

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

**ПОГОДЖЕНО**

**Декан факультету  
інформаційних технологій**  
(назва факультету)

\_\_\_\_\_ Ігор Болбот \_\_\_\_\_  
(підпис) (ім'я ПРІЗВИЩЕ)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2025 р.

**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ**

**Завідувач кафедри  
комп'ютерних наук**  
(назва кафедри)

\_\_\_\_\_ Белла Голуб \_\_\_\_\_  
(підпис) (ім'я ПРІЗВИЩЕ)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2025 р.

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

**на тему**

Система моніторингу параметрів мікроклімату у промисловому пташнику

Спеціальність

\_\_\_\_\_ 122 «Комп'ютерні науки» \_\_\_\_\_  
(код і найменування)

Освітня програма

\_\_\_\_\_ Інформаційні управляючі системи та технології \_\_\_\_\_  
(назва)

Орієнтація освітньої програми

\_\_\_\_\_ Освітньо-професійна \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)

**Гарант освітньої програми**

\_\_\_\_\_ К.Т.Н., доцент \_\_\_\_\_ Белла Голуб \_\_\_\_\_  
(науковий ступінь та вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)

**Керівник магістерської кваліфікаційної роботи**

\_\_\_\_\_ Д.Т.Н., професор \_\_\_\_\_ Олександр Бушма \_\_\_\_\_  
(науковий ступінь та вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)

**Виконав**

\_\_\_\_\_ Богдан Бабій \_\_\_\_\_  
(підпис) (ім'я ПРІЗВИЩЕ здобувача)

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

**ЗАТВЕРДЖУЮ**  
**Завідувач кафедри**  
**комп'ютерних наук**

к.т.н., доцент Белла Голуб  
(науковий ступінь, вчене звання) (підпис) (ім'я ПРІЗВИЩЕ)  
"01" Листопада 2024 року

**З А В Д А Н Н Я**

**ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ ЗДОБУВАЧУ**

Бабію Богдану Юрійовичу

(прізвище, ім'я, по батькові)

Спеціальність

122 «Комп'ютерні науки»

(код і найменування)

Освітня програма

Інформаційні управляючі системи та технології

(назва)

Орієнтація освітньої програми

Освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Тема магістерської кваліфікаційної роботи

Система моніторингу параметрів мікроклімату у промисловому пташнику

затверджена наказом від "01 листопада 2024р. №1964 «С»

Термін подання завершеної роботи на кафедру

01.12.2025

(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи

набір даних від підприємства ФГ "УЛАР" (с. Ланівка)

Перелік питань, що підлягають дослідженню:

1. Дослідження можливості застосування OLAP та Data Mining методів для підвищення ефективності роботи пташників.

2. Вплив параметрів мікроклімату на здоров'я та продуктивність птиці.

Перелік графічного матеріалу (за потреби)

Дата видачі завдання "01 листопада 2024 р.

Керівник магістерської кваліфікаційної роботи

( підпис )

Олександр Бушма

(ім'я ПРІЗВИЩЕ)

Завдання прийняв до виконання

Богдан Бабій

(ім'я ПРІЗВИЩЕ)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської кваліфікаційної роботи	Строк виконання етапів магістерської кваліфікаційної роботи	Примітка
1	Видача завдання	01.11.2024	
2	Аналіз предметної області	01.11.2024 – 10.12.2025	
3	Моделювання предметної області	11.12.2024 – 23.01.2025	
4	Розробка системи	24.01.2025 – 18.05.2025	
5	Аналіз результатів	19.05.2025 – 31.08.2025	
6	Оформлення записки	01.09.2025 – 01.11.2025	
7	Постерна сесія	28.10.2025 – 29.10.2025	
8	Перевірка на плагіат	13.11.2025	
9	Попередній захист	01.12.2025	
10	Захист	16.12.2025	

Студент \_\_\_\_\_ Богдан Бабій  
(підпис) (ім'я та прізвище)

Керівник магістерської кваліфікаційної роботи \_\_\_\_\_ Олександр Бушма  
(підпис) (ім'я та прізвище)

## РЕФЕРАТ

Магістерська кваліфікаційна робота присвячена розробці системи моніторингу параметрів мікроклімату у промисловому пташнику.

