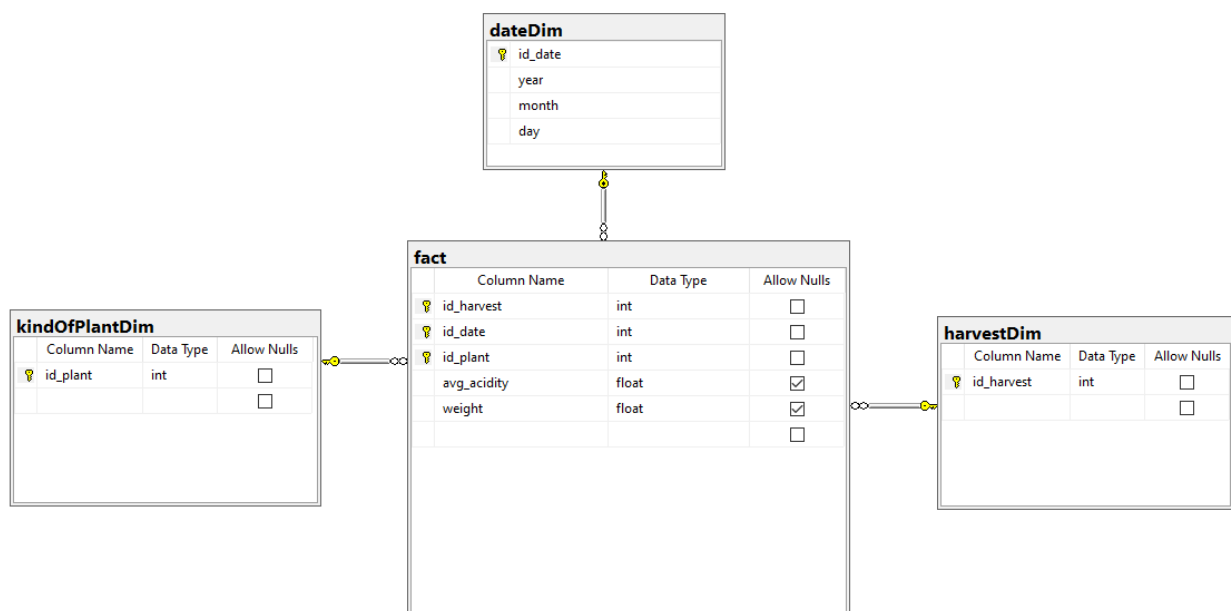


Рисунок 3.2 Структура сховища даних

Сховище даних, яке зображено на рис. 3.2, розроблено для інтегрованої системи управління гідропонікою. Воно забезпечує зберігання та доступ до даних про врожаї, рослини та дати.

Таблиця "dateDim" містить інформацію про різні дати, включаючи рік, місяць та день. Кожна дата ідентифікується унікальним ідентифікатором, що дозволяє асоціювати врожаї з конкретними датами.

Таблиця "harvestDim" призначена для зберігання даних про врожаї. Кожен врожай отримує свій унікальний ідентифікатор. Ця таблиця дозволяє системі фіксувати докладну інформацію про кожний окремий врожай, включаючи параметри як вага та якість.

Таблиця "kindOfPlantDim" містить інформацію про різні види рослин, що вирощуються у гідропоніці. Кожен вид рослин також має свій унікальний ідентифікатор. Ця таблиця важлива для ідентифікації та класифікації рослин у системі управління.

Таблиця "fact" є центральною частиною сховища даних і містить фактичні дані про врожаї. Кожен рядок в цій таблиці асоційований з певним врожаєм, певною датою та певним видом рослин. Це дозволяє зберігати історичні дані про врожаї та вивчати їх розвиток з часом. В цій таблиці фіксуються середні показники кислотності та ваги врожаю для кожного окремого запису.

Ця структура сховища даних дозволяє системі управління гідропонікою докладно відстежувати та аналізувати різні аспекти вирощування рослин, включаючи якість врожаю, тенденції та зміни в часі.

3.1.4 Методи вилучення, обробки і передачі даних. Служби аналізу SQL Server (SSAS) — це багатовимірний OLAP-сервер, а також механізм аналітики, який дозволяє розділяти великі обсяги даних. Він є частиною Microsoft SQL Server і допомагає виконувати аналіз за допомогою різних вимірів. Він має 2 варіанти багатовимірний і табличний.

Для розробки кубу було використано середовище Visual Studio з розширенням SSAS. На першому етапі необхідно визначити джерело даних— база даних OLAP або сховище даних. На основі визначеного джерела даних будуть імпортуватись необхідні дані. На рис. 3.3 зображено формування

підключення джерела даних за допомогою модуля Data Source Wizard, де ми обираємо попередньо створене сховище даних.

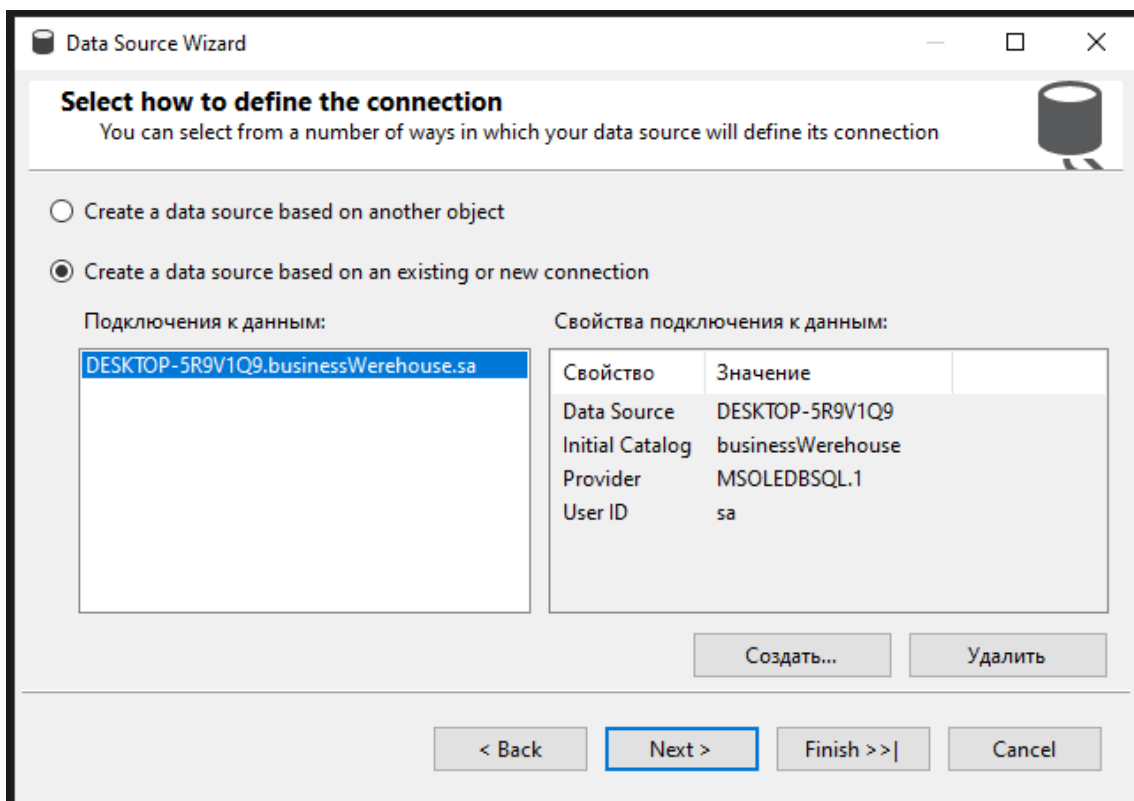


Рисунок 3.3 Підключення до джерела даних

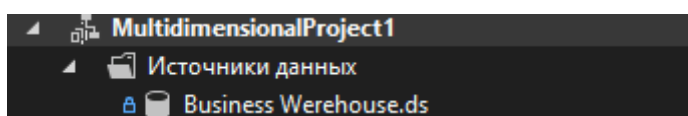


Рисунок 3.4 Підключені джерела даних

Створення Data Source View включає створення підмножини джерела даних, яка буде використовуватися для наповнення сховища. Ця підмножина може включати в себе як таблиці, так і представлення (view) з реляційної бази даних, що є джерелом даних. Для створення такої підмножини джерела необхідно обрати саме те джерело даних та таблиці, які необхідні для вирішення конкретних завдань і вимог.

Наступним кроком у розробці куба є створення Data Source View (DSV), яке є абстракцією реляційного джерела даних і є основою для створення багатовимірних кубів і вимірів у проекті. Основною метою DSV є забезпечення контролю над структурою даних, які використовуються у проекті, незалежно від джерела даних, на яке воно посилається. Це означає, що ви можете

перейменовувати або об'єднувати стовпці у DSV без необхідності безпосередньо впливати на вихідне джерело даних. Під час розробки проекту можна створити кілька DSV у проекті або базі даних Analysis Services, які базуються на одному або кількох джерелах даних, для того щоб вони відповідали конкретним вимогам різних рішень.

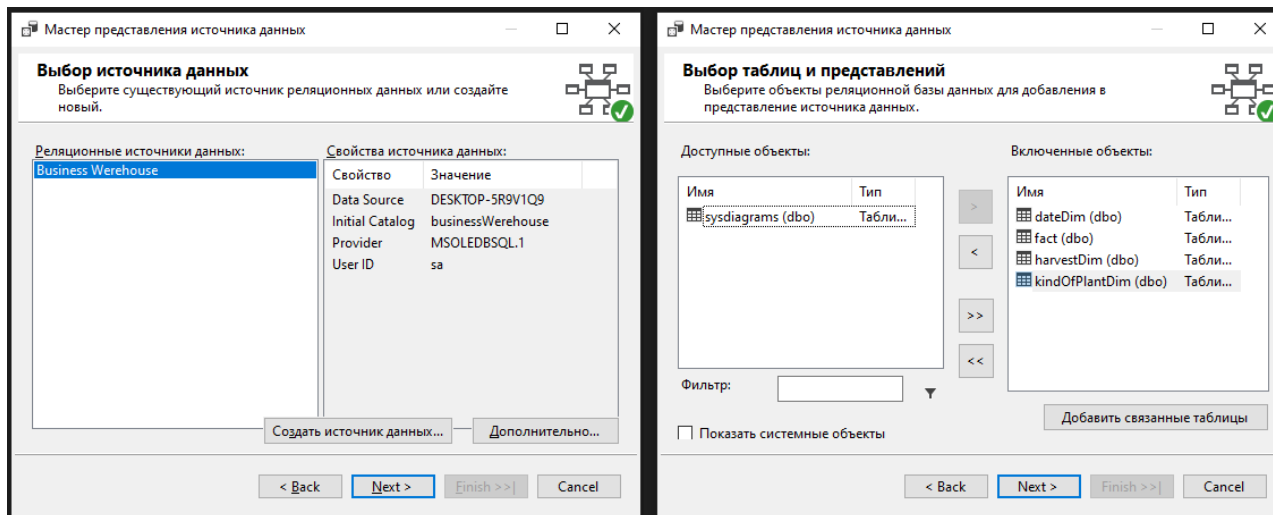


Рисунок 3.5 Створення Data Source View

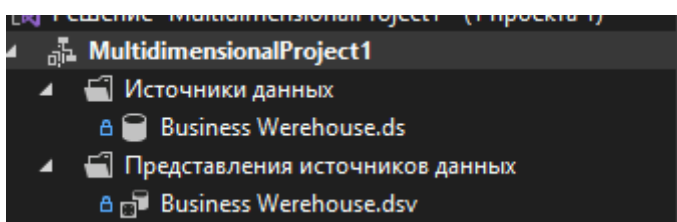


Рисунок 3.6 Створений Data Source View

Виміри (Measures) в багатовимірних OLAP-кубах представляють собою числові дані, які аналізуються та агрегуються під час взаємодії з кубом. Вони відображають ключові метрики та статистику, яку бажають дослідити. Опис вимірів у багатовимірних OLAP-кубах у Visual Studio:

- **Значення вимірів:** Виміри представлені числовими значеннями, такими як суми, кількості, середні, відсотки, середньоквадратичні відхилення тощо. Ці значення розраховуються на основі фактичних даних, які зберігаються в кубі.
- **Деталізація даних:** Виміри можуть бути розглянуті на більш низькому рівні деталізації.

- Функції агрегації: Можна визначити, як саме дані агрегуються для вимірів.
- Розрахунки і вирази: Виміри можуть містити розрахункові поля та вирази.
- Участь у звітах і діаграмах: Виміри використовуються для створення звітів, діаграм та інших візуалізацій, які надають користувачам зрозумілу інформацію з певних аспектів даних.
- Фільтрація і розбиття даних: Виміри можуть використовуватися для фільтрації та розбиття даних.
- Підсумки і підсумкові рівні: Виміри дозволяють створювати підсумки на різних рівнях деталізації, що допомагає користувачам отримувати комплексний огляд даних.

Виміри є основними будівельними блоками багатовимірних OLAP-кубів, і вони дозволяють вам аналізувати та взаємодіяти з даними з різних перспектив та рівнів деталізації.

Було створено виміри:

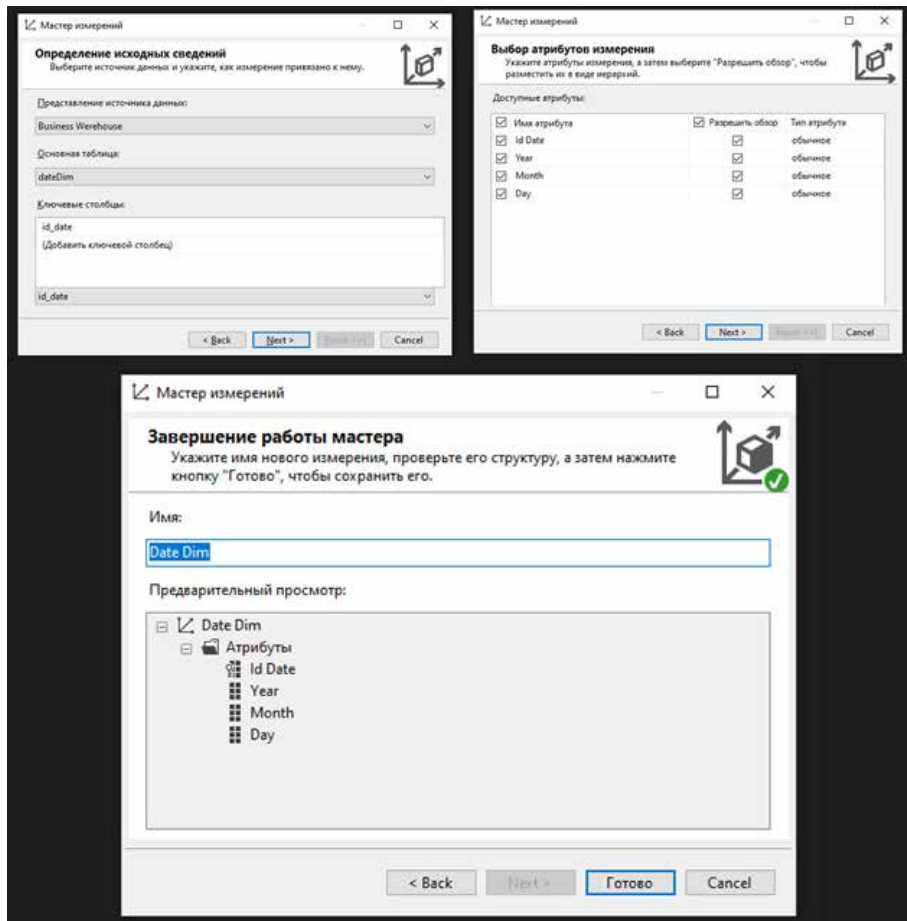


Рисунок 3.7 Створення виміру Date Dim

Так само створено інші виміри.

На основі створеного зрізу джерела за допомогою Cube Wizard було сформовано гіперкуб, який зображено на рис. 3.9.

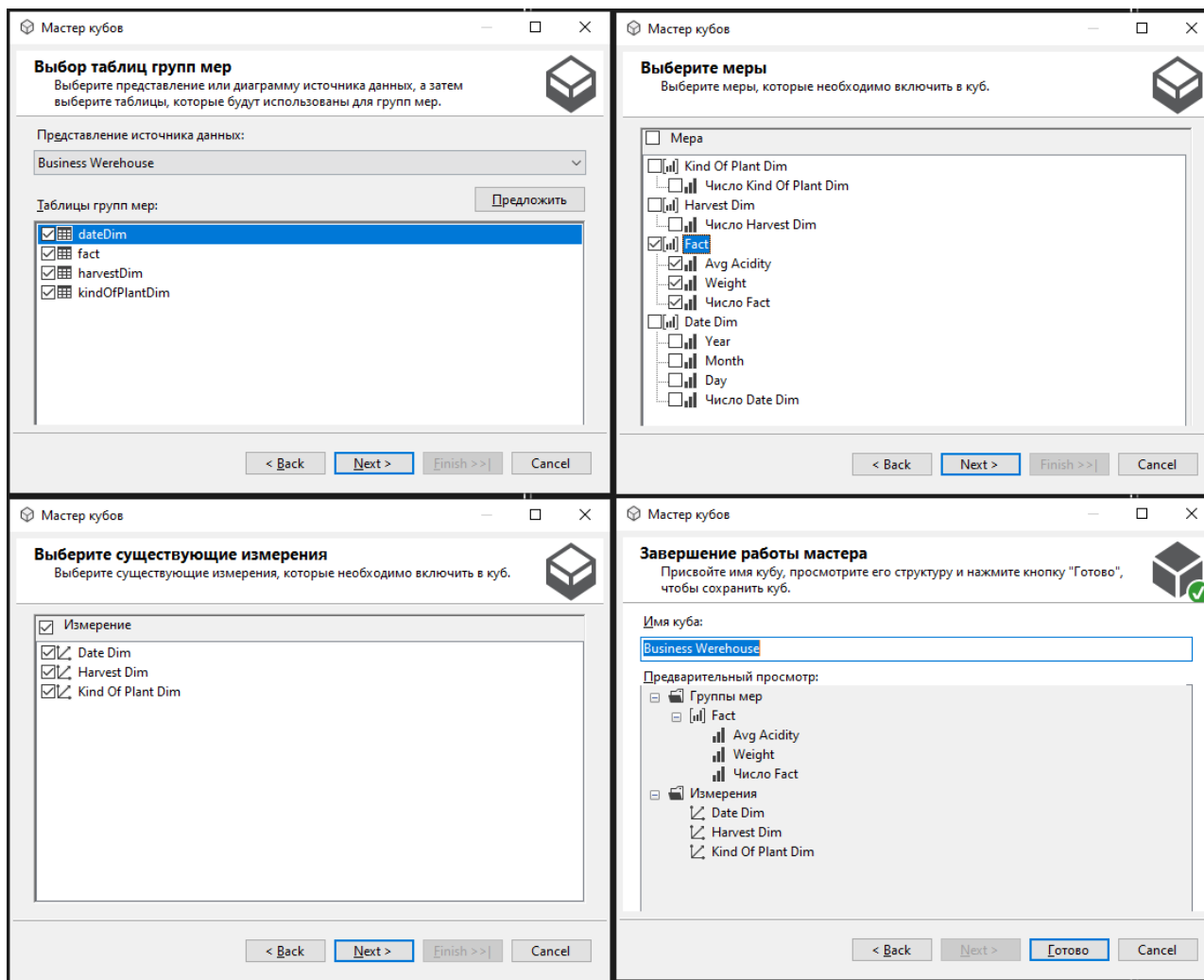


Рисунок 3.8 Створення гіперкубу

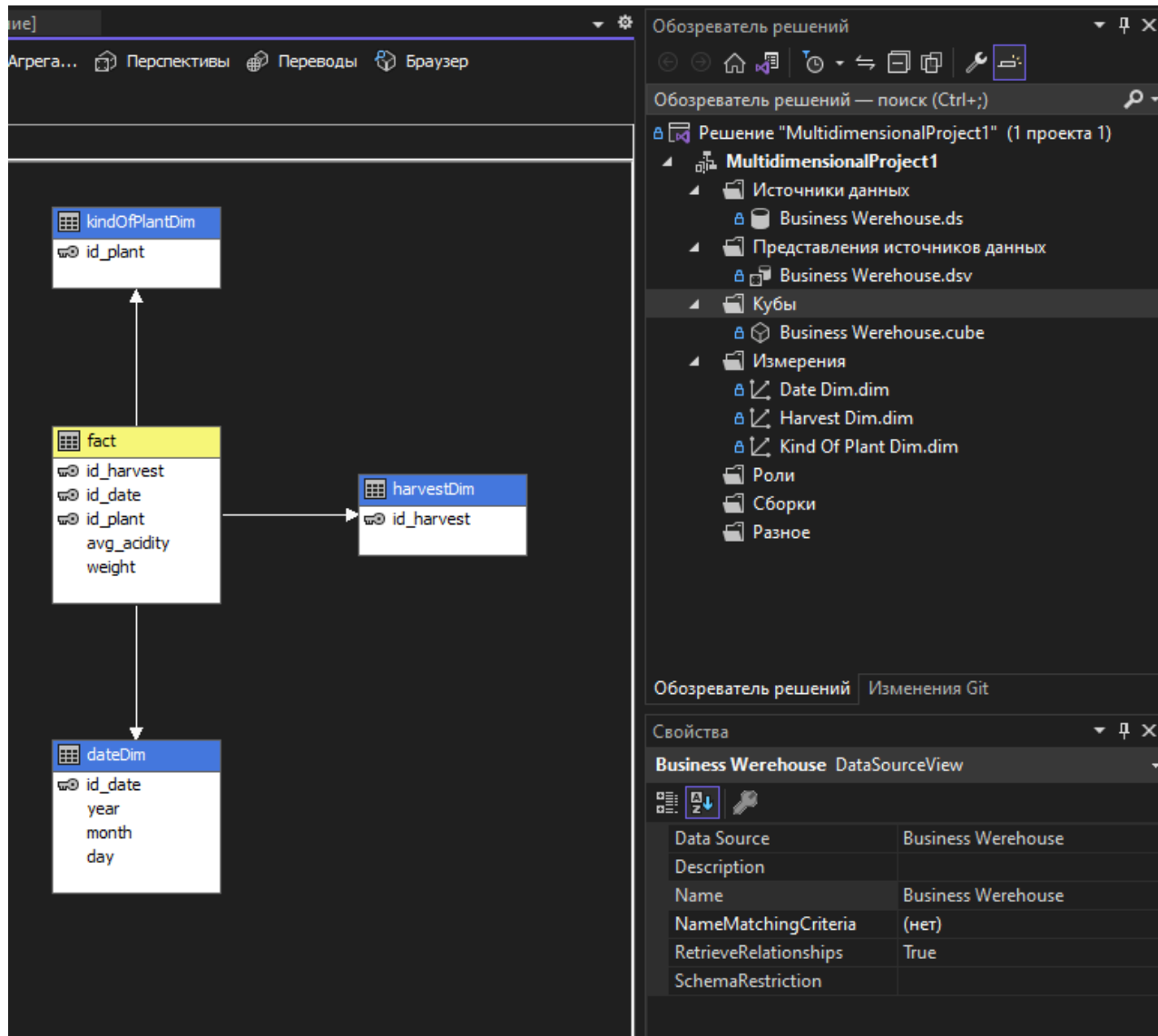


Рисунок 3.9 Сформований куб

Далі введено пароль і запущено, після чого можна побачити на рис. 3.10, що куб розгорнувся вдало:

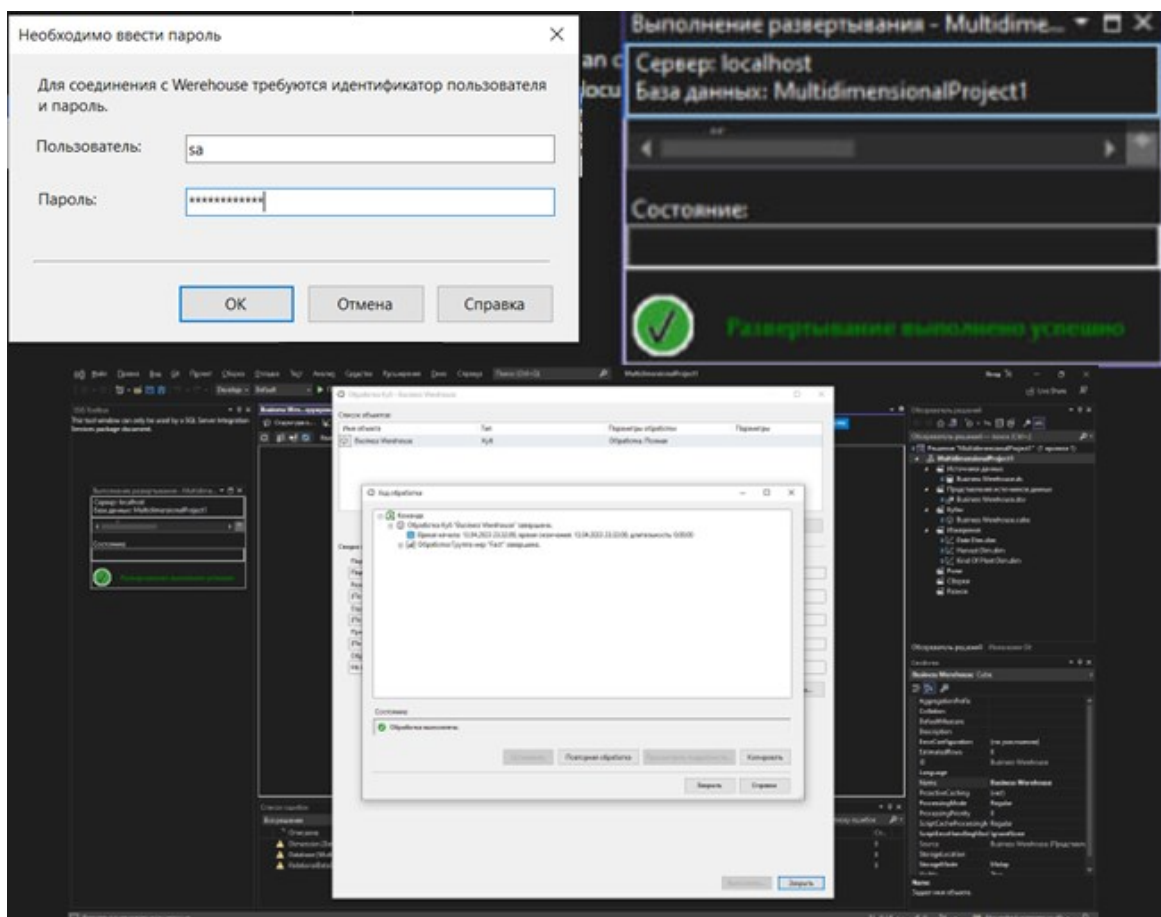


Рисунок 3.10 Вдале розгортання куба

Далі необхідно організувати процес інтеграції даних в багатовимірний куб. Створюємо проект Integration Services Project служби SSIS.

Для встановлення зв'язку з джерелом та приймачем даних необхідно вказати провайдер, сервер та БД. У таблиць джерела та приймача має бути однакова структура, однакові типи даних. Також необхідно впевнитися, що приймач не містить в собі дані, які будуть інтегруватися з джерела, оскільки дублювання даних викличе помилку при виконання завдання потоків даних.

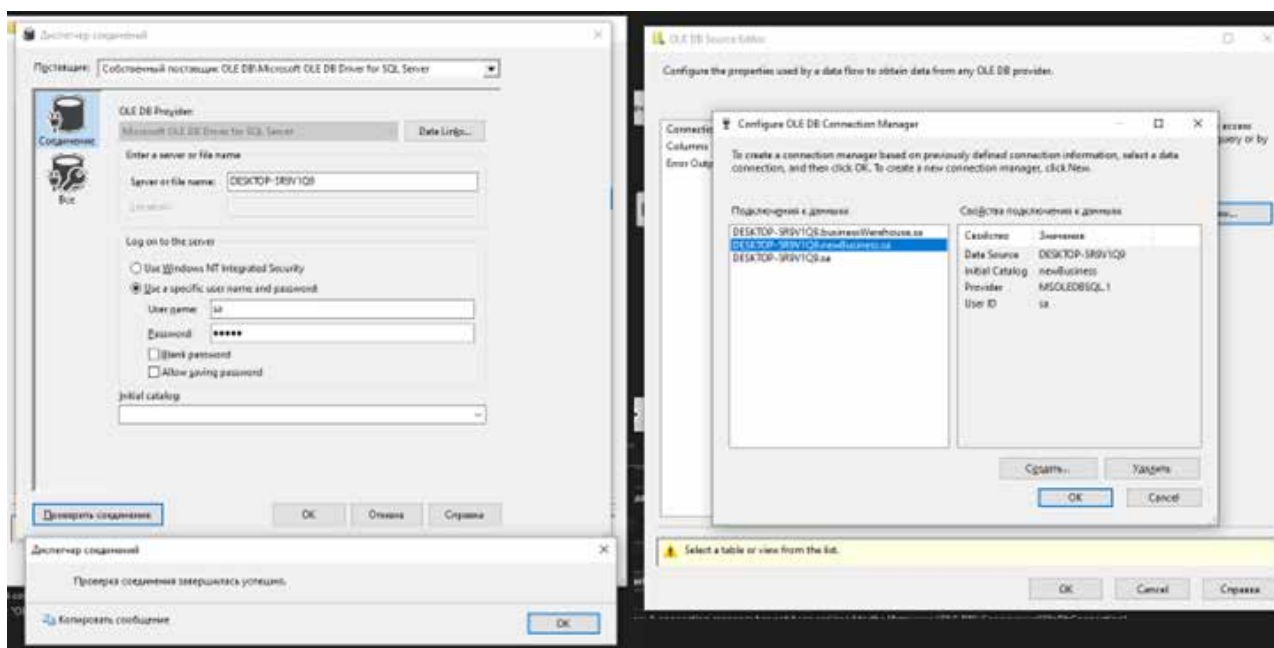


Рисунок 3.11 Встановлення зв'язку

Додаємо до проекту завдання потоку даних. Для коректної інтеграції необхідно реалізувати потоки даних в декілька рівнів від батьківських таблиць до дочірніх. Відповідно до структури сховища, необхідно реалізувати 2 рівні потоків даних:

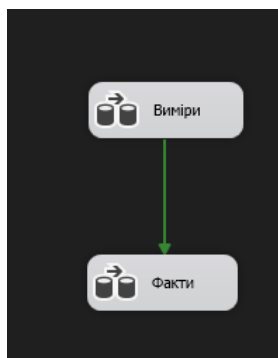


Рисунок 3.12 Рівні потоків даних

На першому рівні було заповнено таблиці виміри, на другому – таблицю фактів (рис. 3.13).