Об'єктом дослідження є процес вирощування птиці у промисловому пташнику. Метою роботи є розробка та застосування системи аналітичного моніторингу, яка поєднує технології OLAP та методи Data Mining для виявлення закономірностей між параметрами мікроклімату та ефективністю утримання птиці.

У роботі використано багатовимірний аналіз даних OLAP для побудови гіперкуба, агрегування показників та формування звітів, що дозволяє досліджувати зміни параметрів середовища у різних вимірах. Для інтелектуального аналізу застосовано методи Data Mining, а саме алгоритми класифікації 1-Rule та наївний Байєс, алгоритм пошуку асоціативних правил для визначення залежностей між мікрокліматом і втратами, а також кластеризацію K-means для групування сценаріїв стану середовища та виявлення критичних режимів утримання.

Наукова складова роботи полягає у поєднанні сховища даних, OLAP-аналітики та Data Mining у єдиній системі моніторингу мікроклімату, що забезпечує можливість виявлення прихованих закономірностей у великих масивах інформації з різних господарств.

Практична значимість полягає у можливості оперативного контролю параметрів середовища, прогнозування ризиків втрат і підтримки управлінських рішень.

Результати дослідження можуть бути впроваджені у сучасні птахівницькі підприємства для підвищення продуктивності та зниження економічних витрат.

Магістерська кваліфікаційна робота містить 73 сторінок, 34 рисунка, 1 таблицю, 7 додатків та список використаних джерел із 21 найменування.

## **ABSTRACT**

The master's qualification work is devoted to the development of a system for monitoring microclimate parameters in an industrial poultry house.

The object of the study is the process of raising poultry in an industrial poultry house. The aim of the work is to develop and apply an analytical monitoring system that combines OLAP technologies and Data Mining methods to identify patterns between microclimate parameters and the efficiency of poultry keeping.

The work uses multidimensional OLAP data analysis to build a hypercube, aggregate indicators and generate reports, which allows you to study changes in environmental parameters in different dimensions. Data Mining methods are used for intellectual analysis, namely 1-Rule classification algorithms and naive Bayes, an algorithm for searching for associative rules to determine dependencies between microclimate and losses, as well as K-means clustering to group environmental state scenarios and identify critical maintenance modes.

The scientific component of the work consists in combining a data warehouse, OLAP analytics and Data Mining in a single microclimate monitoring system, which provides the ability to identify hidden patterns in large amounts of information from different farms.

The practical significance lies in the possibility of operational control of environmental parameters, forecasting the risks of losses and supporting management decisions.

The results of the study can be implemented in modern poultry enterprises to increase productivity and reduce economic costs.

The master's thesis contains 73 pages, 34 figures, 1 table, 7 appendices, and a list of sources used with 21 titles.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Процес вирощування птиці у промисловому пташнику як об’єкт дослідження.....	8
1.2 Огляд існуючих рішень .....	12
1.3 Огляд наукових праць.....	19
1.4 Постановка завдання.....	21
2 МОДЕЛЮВАННЯ ПРОЦЕСІВ ВИРОЩУВАННЯ ПТИЦІ У ПТАШНИКУ ..	23
2.1 Основні концепції .....	23
2.2 Діаграма прецедентів.....	24
2.3 Діаграма класів.....	27
3 РОЗРОБКА СИСТЕМИ .....	30
3.1 Архітектура системи.....	30
3.2 Опис джерела даних.....	32
3.3 Проектування сховища даних .....	33
3.4 Механізм вилучення, обробки і передачі даних.....	35
3.5 Модуль аналітики даних.....	44
4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ.....	50
4.1 Методи Data Mining .....	50
4.2 Методи OLAP.....	66
ВИСНОВКИ .....	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	71
ДОДАТКИ .....	74

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

OLAP – On–Line Analytical Processing.