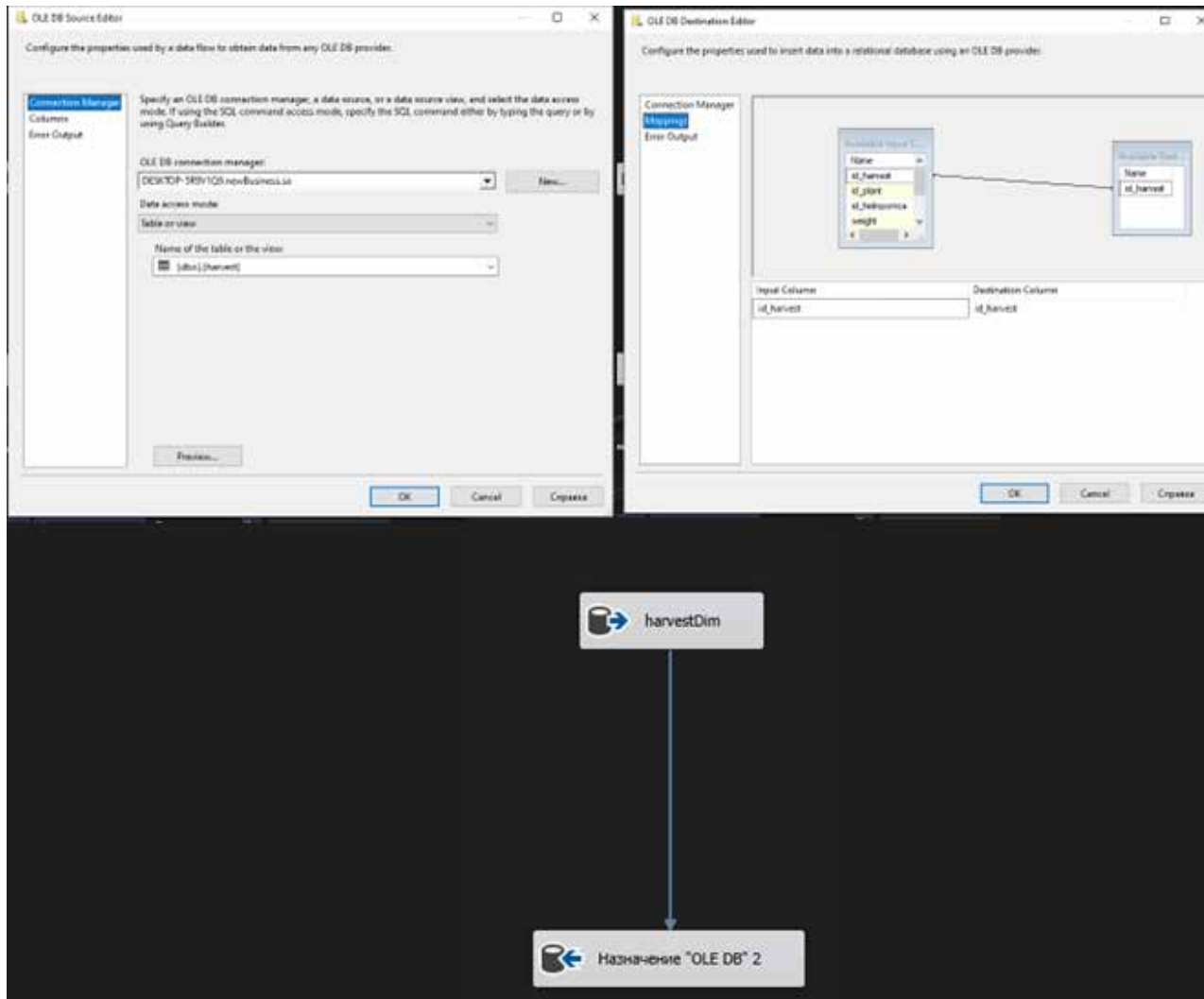


Рисунок 3.13 Заповнення таблиці виміру

Для заповнення таблиці DateDim був написаний sql запит який бере дані з накладних та розділяє дату на рік та місяць. Також додатково потрібно було використати сортування для уникнення повторів дат:

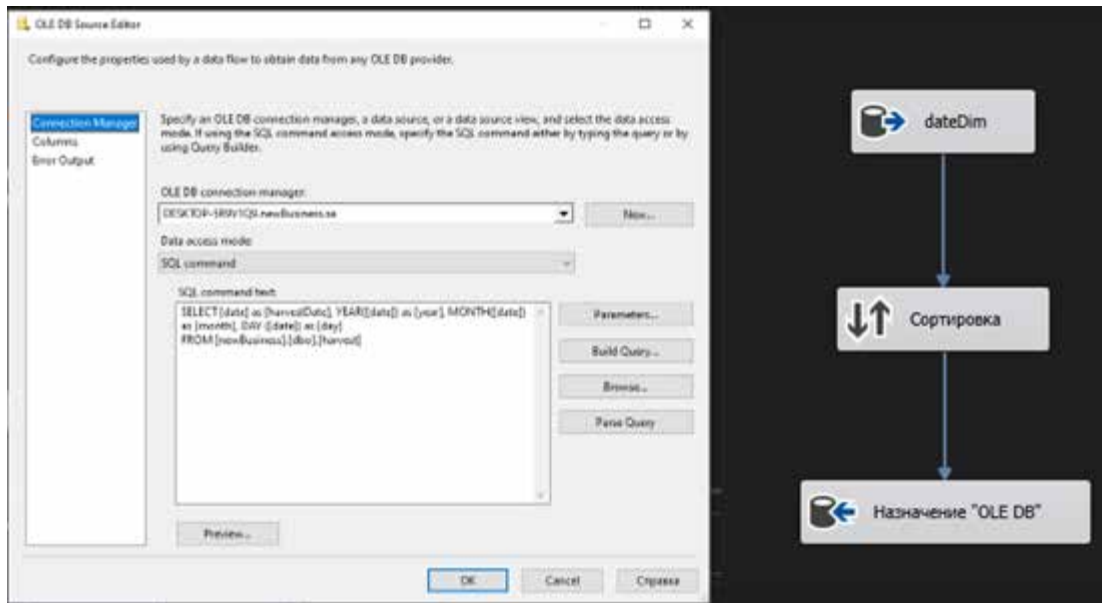


Рисунок 3.13 Заповнення таблиці Date Dim

Для заповнення таблиці фактів також був написаний деякий код:

```

SELECT
  d.id date,
  h.id harvest,
  h.id plant,
  AVG(a.value) as avgAcidity
FROM
  [newBusiness].[dbo].[harvest] h
  JOIN [businessWarehouse].[dbo].[DateDim] d ON h.date = CONVERT(DATE,
  CONCAT(d.vear, '-', d.month, '-', d.day))
  LEFT JOIN newBusiness.dbo.aciditySensor a ON h.id_harvest = a.id_harvest
GROUP BY
  d.id date,
  h.id harvest,
  h.id plant;

```

Рисунок 3.14 Код для заповнення таблиці фактів

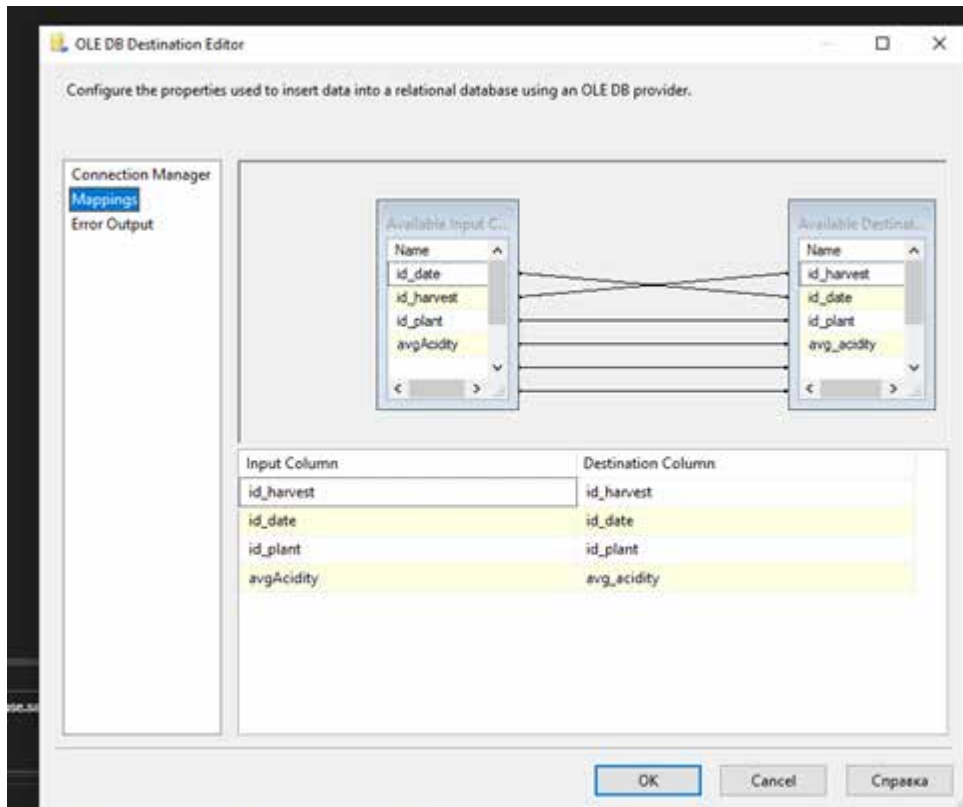


Рисунок 3.15 Мапінг заповнення таблиці фактів

На рис. 3.16 зображено, що всі потоки пройшли успішно після запуску потоків даних:

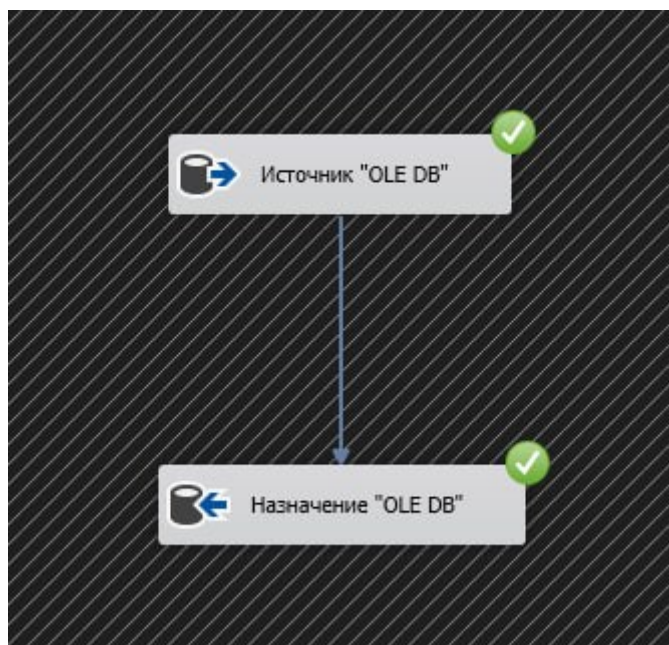


Рисунок 3.16 Успішне проходження потоків даних

3.2 Основи Data Mining

Технологія Data Mining є складовою частиною більш широкої галузі знань - Data Science. Вона включає в себе процес видобування корисної інформації з великого обсягу даних, що зберігаються в різних базах даних.

Основним завданням технології Data Mining є автоматичне виявлення раніше невідомих та цінних залежностей і закономірностей в даних, що дозволяє зробити прогнози та здійснювати стратегічне планування.

Основні методи, які використовуються в технології Data Mining, включають:

- Кластерний аналіз (аналіз кластерів) - групування схожих об'єктів у відповідності з їх характеристиками.
- Класифікаційний аналіз - віднесення об'єктів до певного класу відповідно до їх характеристик.
- Асоціативний аналіз - виявлення зв'язків між різними характеристиками об'єктів в даних.
- Прогнозування - визначення того, що може статися в майбутньому на основі аналізу минулих даних.

У контексті нашого проекту технологія Data Mining буде використовуватися для аналізу даних, що отримуються з сенсорів, що вимірюють показники вирощування салату в гідропоніці. Це дозволить автоматично виявляти залежності між різними факторами, які впливають на якість та врожайність салату, а також зробити прогнози для оптимізації процесу вирощування.

3.3 Аналіз інструментарію для виконання завдань Data Mining

Мова R є однією з найбільш популярних мов для виконання задач Data Mining та аналізу даних. Вона має велику кількість бібліотек та пакетів для роботи з даними, що значно полегшує виконання задач Data Mining та забезпечує високу продуктивність[15].

Також вказано, що використання мови R дозволило використовувати різноманітні алгоритми та методи для вирішення поставлених завдань, зокрема, класифікації, кластеризації, прогнозування, аналізу асоціацій та інших. Всі ці

алгоритми та методи були використані для обробки та аналізу даних про вирощування салату в гідропоніці.

Зазначена мова програмування R є відкритою мовою програмування для статистичних обчислень та графічної візуалізації даних. R є популярним інструментом серед дослідників, статистиків та дата-аналітиків завдяки своїй потужності та гнучкості для аналізу та візуалізації даних.

Основною перевагою використання R є те, що це вільна та відкрита мова програмування з великою кількістю бібліотек та пакетів для виконання різноманітних статистичних та аналітичних завдань. Це дає змогу швидко та ефективно виконувати завдання Data Mining, а також проводити глибинний аналіз даних з метою отримання нових знань та інсайтів.

Крім того, R має велику спільноту користувачів, яка надає безкоштовну підтримку, розвиває бібліотеки та пакети, а також публікує різноманітні матеріали для вивчення мови програмування R.

У R є багато пакетів та бібліотек, що розширюють її функціональність і дають змогу вирішувати різноманітні задачі Data Mining. Наприклад, пакети `dplyr` та `tidyr` використовуються для обробки та трансформації даних, `ggplot2` - для візуалізації даних, а `caret` - для машинного навчання.

Окрім того, мова R має досить простий синтаксис, що дозволяє швидко створювати складні скрипти та програми. Крім того, з R можна легко інтегруватися з іншими інструментами та мовами програмування, наприклад, з Python або SQL.

Таким чином, використання мови R для вирішення задач Data Mining є досить зручним та ефективним підходом, що дозволяє швидко та ефективно аналізувати великі обсяги даних та знаходити нові знання.

3.4 Дані для проведення аналізу

Для нашого дослідження були використані дані з власної невеликої гідропоніки, яка складалась всього з двох місць для посадки. Для аналізу було взято за основу датчик кислотності, так як від цього залежить ріст рослини.

Загалом було отримано багато значень, так як запис робився кожну хвилину протягом трьох тижнів, значення в середньому по посіву є однаковими, так як при підніманні кислотності вмикались насоси і додавались добрива для зменшення кислотності, частина записів зображена на рис. 3.17.

```
SELECT TOP (1000) [id_acidity]
, [id_sensor]
, [id_harvest]
, [value]
, [dateTime]
FROM [newBusiness].[dbo].[aciditySensor]
```

	id_acidity	id_sensor	id_harvest	value	dateTime
1	1	1	1	6,5	NULL
2	2	1	1	6,5	NULL
3	3	1	1	6,5	NULL
4	4	1	1	6,5	NULL
5	5	1	1	6,5	NULL
6	6	1	1	6,5	NULL
7	7	1	1	6,5	NULL
8	8	1	1	6,5	NULL
9	9	1	1	6,5	NULL
10	10	1	1	6,5	NULL
11	11	1	1	6,5	NULL
12	12	1	1	6,5	NULL
13	13	1	1	6,5	NULL
14	14	1	1	6,5	NULL
15	15	1	1	6,5	NULL
16	16	1	1	6,5	NULL
17	17	1	1	6,5	NULL
18	18	1	1	6,5	NULL
19	19	1	1	6,5	NULL
20	20	1	1	6,5	NULL
21	21	1	1	6,5	NULL
22	22	1	1	6,5	NULL
23	23	1	1	6,5	NULL
24	24	1	1	6,5	NULL
25	25	1	1	6,5	NULL
26	26	1	1	6,5	NULL
27	27	1	1	6,5	NULL
28	28	1	1	6,5	NULL

Рисунок 3.17 Дані зчитувані з датчика кислотності

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

4.1 Дослідження використання задач класифікації

4.1.1 Використання 1-Rule для класифікації. 1-Rule (One-Rule) є простим та ефективним алгоритмом класифікації, який використовується в data mining. Його основна ідея полягає у тому, що для кожного класу досліджуваної вибірки будується одне правило класифікації, яке забезпечує максимальну точність класифікації для цього класу[16].

Алгоритм 1-Rule діє наступним чином: для кожного класу вибірки він знаходить найбільш важливий (з точки зору класифікації) атрибут, який можна використовувати для визначення класу. Потім він порівнює значення цього атрибуту з пороговим значенням, яке визначається на основі аналізу значень атрибуту та класу.

1-Rule може бути використаний для класифікації даних з багатьох галузей, включаючи медицину, фінанси, маркетинг та інші. Основні переваги 1-Rule полягають у його простоті та швидкості, що робить його корисним для швидкого аналізу великих обсягів даних.

Загалом, використання 1-Rule для класифікації є ефективним інструментом для розв'язання задач Data Mining. Він дозволяє знайти найбільш важливі атрибути та забезпечити точну класифікацію для кожного класу досліджуваної вибірки.

Крім того, в Data Mining використовуються не тільки алгоритми класифікації, а й інші методи аналізу даних, такі як кластерний аналіз, регресійний аналіз, асоціативний аналіз тощо. Вибір методу залежить від задачі, типу даних та багатьох інших чинників. Тому важливо мати розуміння різноманітних методів та їхніх переваг і недоліків, щоб знати, який метод найкраще використовувати для конкретної задачі.

```
#Завантажуємо дві бібліотеки, щоб отримати доступ до необхідних функцій: odbc та dplyr.
library(odbc)
library(dplyr)

#Підключаємося до бази даних з використанням ODBC драйвера та визначаємо параметри підключення (ім'я сервера, ім'я бази даних, ім'я користувача та пароль).
con <- dbconnect(odbc(),
  driver = "ODBC Driver 17 for SQL Server",
  server = "DESKTOP-5R9V1Q9",
  database = "businesswarehouse",
  uid = "sa",
  pwd = "12345")
```

Рисунок 4.1 Підключення бібліотек і бази даних

Було виконано SQL запит для вибірки всіх даних з таблиці fact (рис. 4.1).

```
query <- "SELECT * FROM businesswarehouse.dbo.fact"
```

Рисунок 4.2 Виконання запиту

Завантажені дані з бази даних в змінну fact (рис. 4.2).

```
fact <- dbgetquery(con, query)
```

Рисунок 4.3 Завантаження даних

Обчислено середнє значення для змінної weight (рис. 4.3).

```
mean <- mean(fact$weight)
```

Рисунок 4.4 Обчислення середнього значення для weight

Проведено класифікацію даних на основі значень змінної weight. Якщо значення змінної більше за середнє значення, то класифікуємо його як "H" (високе значення), в іншому випадку - як "L" (низьке значення) (рис. 4.5).

```
n <- nrow(fact)
results <- character(n)
for (i in 1:n) {
  if (fact[i, 5] > mean) {
    results[i] <- "H"
  } else {
    results[i] <- "L"
  }
}
```

Рисунок 4.5 Класифікація даних weight

Додано стовпець level до таблиці fact, який містить результати класифікації та вивели дані.

```

> #Виводимо перші 6 рядків таблиці fact.
> head(fact)
  id_harvest id_date id_plant avg_acidity weight level
1          1         316         1         6.5 331.20    L
2          2         317         1         6.6 342.24    L
3          3         318         1         6.7 375.36    H
4          4         319         1         6.8 390.08    H
5          5         320         1         6.9 382.72    H
6          6         321         1         7.0 371.68    H
> #Виводимо останні 6 рядків таблиці fact.
> tail(fact)
  id_harvest id_date id_plant avg_acidity weight level
10          10         325         1         6.7 371.68    H
11          11         326         1         6.8 393.76    H
12          12         327         1         6.9 382.72    H
13          13         328         1         7.0 368.00    H
14          14         329         1         7.1 360.64    L
15          15         330         1         7.2 345.92    L

```

Рисунок 4.6 Дані після класифікації

Визначено незалежну змінну `independent_variable` (стовпець `avg_acidity`) та класову змінну `class_variable` (стовпець `level`) (рис. 4.7).

```

independent_variable <- fact$avg_acidity
class_variable <- fact$level

```

Рисунок 4.7 Визначення незалежної та класової змінних

Створено нову таблицю `data`, яка містить тільки незалежну та класову змінні (рис. 4.8).

```

data <- data.frame(independent_variable, class_variable)

```

Рисунок 4.8 Створення таблиці `data`

Було реалізовано алгоритм 1-Rule, який полягає у визначенні найчастішого класу для кожного значення незалежної змінної `independent_variable`. Для цього були згруповано дані за значенням незалежної змінної та обчислено найчастіший клас (рис. 4.9).

```

rule <- data %>%
  group_by(independent_variable) %>%
  summarise(most_frequent_class = names(sort(-table(class_variable))[1]))

```

Рисунок 4.9 Реалізація алгоритму

Була підрахована кількість спостережень для кожного правила, використовуючи функції `inner_join` та `summarise` та підрахована відносна частота кожного класу для кожного значень (рис. 4.10).

```

#підраховуємо кількість спостережень для кожного правила, використовуючи функції inner_join та summarise
rule_count <- data %>%
  inner_join(rule, by = "independent_variable") %>%
  group_by(independent_variable, most_frequent_class) %>%
  summarise(count = sum(class_variable == most_frequent_class))

#підраховуємо відносну частоту кожного класу для кожного значень
class_prob <- rule_count %>%
  group_by(independent_variable) %>%
  mutate(total_count = sum(data$independent_variable == independent_variable),
         class_probability = count / total_count)

print(class_prob)

```

Рисунок 4.10 Підрахунок кількості спостережень

Результат виконаних кроків зображено на рис. 4.11:

independent_variable	most_frequent_class	count	total_count	class_probability
<dbl>	<chr>	<int>	<int>	<dbl>
6.5	L	1	1	1
6.6	L	2	2	1
6.7	H	2	2	1
6.8	H	2	2	1
6.9	H	2	2	1
7	H	2	2	1
7.1	L	2	2	1
7.2	L	2	2	1

Рисунок 4.11 Результат 1-Rule

За даними видно, що `independent_variable` має вісім різних значень: L, L, H, H, H, L, L. Значення «L» означає що наш врожай при певному рівні менший за середнє значення і нам не вигідно в майбутньому підтримувати рівень кислотності на тому рівні, а саме на рівні 6.5, 6.6, 7,1 і 7,2. Значення «H» означає що наш врожай більший за середнє значення і нам вигідно підтримувати рівень кислотності на тому рівні, а саме на рівні 6.7, 6.8, 6.9 і 7.0. Можна зробити висновок, що проміжок найкращого врожаю стоїть з 6.6 до 7.0 і нема сенсу пробувати збільшити або зменшити кислотність вище або нижче цих рівнів, але є сенс спробувати проекспериментувати з кислотністю між цих рівнів для пошуку найбільш вигіднішого результату.

4.2 Дослідження використання методу асоціативних правил

Метод асоціативних правил є одним з найпопулярніших методів дослідження даних в галузі машинного навчання та аналізу бізнес-даних. Його застосовують для виявлення взаємозв'язків між різними елементами в наборі даних, таких як покупки товарів у магазині чи відвідування сторінок на веб-сайті[17].

Метод асоціативних правил базується на пошуку статистично значущих зв'язків між різними елементами набору даних. Наприклад, якщо багато покупців, які купують хліб, також купують молоко, то можна стверджувати, що між цими двома продуктами існує асоціація. Цей метод використовують для виявлення таких зв'язків в більш складних наборах даних.

Дослідження використання методу асоціативних правил дозволяє зрозуміти, які елементи набору даних мають схильність взаємодіяти між собою та які з них можуть бути використані для побудови стратегій бізнесу. Наприклад, якщо веб-сайт зазначає, що багато користувачів, які відвідують сторінку товару А, також переходять на сторінку товару В, то можна рекомендувати товар В користувачам, які переглядають товар А.