Data Mining – Інтелектуальний аналіз даних.

KPI – Key Performance Indicator (ключовий показник ефективності).

SQL – Structured Query Language.

SSAS – SQL Server Analysis Services.

СД – сховище даних.

БД – база даних.

## ВСТУП

**Актуальність.** У сучасному сільському господарстві птахівництво є однією з провідних галузей, ефективність якої значною мірою залежить від підтримання оптимальних параметрів мікроклімату у пташниках. Відхилення параметрів мікроклімату від норми призводить до зниження продуктивності та збільшення економічних втрат. Підприємства галузі накопичують великі обсяги даних про показники середовища, виробничі параметри та результати діяльності. Для ефективного управління цими процесами необхідні інструменти, здатні забезпечити аналітичну обробку, оцінку та візуалізацію даних. Тому розробка системи моніторингу параметрів мікроклімату, яка використовує сучасні технології зберігання, аналізу та інтелектуальної обробки інформації, є актуальним напрямом дослідження. Така система дозволить виявляти закономірності, контролювати зміни та приймати обґрунтовані рішення для оптимізації умов утримання птиці.

**Об'єктом дослідження** є процес вирощування птиці у промисловому пташнику.

**Предметом дослідження** є інтелектуальна система моніторингу та аналізу параметрів мікроклімату у пташнику.

**Метою дослідження** є практичне застосування методів OLAP та Data Mining для аналізу параметрів мікроклімату з метою виявлення закономірностей, що дозволяють підвищити ефективність управління умовами утримання птиці.

**Зміст поставлених завдань.** Для досягнення поставленої мети необхідно виконати такі завдання:

- 1) провести системний аналіз предметної області;
- 2) побудувати модель предметної області та спроектувати архітектуру системи, включно з моделями оперативної бази даних і сховища даних, визначити джерела даних системи;

- 3) розробити багатовимірну модель і реалізувати OLAP-куб для аналітичної обробки інформації;
- 4) застосувати методи Data Mining для дослідження залежностей між параметрами мікроклімату;
- 5) сформулювати висновки щодо ефективності застосованих підходів до аналізу даних.

**Методи дослідження.** У роботі використано технології OLAP для багатовимірного аналізу даних. Для інтелектуального аналізу застосовано алгоритми Data Mining: 1-Rule, Наївного Байєса, асоціативних правил і кластеризації, які дозволяють виявити закономірності у показниках мікроклімату та рівнях смертності птиці.

**Наукова складова.** У результаті дослідження запропоновано використати підхід до аналітичного моніторингу параметрів мікроклімату у промисловому пташнику на основі інтеграції технологій OLAP і методів Data Mining. Розроблено унікальне сховище даних для ефективного зберігання та обробки інформації з багатьох господарств, що забезпечує можливість виявлення закономірностей між параметрами мікроклімату та показниками продуктивності виробництва.

#### **Апробація.**

1. Бабій Б. Ю., Голуб Б. Л. Система моніторингу параметрів мікроклімату у промисловому пташнику. Збірник наукових праць за матеріалами VII Всеукраїнської науково-практичної конференції студентів і аспірантів «ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ 2025», 24 квітня 2025 року, НУБіП України, Київ. С 58 – 59. (електронне видання)
2. Бабій Б. Ю. Впровадження сховища даних у системі моніторингу параметрів мікроклімату промислового пташника. Матеріали II Міжнародної науково-практичної конференції «Актуальні питання розвитку науки та техніки» (Боярка, 2025 р.): зб. наук. тез доп. /

Боярський ФК НУБіП України. – Боярка: ВСП БФК НУБіП України, 2025. С 137 – 139. (електронне видання)

3. XVI Міжнародна науково-практична конференція молодих вчених «Інформаційні технології: економіка, техніка, освіта» (м. Київ. 2025 р.).