Отже, дослідження використання методу асоціативних правил може бути корисним інструментом для розуміння взаємозв'язків в даних та для побудови ефективних стратегій бізнесу.

Розглянемо приклад на основі вказаного вище сховища даних та з використанням мови програмування R.

Було встановлено необхідні бібліотеки: `odbc`, `dplyr`, `arules` та `arulesViz` (рис. 4.12).

```
library(odbc)
library(dplyr)
library(arules)
library(arulesviz)
```

Рисунок 4.12 Встановлення необхідних бібліотек

Встановлено з'єднання з базою даних "businessWarehouse" на сервері "DESKTOP-5R9V1Q9" з використанням драйвера ODBC і зазначеними ідентифікатором користувача і паролем (рис. 4.13).

```
con <- dbConnect(odbc(),
  Driver = "ODBC Driver 17 for SQL Server",
  Server = "DESKTOP-5R9V1Q9",
  Database = "businesswarehouse",
  UID = "sa",
  PWD = "12345")
```

Рисунок 4.13 Підключення бази даних

Задано SQL-запит на вибірку всіх даних з таблиці "fact" бази даних "businessWarehouse" (рис. 4.14).

```
query <- "SELECT * FROM businesswarehouse.dbo.fact"
```

Рисунок 4.14 Виконання запиту

Виконано запит на отримання даних з бази даних за допомогою зазначеного з'єднання і збережено результати в змінній fact (рис. 4.15).

```
fact <- dbGetQuery(con, query)
```

Рисунок 4.15 Завантаження даних

Обчислено середнє значення ваги зі змінної weight в fact і збережено результат в змінній mean (рис. 4.16).

```
mean <- mean(fact$weight)
```

Рисунок 4.16 Обчислення середнього значення для weight

Встановлено кількість рядків в fact і збережено результат в змінній n (рис. 4.17).

```
n <- nrow(fact)
```

Рисунок 4.17 Рахування кількості рядків в fact

Було створено вектор results довжиною n, що заповнений порожніми рядками та заповнено вектор results згідно з умовою: якщо значення в стовпці 5 (weight) більше за mean, то присвоєно значення "H", інакше - "L" (рис. 4.18).

```
results <- character(n)
for (i in 1:n) {
  if (fact[i, 5] > mean) {
    results[i] <- "H"
  } else {
    results[i] <- "L"
  }
}
```

Рисунок 4.18 Класифікація даних weight

Було додано новий стовпець з назвою "level" до таблиці fact і заповнено його значеннями з вектору results (рис. 4.19).

```
fact$level <- results
```

Рисунок 4.19 Заповнення нового стовпця level з значеннями класифікації

Створено новий `data.frame` з 2 стовпцями: `avg_acidity` і `level`, в якому дані беруться зі стовпців `fact$avg_acidity` і `fact$level` (рис. 4.20).

```
data <- data.frame(fact$avg_acidity, fact$level)
```

Рисунок 4.20 Створення `data.frame` з стовпцями `avg_acidity` і `level`

Було застосовано алгоритм Apriori до даних `data` з параметрами `supp = 0.1` і `conf = 0.5` для знаходження асоціаційних правил між елементами (рис.4 .21).

```
rules <- apriori(data, parameter = list(supp = 0.1, conf = 0.5))
```

Рисунок 4.21 Знаходження асоціаційних правил

Виведено у консоль список асоціаційних правил, знайдених за допомогою алгоритму Apriori (рис. 4.22).

lhs	rhs	support	confidence	coverage	lift	count
[1] {}	=> {fact.level=H}	0.5333333	0.5333333	1.0000000	1.000000	8
[2] {fact.avg_acidity=[6.77,7]}	=> {fact.level=H}	0.2666667	1.0000000	0.2666667	1.875000	4
[3] {fact.level=H}	=> {fact.avg_acidity=[6.77,7]}	0.2666667	0.5000000	0.5333333	1.875000	4
[4] {fact.avg_acidity=[6.5,6.77]}	=> {fact.level=L}	0.2000000	0.6000000	0.3333333	1.285714	3
[5] {fact.avg_acidity=[7,7.2]}	=> {fact.level=L}	0.2666667	0.6666667	0.4000000	1.428571	4
[6] {fact.level=L}	=> {fact.avg_acidity=[7,7.2]}	0.2666667	0.5714286	0.4666667	1.428571	4

Рисунок 4.22 Асоціаційні правила

Створено графи асоціаційних правил для значень «L» і «H» (рис. 4.23).

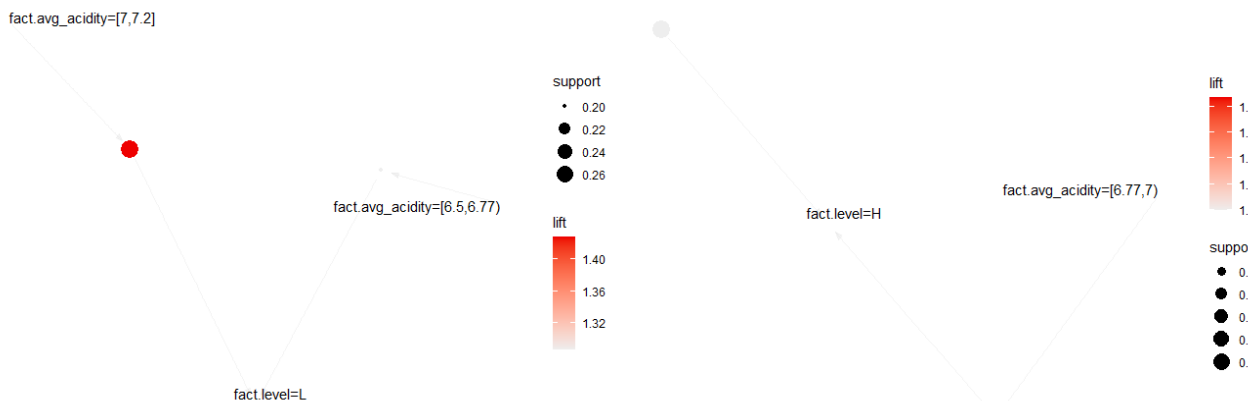


Рисунок 4.23 Графи асоціаційних правил

На графах (рис. 4.23) можна побачити який рівень кислотності варто використовувати для вирощування салату і який не варто. На першому графу з значенням «L» показано рівні 6.5, 6.77, які не варто використовувати, тому що врожайність буде низькою. На другому графу з значенням «H» показано рівні 6.77, 7, які варто використовувати, тому що врожайність буде великою.

4.3 Дослідження використання алгоритмів кластеризації

Дослідження використання алгоритмів кластеризації є важливим напрямком в галузі машинного навчання. Кластеризація дозволяє групувати об'єкти за спільними ознаками без попереднього знання про кількість кластерів та їх характеристики[18].

Одним з основних завдань кластеризації є зменшення кількості даних до більш компактної форми, яка дозволяє полегшити подальші аналітичні операції з даними. Кластеризація також може використовуватися для побудови рекомендаційних систем, аналізу соціальних мереж, аналізу зображень та багатьох інших завдань.

Для розв'язання завдання кластеризації існує багато різних алгоритмів, таких як k-means, DBSCAN, hierarchical clustering та інші. Кожен з них має свої переваги та недоліки, які варто враховувати при виборі алгоритму для конкретного завдання.

Один з основних параметрів, який впливає на результат кластеризації - це кількість кластерів. Для визначення оптимальної кількості кластерів можна використовувати різні підходи, наприклад, метод "ліктя" (elbow method) або метод "середньої відстані" (average silhouette method).

Метод "ліктя" полягає у визначенні значення суми квадратів відстаней між точками і центрами їхніх кластерів для різної кількості кластерів. Графік залежності цієї суми від кількості кластерів схожий на зігнуту ліктьову лінію. Оптимальна кількість кластерів визначається на тому місці, де зменшення суми квадратів відстаней перестає бути значним.

Розглянемо приклад на основі вказаного вище сховища даних та з використанням мови програмування R.

Спочатку було завантажено необхідні пакети R (рис. 4.24).

```
library(odbc) # завантаження бібліотеки для роботи з базами даних
library(dplyr) # завантаження бібліотеки для роботи з даними
library(stats) # завантаження бібліотеки для статистичного аналізу
```

Рисунок 4.24 Встановлення необхідних бібліотек

Створено dataframe з даними, необхідними для кластеризації (рис. 4.25).

```
x <- data.frame(fact$weight, fact$avg_acidity)
```

Рисунок 4.25 Створення dataframe з даними для кластеризації

Створено вектору "k.values" з числами від 1 до 10, які представляють кількість кластерів, які будуть використовуватись під час кластеризації (рис. 4.26).

```
k.values <- 1:10 # створення вектору із кількості кластерів
```

Рисунок 4.26 Створення вектору

Створено вектору "wss_values", який буде містити внутрішньокластерні відстані для кожної кількості кластерів (рис. 4.27).

```
wss_values <- vector(length = length(k.values))
```

Рисунок 4.27 Створення вектору з внутрішньокластерними відстані

Далі було реалізовано процес кластеризації методом k-середніх з використанням методу “ліктя” для визначення оптимальної кількості кластерів (рис. 4.28).

```
for(i in k.values) {
  kmeans_result <- kmeans(x, centers = i) # виконання кластеризації методом K-means
  wss_values[i] <- kmeans_result$tot.withinss # зберігання внутрішньокластерної відстані для поточної кількості кластерів
}
```

Рисунок 4.28 Процес кластеризації

Був створений графік, який відображає залежність внутрішньокластерної відстані від кількості кластерів (рис. 4.29).

```
plot(k.values, wss_values, type = "b", pch = 19, frame = FALSE,
     xlab = "Number of clusters k", ylab = "Total within-clusters sum of squares")
```