**Структура роботи.** Магістерська кваліфікаційна робота складається з 73 сторінок, містить 4 розділи, висновки, список використаних джерел із 21 найменування та 7 додатків. Робота містить 34 рисунка і 1 таблицю.

- 1) Перший розділ присвячено аналізу предметної області. Розглянуто процес вирощування птиці у промисловому пташнику як об'єкт дослідження, проведено огляд існуючих систем моніторингу параметрів мікроклімату та сформульовано постанову завдання;
- 2) другий розділ містить моделювання системи. Наведено основні концепції побудови, розроблено діаграми прецедентів та класів які описують логіку функціонування системи моніторингу параметрів мікроклімату;
- 3) третій розділ присвячений розробці системи. Виконано побудову архітектури системи, описано джерела даних, спроектовано сховище даних, розроблено процес його наповнення для подальшої аналітичної обробки і розглянуто модуль аналітики даних;
- 4) четвертий розділ містить результати дослідження. Наведено використання OLAP-технології для багатовимірного аналізу даних та застосування методів Data Mining для виявлення закономірностей між параметрами мікроклімату та результатами утримання птиці.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Процес вирощування птиці у промисловому пташнику як об'єкт дослідження

Процес промислового вирощування птиці є складним багатофакторним технологічним процесом, метою якого є отримання максимальної продуктивності від поголів'я за призначенням (м'ясо або яйця) при забезпеченні належних умов утримання і дотриманні ветеринарно-санітарних норм. У сучасному птахівництві основну частку становить вирощування свійської птиці, передусім курей, які розводяться за двома основними напрямками: бройлери – курчата, інтенсивно відгодовувані на м'ясо протягом короткого циклу, та несучки – кури яєчного напрямку, яких утримують протягом тривалого періоду для отримання яєць. Окрім курей, у промислових пташниках можуть вирощуватися індички, качки, гуси та інша птиця (переважно на м'ясо), але кури залишаються домінуючим об'єктом завдяки високій ефективності їх продуктивності. В обох випадках кінцевою метою є отримання якісної продукції (м'ясо бройлерів або курячі яйця) при мінімальних затратах, що вимагає створення оптимального мікроклімату та умов годівлі й напування для забезпечення здоров'я і високої продуктивності стада.

Вирощування птиці в умовах промислового птахівництва відбувається у спеціально обладнаних приміщеннях – пташниках, у яких створюється і підтримується оптимальний мікроклімат. До основних параметрів мікроклімату відносять температуру повітря, відносну вологість, швидкість руху повітря, склад повітря (концентрації кисню та шкідливих газів, таких як вуглекислий газ  $\text{CO}_2$  і аміак  $\text{NH}_3$ ), рівень освітленості, а також запиленість середовища. Підтримання кожного з цих параметрів у допустимих межах є критично важливим для здоров'я, росту і продуктивності птиці. Наприклад,

температурний режим має бути пристосований до виду і віку птиці: добовим курчатам потрібна підвищена температура близько 32–34 °С, яку поступово знижують до ~20–22 °С у віці 5–6 тижнів. Дорослі несучки, навпаки, найкраще продуктивні за помірніших температур – в межах приблизно 20–24 °С (оптимум близько 20 °С). Відносна вологість повітря у пташнику зазвичай підтримується на рівні 60–70%, оскільки надто висока вологість може призводити до накопичення конденсату, розвитку мікроорганізмів та погіршення підстилки, а надто низька – до зайвої запиленості і пересихання слизових оболонок у птиці. Газовий склад повітря контролюється шляхом вентиляції: необхідно забезпечити достатнє надходження кисню та видалення надлишкового CO<sub>2</sub>, аміаку та інших шкідливих газів. В пташниках встановлено гранично допустимі концентрації: вміст CO<sub>2</sub> бажано утримувати менше ~0,25% (2500 ppm), а аміаку – не більше 20–25 ppm, оскільки перевищення цих норм призводить до пригнічення птахів (сонливість, втрата апетиту при CO<sub>2</sub> > 3000 ppm) та подразнення слизових оболонок і шкіри (при NH<sub>3</sub> > 25 ppm). Швидкість руху повітря (вентиляція) впливає як на тепловіддачу, так і на рівномірність розподілу температури у приміщенні. Для молодняку у перші дні життя уникають протягів – рекомендується невелика швидкість на рівні пташеняти (~0,1–0,2 м/с), оскільки навіть 0,5 м/с може викликати переохолодження курчат. Натомість для дорослої птиці в спекотний період підвищена швидкість повітря (0,5–2 м/с) на рівні птиці допомагає ефективніше охолоджувати організм і запобігати тепловому стресу [1]. Важливим фактором є також освітлення: інтенсивність та режим світла регулюються відповідно до виду птиці та фази вирощування. Наприклад, для бройлерів у перші дні підтримують освітленість ~20–40 лк цілодобово, щоб стимулювати споживання корму і води, а з другого тижня поступово вводять періоди затемнення, знижуючи освітленість до ~5–10 лк, щоб покращити конверсію корму та запобігти стресам. Таким чином, оптимальні умови утримання є сукупністю збалансованих параметрів мікроклімату: за оптимальної температури та вологості птиця почувається комфортно і має максимальний приріст чи продуктивність, тоді як відхилення (надто спекотно, холодно, сиро чи

сухо, недостатня вентиляція тощо) призводять до стресу, захворювань і зниження продуктивності.

Формування мікроклімату пташника залежить як від зовнішніх, так і внутрішніх факторів. До зовнішніх належать кліматичні умови і сезонність: температура зовнішнього повітря, вологість та погодні коливання впливають на тепловтрати будівлі, режим вентиляції і роботу систем опалення чи охолодження. Наприклад, взимку низька температура довкілля вимагає інтенсивного обігріву пташника, але водночас знижує відносну вологість всередині приміщення; влітку висока зовнішня температура потребує активнішої вентиляції та охолодження для запобігання перегріву птиці. Внутрішні фактори включають щільність посадки птиці (кількість голів на одиницю площі), стан та вологість підстилки, тепло, що виділяється самою птицею, роботу систем вентиляції та мікрокліматичного обладнання. Зростання щільності посадки або біомаси птиці збільшує тепловиділення і вологовиділення в приміщенні, прискорює накопичення вуглекислого газу і аміаку, тому при великому поголів'ї потрібна більш ефективна вентиляція. Волога підстилка є джерелом випаровування і аміаку внаслідок розкладання посліду, що може погіршувати якість повітря. Несправність або неправильно налаштована робота систем вентиляції, опалення чи зволоження призводить до дисбалансу параметрів – наприклад, недостатня вентиляція викликає накопичення  $\text{CO}_2$  та  $\text{NH}_3$ , а надмірна – протяги і переохолодження. Будь-яка зміна вказаних факторів впливає на тепловий баланс приміщення та загальний стан мікроклімату [2]. Таким чином, підтримання стабільного мікроклімату є результатом постійного управління, що враховує як погодні умови, так і динаміку росту та поведінки самої птиці.

Для підтримання оптимальних параметрів середовища у сучасних птахівничих комплексах широко застосовуються автоматизовані системи контролю мікроклімату. До складу такої системи входять численні датчики: температурні датчики (контролюють температуру повітря на рівні птиці), гігрометри (вимірюють відносну вологість), датчики концентрації  $\text{CO}_2$  та  $\text{NH}_3$  (для контролю газового складу повітря), а також сенсори освітленості та, за