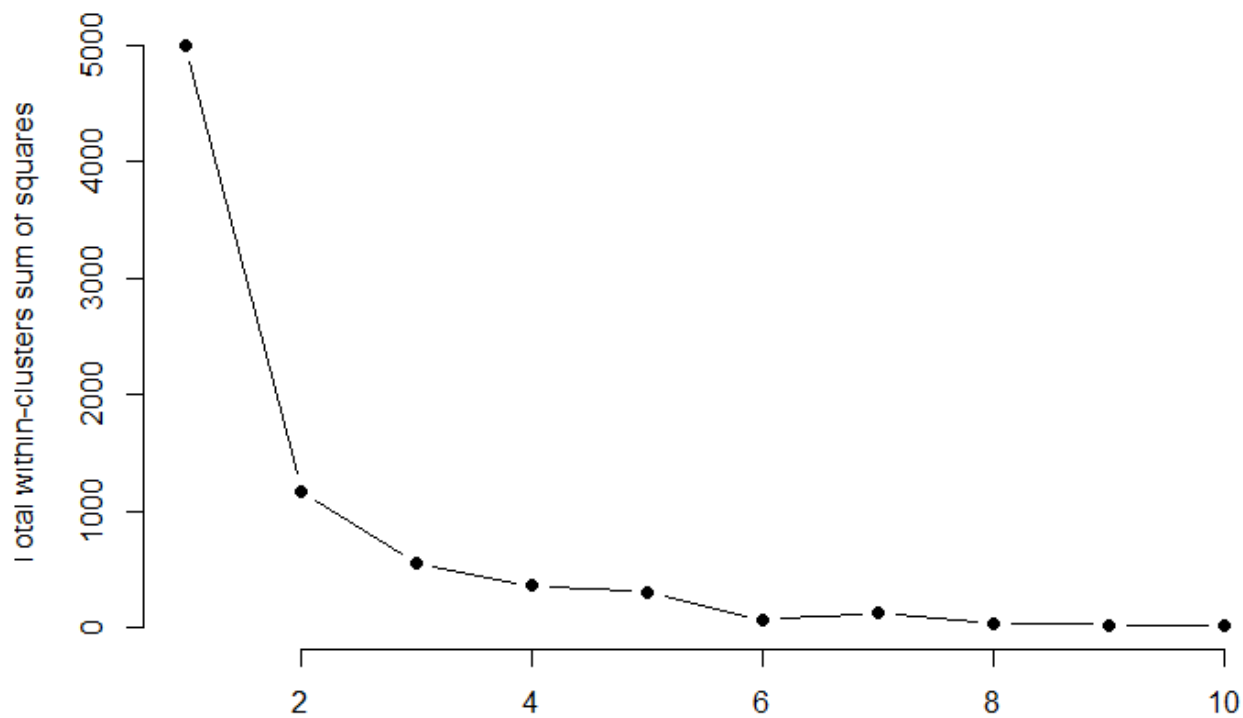


Рисунок 4.29 Граф з залежністю внутрішньокластерної відстані від кількості кластерів

Обчислені відстані між внутрішньокластерними відстанями (рис. 4.30).

```
dist <- wss_values[2:length(wss_values)] - wss_values[1:(length(wss_values)-1)]
```

Рисунок 4.30 Обчислення відстані між внутрішньокластерними відстанями

Створення графіку, який відображає залежність різниці внутрішньокластерних відстаней від кількості кластерів (рис. 4.31).

```
plot(k.values[-1], dist, type = "b", pch = 19, frame = FALSE,
     xlab = "Number of clusters K",
     ylab = "Difference in within-cluster sum of squares")
```

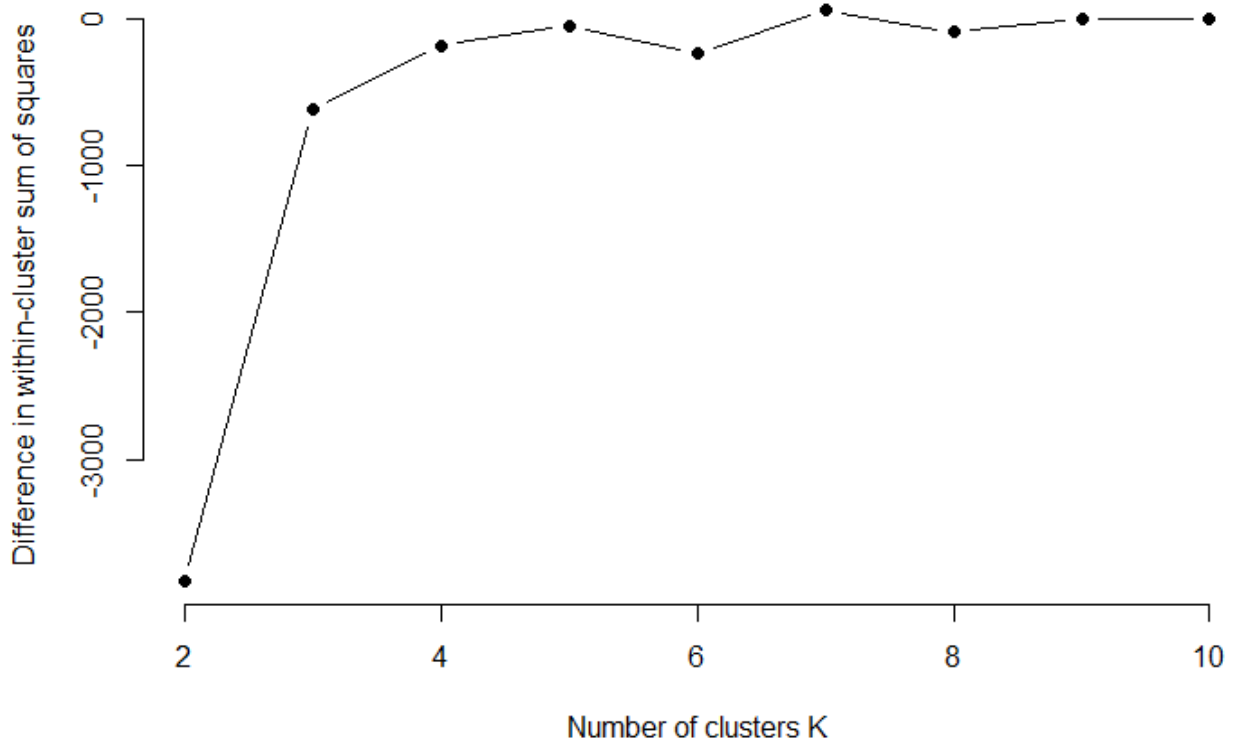


Рисунок 4.31 Графік з залежностями різниці внутрішньокластерних відстаней від кількості кластерів

Повернуто індекс елемента з найбільшим значенням у векторі `dist` (рис. 4.32).

```
kmeans_result <- kmeans(x, centers=m1)#застосування алгоритму k-середніх для знайденої оптимальної кількості кластерів
cat("Optimal number of clusters:", m1)
optimal number of clusters: 7
```

Рисунок 4.32 Індекс елемента з найбільшим значенням у векторі `dist`

Було застосовано алгоритму `k`-середніх для знайденої оптимальної кількості кластерів (рис. 4.33).

```

kmeans_result <- kmeans(x, centers=mi)
print(kmeans_result)
k-means clustering with 7 clusters of sizes 2, 4, 1, 2, 4, 1, 1

Cluster means:
  fact.weight fact.avg_acidity
1      358.80           7.10
2      345.92           6.90
3      331.20           6.50
4      382.72           6.90
5      371.68           6.85
6      390.08           6.80
7      393.76           6.80

Clustering vector:
 [1] 3 2 5 6 4 5 1 2 2 5 7 4 5 1 2

within cluster sum of squares by cluster:
 [1]  6.7712 27.4448  0.0000  0.0000 27.1748  0.0000  0.0000
 (between_ss / total_ss = 98.8 %)

```

Рисунок 4.33 Дані з алгоритму k-середніх для знайденої оптимальної кількості кластерів

Було побудовано графік, де кожен зразок (точка) зображається кольором, який відповідає його кластеру та додавання точок, які представляють центри кластерів. Кожен центр має відмінну форму та кольори від кожного кластеру (рис. 4.34).

```
plot(x, col=kmeans_result$cluster, pch=20)
points(kmeans_result$centers, col=1:3, pch=23, cex=3)
```

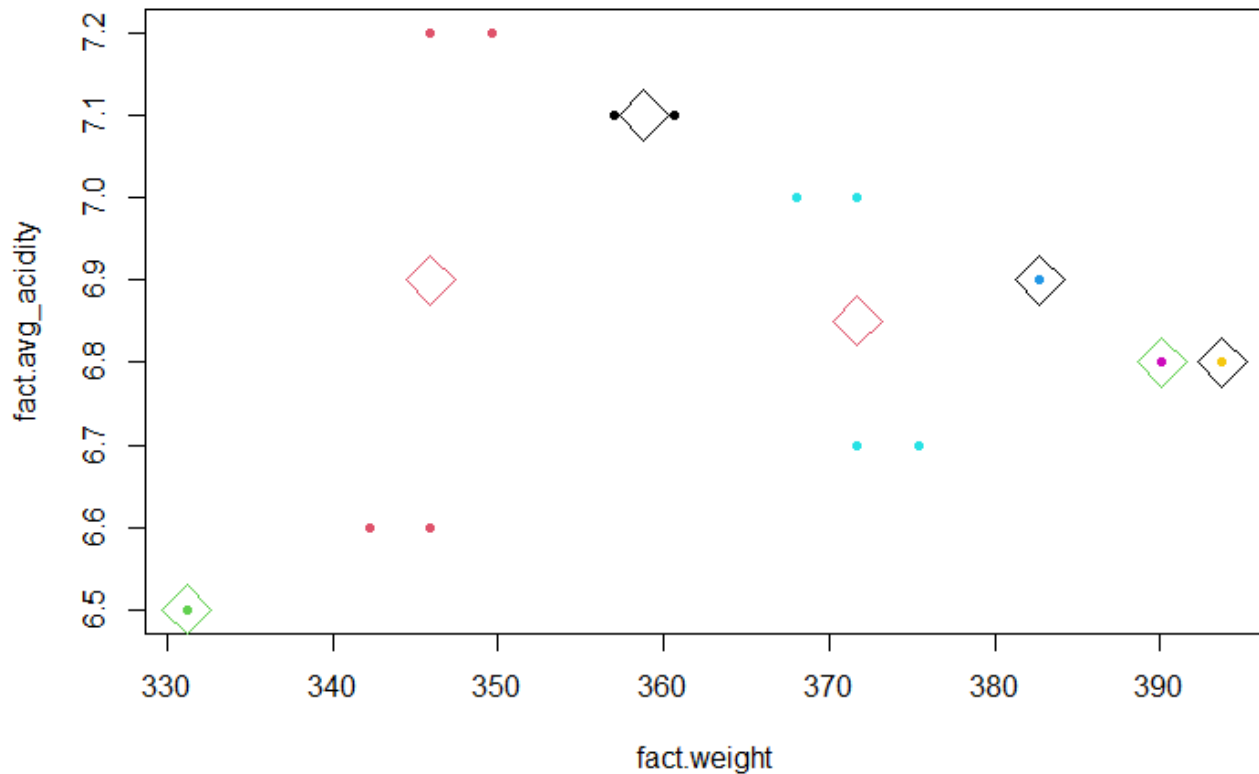


Рисунок 4.34 Граф з кластерами

За результатами кластеризації було сформовано 7 кластерів різної кількості спостережень. Можна побачити, що кластери мають різні середні значення для двох змінних - «fact.weight» і «fact.avg_acidity».

В цілому, можна стверджувати, що кластеризація була досить успішною, тому що можна виділити підгрупи спостережень зі схожими характеристиками, що може бути корисно для подальшого аналізу даних.

ВИСНОВКИ

Отже, використання data mining є важливим інструментом для аналізу різноманітних систем. При належному використанні ця технологія може допомогти підприємствам забезпечити належний рівень ефективності та конкурентоздатності на ринку. Проте, для успішного впровадження необхідно мати якісні дані та належні системи. Ретельне планування та реалізація процесу збору та збереження даних є ключовим фактором для успішного використання data mining у різних сферах, включаючи гідропоніку.