потреби, датчики запиленості. Інформація з датчиків в режимі реального часу надходить до центрального контролера (комп'ютеризованої системи керування кліматом). Контролер порівнює отримані значення з заданими оптимальними параметрами і за відхилення автоматично керує виконавчими пристроями: вентиляторами і клапанами вентиляційної системи, обігрівачами, тепловими пушками або системами підігріву підлоги, зволожувачами або, навпаки, вентиляторами охолодження та системами випарного охолодження, а також системами освітлення. Таким чином реалізується принцип автоматичного регулювання мікроклімату: при підвищенні температури понад норму – посилюється вентиляція або вмикається охолодження, при зниженні – запускається опалення; при зростанні вологості – підсилюється вентиляція, при надто сухому повітрі – може вмикатись зволоження тощо. Всі зібрані датчиками дані можуть накопичуватися в базі даних комп'ютерної системи моніторингу, де вони зберігаються для подальшої обробки та аналізу. Наявність історичних даних про параметри середовища, технологічні операції та стан поголів'я дозволяє відстежувати тенденції, виявляти проблемні періоди та ухвалювати обґрунтовані управлінські рішення для покращення умов утримання. Крім того, такі системи забезпечують можливість віддаленого контролю оператором, сигналізацію про аварійні ситуації (наприклад, різке відхилення температури або відмову обладнання) та інтеграцію з іншими підсистемами (годівлі, напування, відеоспостереження тощо) для комплексного керування пташником.

Одним із головних показників ефективності технології утримання є продуктивність птиці, тобто досягнуті зоотехнічні та економічні результати. Для бройлерів ключовими параметрами продуктивності є середньодобовий приріст живої маси та кінцева жива маса, конверсія корму (співвідношення спожитого корму до приросту маси), рівень збереженості поголів'я (відсоток випадків падежу), а також якість м'яса (категорія тушок, вихід їстівних частин). Для несучок основними показниками продуктивності є інтенсивність яйцекладки (кількість яєць на курку за певний період), якість яєць (маса, міцність шкаралупи, категорія товарності), тривалість продуктивного періоду та також показники

збереженості стада. Важливо, що на всі ці результати істотно впливають умови утримання, особливо параметри мікроклімату. Встановлення закономірностей між змінними мікроклімату (такими як температура, вологість, концентрація газів) і показниками продуктивності дає змогу оцінити, наскільки ефективними є поточні умови утримання, та виявити резерви для вдосконалення технологічного процесу. Наприклад, аналіз даних може показати, що при підвищенні температури понад оптимум у певний період зростає падіж або погіршується конверсія корму, або що коливання вологості позначаються на рівні захворюваності птиці. Виявлені залежності дозволяють коригувати режими вентиляції, опалення чи охолодження, змінювати щільність посадки чи інші умови з метою оптимізації продуктивності та добробуту птиці [3].

Отже, процес вирощування птиці у промисловому пташнику виступає складним багатопараметричним об'єктом дослідження, в якому велика кількість факторів перебувають у тісному взаємозв'язку. Аналітичне вивчення цих залежностей на основі сучасних технологій обробки даних дає можливість розробити ефективну систему моніторингу мікроклімату, здатну забезпечити стабільність виробництва, зниження витрат і підвищення якості продукції.

## **1.2 Огляд існуючих рішень**

На сьогоднішній день на ринку існує низка промислових систем, призначених для моніторингу та автоматизованого керування мікрокліматом у пташниках. Такі системи зазвичай поєднують у собі сенсорні пристрої, контролери, засоби збору даних та програмне забезпечення для візуалізації і керування процесами.

Для порівняння було розглянуто три найпоширеніші системи — SKOV BlueControl, Big Dutchman ViperTouch та Fansom Lumina, які застосовуються на великих птахівничих фермах і мають різний рівень інтеграції апаратного та аналітичного забезпечення.

SKOV BlueControl — це лінійка інтелектуальних контролерів датської компанії SKOV, призначених для автоматизованого моніторингу та керування параметрами мікроклімату у пташниках для бройлерів, несучок та племінної птиці. Система побудована за модульним принципом, що дозволяє адаптувати її конфігурацію до потреб конкретного виробництва — від невеликих пташників до великих промислових комплексів.