Завдяки аналізу цих даних, ми виявили багато корисної інформації. Ми здійснили вимірювання та визначили оптимальні умови для салату. Цей аналіз дозволяє нам краще розуміти вимоги кожного виду рослин і вчасно реагувати на будь-які аномалії.

Також, ми виявили кореляції між різними параметрами та зрозуміли, як вони впливають на ріст та врожайність рослин. Наприклад, ми встановили, що рівень кислотності води в гідропонічній системі досить сильно впливає на ріст та врожайність певних видів рослин.

Дослідження, які ми провели, мають велике значення для нашого гідропонічного проекту. Вони дозволили нам ліпше розуміти потреби рослин та виробничі умови, які найбільш підходять для їхнього росту та розвитку. Ми можемо вдосконалити умови вирощування кожного виду рослин та підтримувати їх на оптимальному рівні.

З наших досліджень випливає, що важливість ведення бази даних та аналізу даних в гідропоніці важлива для досягнення найкращих результатів. Рекомендуємо продовжувати збирати дані та вдосконалювати аналіз, щоб підтримувати найоптимальніші умови для рослин.

У майбутньому, можливо, варто розширити функціональність бази даних, додавши можливість передбачати оптимальні умови для кожного виду рослин на підставі зібраних даних та інтегрувати систему з автоматизованими процесами управління гідропонічними системами.

В цілому, наші дослідження дозволяють нам підтримувати та вдосконалювати наш гідропонічний проект, забезпечуючи оптимальні умови для росту рослин та збільшення врожайності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Howard M. Resh (2018) "Hydroponic Food Production: A Definitive Guidebook for the Advanced Home Gardener and the Commercial Hydroponic Grower" (CRC Press)
- [2] Daniel Schwartz (2017) "Hydroponic Gardening: A Practical Guide to Gaining Food Independence" (CreateSpace Independent Publishing Platform)
- [3] Jeffrey Winterborne (2018) "Hydroponics: A Comprehensive Guide to Hydroponics" (Independently published)
- [4] George F. Van Patten (2018) "Hydroponic Basics by George F. Van Patten" (Van Patten Publishing)
- [5] Scott Matson (2017) "DIY Hydroponic Gardens: How to Design and Build an Inexpensive System for Growing Plants in Water" (Cool Springs Press)
- [6] Jack Ross (2019) "Hydroponics for Beginners: The Ultimate Hydroponics Crash Course Guide: Master Hydroponics at Home in Less Than 24 Hours" (Independently published)
- [7] Simon Hamilton (2020) "The Complete Hydroponic Garden Book: A Simple Step by Step Guide to Hydroponic Gardening" (Independently published)
- [8] Thomas Carter (2019) "Hydroponics: The Beginner's Guide to Building an Efficient Hydroponic System" (Independently published)
- [9] Calvin Hale (2018) "Hydroponics: The Ultimate Step-by-Step Guide to Effective Home Gardening" (CreateSpace Independent Publishing Platform)
- [10] Mark Thoma (2019) "Hydroponics for Beginners: The Complete Step-by-Step Guide to Self-Produce Your Flavorful Vegetables and Herbs in Your Own Vegetable Garden" (Independently published)
- [11] Christopher Selby (2020) "Hydroponics: A Comprehensive Guide to Hydroponics for Beginners" (Independently published)
- [12] Dan Woods (2017) "Hydroponics: The Ultimate Step-by-Step Guide to Effective Home Gardening" (CreateSpace Independent Publishing Platform)

- [13] Jiawei Han, Micheline Kamber, and Jian Pei (2011) "Data Mining: Concepts and Techniques" (Morgan Kaufmann)
- [14] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar (2018) "Introduction to Data Mining" (Pearson)
- [15] Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal (2016) "Data Mining: Practical Machine Learning Tools and Techniques" (Morgan Kaufmann)
- [16] Christopher J. Merz and Ashwin Srinivasan (2016) "1-Rule Classification: A Review" (Morgan Kaufmann)
- [17] Johannes Fürnkranz and Eyke Hüllermeier (2018) "Preference Learning" (Springer)
- [18] Charu C. Aggarwal (2013) "Data Clustering: Algorithms and Applications" (Chapman and Hall/CRC)

ДОДАТОК А

КОД РЕАЛІЗАЦІЯ АЛГОРИТМУ 1-RULE


```
library(odbc)
```

```
library(dplyr)
```

```
library(arules)
```

```
library(arulesViz)
```

```
#встановлює з'єднання з базою даних "businessWarehouse" на сервері  
"DESKTOP-5R9V1Q9" з використанням драйвера ODBC і зазначеними  
ідентифікатором користувача і паролем.
```

```
con <- dbConnect(odbc(),
```

```
  Driver = "ODBC Driver 17 for SQL Server",
```

```
  Server = "DESKTOP-5R9V1Q9",
```

```
  Database = "businessWarehouse",
```

```
  UID = "sa",
```

```
  PWD = "12345")
```

```
#задає SQL-запит на вибірку всіх даних з таблиці "fact" бази даних  
"businessWarehouse".
```

```
query <- "SELECT * FROM businessWarehouse.dbo.fact"
```

```
#виконує запит на отримання даних з бази даних за допомогою зазначеного  
з'єднання і зберігає результати в змінній fact.
```

```
fact <- dbGetQuery(con, query)
```

```
#обчислює середнє значення ваги зі змінної weight в fact і зберігає результат в  
змінній mean.
```

```
mean <- mean(fact$weight)
```

```
#встановлює кількість рядків в fact і зберігає результат в змінній n.
```

```
n <- nrow(fact)

#створює вектор results довжиною n, що заповнений порожніми рядками.
results <- character(n)
for (i in 1:n) {
  if (fact[i, 5] > mean) {
    results[i] <- "H"
  } else {
    results[i] <- "L"
  }
}

#Додає новий стовпець з назвою "level" до таблиці fact і заповнює його
значеннями з вектору results.
fact$level <- results

#Створює новий data.frame з трьома стовпцями: id_plant, id_harvest і level, в
якому дані беруться зі стовпців fact$id_plant, fact$id_harvest і fact$level.
data <- data.frame(fact$avg_acidity, fact$level)

#Застосовуємо алгоритм Apriori до даних data з параметрами supp = 0.1 і conf =
0.5 для знаходження асоціаційних правил між елементами.
rules <- apriori(data, parameter = list(supp = 0.1, conf = 0.5))

#Виводить у консоль список асоціаційних правил, знайдених за допомогою
алгоритму Apriori.
inspect(rules)

#Вибирає підмножину асоціаційних правил зі значенням правої частини
"fact.level=L" і зберігає їх у нову змінну rulesDem.
```



```
rulesDem <- subset(rules, subset = rhs %in% "fact.level=L")
```

#Виводить у консоль перші 10 асоціаційних правил зі значенням підтримки, які були вибрані на попередньому кроці.

```
inspect(head(rulesDem, n = 10, by = "support"))
```

#Створює граф асоціаційних правил з першими 10 правилами зі значенням підтримки та налаштуваннями, які вказані в control.

```
graph <- plot(head(sort(rulesDem, by = "support"), 10), method = "graph",
              control = list(nodeCol = grey.colors(10),
                             edgeCol = grey(.7), alpha = 1))
```

#Відображає граф, створений на попередньому кроці.

```
plot(graph)
```

#Вибирає підмножину асоціаційних правил зі значенням правої частини "fact.level=H" і зберігає їх у нову змінну rulesDem.

```
rulesDem <- subset(rules, subset = rhs %in% "fact.level=H")
```

#Виводить у консоль перші 10 асоціаційних правил зі значенням підтримки, які були вибрані на попередньому кроці.

```
inspect(head(rulesDem, n = 10, by = "support"))
```

#Створює граф асоціаційних правил з першими 10 правилами зі значенням підтримки та налаштуваннями, які вказані в control.

```
graph <- plot(head(sort(rulesDem, by = "support"), 10), method = "graph",
              control = list(nodeCol = grey.colors(10),
                             edgeCol = grey(.7), alpha = 1))
```

#Відображає граф, створений на попередньому кроці.

```
plot(graph)
```

#Закриває з'єднання з базою даних.

dbDisconnect(con)

ДОДАТОК Б**КОД РЕАЛІЗАЦІЯ АЛГОРИТМУ КЛАСТЕРИЗАЦІЇ**


```
library(odbc) # завантаження бібліотеки для роботи з базами даних
library(dplyr) # завантаження бібліотеки для роботи з даними
library(stats) # завантаження бібліотеки для статистичного аналізу

con <- dbConnect(odbc(),
  Driver = "ODBC Driver 17 for SQL Server",
  Server = "DESKTOP-5R9V1Q9",
  Database = "businessWarehouse",
  UID = "sa",
  PWD = "12345") # встановлення з'єднання з базою даних

query <- "SELECT * FROM businessWarehouse.dbo.fact"

fact <- dbGetQuery(con, query) # виконання запиту до бази даних та отримання
результатів

x <- data.frame(fact$weight, fact$avg_acidity) # створення dataframe з даними,
необхідними для кластеризації

k.values <- 1:10 # створення вектору із кількості кластерів

wss_values <- vector(length = length(k.values)) # створення вектору для зберігання
внутрішньокластерних відстаней

for(i in k.values) {
  kmeans_result <- kmeans(x, centers = i) # виконання кластеризації методом K-
means
  wss_values[i] <- kmeans_result$tot.withinss # зберігання внутрішньокластерної
відстані для поточної кількості кластерів
```

```
}
```

```
plot(k.values, wss_values, type = "b", pch = 19, frame = FALSE,
```

```
  xlab = "Number of clusters K", ylab = "Total within-clusters sum of squares") #  
створення графіку, який відображає залежність внутрішньокластерної відстані  
від кількості кластерів
```

```
dist <- wss_values[2:length(wss_values)] - wss_values[1:(length(wss_values)-1)] #  
обчислення відстані між внутрішньокластерними відстанями
```

```
plot(k.values[-1], dist, type = "b", pch = 19, frame = FALSE,
```

```
  xlab = "Number of clusters K",  
  ylab = "Difference in within-cluster sum of squares") # створення графіку, який  
відображає залежність різниці внутрішньокластерних відстаней від кількості  
кластерів
```

```
mi <- which.max(dist) + 1 #повертає індекс елемента з найбільшим значенням у  
векторі dist
```

```
cat("Optimal number of clusters:", mi)# функція виводу на консоль. Виводить текст  
"Optimal number of clusters:" та оптимальне число кластерів
```

```
kmeans_result <- kmeans(x, centers=mi)#застосування алгоритму k-середніх для  
знайденої оптимальної кількості кластерів
```

```
print(kmeans_result)#виведення результатів кластеризації
```

```
plot(x, col=kmeans_result$cluster, pch=20)#побудова графіку, де кожен зразок  
(точка) зображається коліром, який відповідає його кластеру
```

```
points(kmeans_result$centers, col=1:3, pch=23, sех=3)#додавання точок, які  
представляють центри кластерів. Кожен центр має відмінну форму та кольори  
від кожного кластеру.
```

`dbDisconnect(con)`#відключення від бази даних.