BlueControl поєднує спеціалізоване апаратне забезпечення та програмні модулі для повного контролю умов середовища та виробничих процесів. Керування здійснюється через великий сенсорний екран із графічними інтерфейсними елементами, що забезпечує високу зручність та інтуїтивність для оператора. Меню мають ієрархічну структуру з можливістю налаштування прав доступу для різних рівнів персоналу, що підвищує безпеку і контрольованість процесів. Система підтримує понад 40 мов інтерфейсу, що робить її універсальною для міжнародного використання.



Рис. 1 SKOV BlueControl

Основні можливості SKOV BlueControl включають:

- регулювання параметрів мікроклімату — температура, вологість, швидкість руху повітря, вентиляція, а також режими опалення та охолодження;
- керування виробничими процесами: подачею води та корму, освітленням, системами зволоження або осушення повітря;
- модульність конфігурації — можливість встановлення лише тих функціональних блоків, які потрібні конкретному господарству;
- зручний інтерфейс із сенсорним екраном та графічними іконками, що забезпечує інтуїтивність керування;
- гнучка система рівнів доступу — можливість налаштування меню та параметрів під різні ролі користувачів;
- підтримка понад 40 мов, що спрощує використання для операторів із різних країн.

Система дозволяє відстежувати параметри роботи обладнання та мікроклімату в реальному часі, швидко змінювати режими утримання птиці відповідно до віку та технологічних вимог, а також адаптувати сценарії вентиляції залежно від поточних умов [4].

Недоліком SKOV BlueControl є те, що, попри ефективність керування процесами, система орієнтована переважно на оперативний контроль і не передбачає широких можливостей для аналітичного або прогнозного оброблення даних. Історія параметрів зберігається головним чином у внутрішній пам'яті контролера, а інтеграція з зовнішніми інструментами аналітики (OLAP, Data Mining) обмежена. Це ускладнює виконання глибинного аналізу, порівняння показників між партіями вирощування або виявлення довгострокових закономірностей.

Таким чином, SKOV BlueControl ефективно підтримує стабільність мікроклімату і зручне керування виробничими процесами, проте не забезпечує розширеної аналітики, необхідної для оптимізації технологічних рішень на основі історичних даних.

ViperTouch — це комплексна система керування мікрокліматом і виробничими процесами у пташниках, розроблена німецькою компанією Big Dutchman. Вона призначена для бройлерів, індиків, качок та племінного поголів'я. Система поєднує контроль середовища, керування годівлею і водопостачанням, моніторинг продуктивності та відображення ключових параметрів у реальному часі.

Основою системи є центральний контролер із сенсорним екраном, який забезпечує інтуїтивне керування через графічне меню. Система підтримує декілька режимів керування мікрокліматом — від базового до розширеного, що дозволяє адаптувати її до різних умов господарства.



Рис. 2 ViperTouch

Основні можливості ViperTouch охоплюють:

- контроль вентиляційних стратегій (природна, механічна, тунельна, комбінована вентиляція);
- регулювання подачі свіжого та відпрацьованого повітря, опалення, охолодження та зволоження;
- підтримка підключення до 8 температурних датчиків, а також датчиків CO<sub>2</sub>, NH<sub>3</sub>, вологості та атмосферного тиску;
- можливість окремого керування зонами всередині пташника;

- аварійне відкриття вентиляційних каналів у разі критичних відхилень;
- відображення історичних трендів параметрів;
- моніторинг споживання корму та води;
- керування програмами годівлі та освітлення;
- контроль ваги птиці та облік показників смертності;
- підтримка обладнання для підрахунку яєць (для племінних господарств);
- простий у використанні інтерфейс із можливістю налаштування головного екрана;
- підтримка 32 мов;
- система багаторівневого доступу через паролі;
- можливість збереження налаштувань і історії параметрів.

Хоча ViperTouch забезпечує широкі можливості керування та моніторингу, його функції аналізу даних обмежуються базовою звітністю та переглядом трендів. Система не має вбудованих засобів глибинної аналітики чи прогнозування та не підтримує OLAP або Data Mining, що знижує її ефективність для стратегічного аналізу або виявлення довгострокових закономірностей у даних [5].

Таким чином, ViperTouch є надійним і гнучким рішенням для оперативного керування мікрокліматом і виробничими процесами, проте не забезпечує інтелектуального аналізу великих обсягів даних, необхідного для оптимізації технологічних рішень на основі історичних показників.

Fansom Lumina — це лінійка контролерів клімату та виробничих процесів для птахівничих господарств, розроблена нідерландською компанією Fansom. Система орієнтована на забезпечення оптимальних умов вирощування птиці завдяки точному керуванню вентиляцією, температурою, вологістю та освітленням. Модульна архітектура дозволяє адаптувати систему до різних типів пташників — від невеликих ферм до великих промислових комплексів.

Ключовою особливістю Lumina є зручність роботи: інтерфейс контролера побудований на основі графічних іконок та кольорових індикаторів. Зелений колір означає, що всі параметри в нормі, тоді як оранжевий вказує на фактор, який потребує уваги оператора. Це значно спрощує використання системи та знижує ризик помилок при зміні налаштувань.



Рис. 3 Fancom Lumina

Система дозволяє:

- вимірювати та автоматично регулювати температуру, вологість, рівень CO<sub>2</sub> і вентиляцію;
- керувати вентиляційними системами різної конфігурації;
- контролювати інтенсивність освітлення та його програми;
- підтримувати оптимальні умови з урахуванням віку птиці та виробничої технології.

Контролери Lumina забезпечують більш глибоке розуміння мікроклімату завдяки можливості збільшення масштабу показників та відображення детальних даних по кожному параметру в реальному часі.

Усі пристрої Lumina підтримують мобільний додаток Infinia Remote App, що забезпечує:

- цілодобовий доступ до стану пташника з будь-якої точки світу;

- перегляд ключових параметрів у спрощеному форматі;
- можливість миттєво змінювати налаштування клімату;
- автоматичні сповіщення про відхилення параметрів від норми.

Таким чином, оператор може реагувати на проблеми навіть без присутності на фермі, що підвищує стабільність мікроклімату та знижує ризики втрат.

Історія параметрів може зберігатися в Lumina Cloud, що дозволяє:

- переглядати динаміку змін умов утримання птиці;
- формувати графіки та базові звіти;
- оцінювати ефективність роботи системи за період.

Однак, аналітичні можливості системи обмежені:

- відсутні інструменти прогнозування або виявлення прихованих закономірностей;
- неможливо застосовувати методи OLAP чи Data Mining до накопичених даних;
- система орієнтована переважно на моніторинг і оперативне керування, а не на глибинний аналіз.

Таким чином, Fansom Lumina є ефективним рішенням для щоденного контролю мікроклімату та оперативної стабілізації умов вирощування птиці. Завдяки простому інтерфейсу та можливості дистанційного керування система зручна у використанні [6]. Проте обмежені можливості інтелектуального аналізу даних не дозволяють застосовувати її для глибокого дослідження взаємозв'язків між параметрами середовища та продуктивністю, що створює потребу у додаткових аналітичних засобах.

Розглянуті системи SKOV BlueControl, Big Dutchman ViperTouch та Fansom Lumina демонструють високий рівень автоматизації процесів утримання птиці та забезпечують стабільний контроль мікроклімату в пташниках. Вони дозволяють регулювати параметри середовища, керувати вентиляцією, годуванням, освітленням та іншими технологічними процесами. Однак спільною

характеристикою всіх описаних рішень є їх орієнтація переважно на оперативне керування, а не на аналітичне опрацювання накопичених даних.

Дані, що збираються цими системами, зберігаються у внутрішніх середовищах або хмарних сервісах, але доступ користувача до глибокого аналізу, побудови багатовимірних моделей, виявлення закономірностей або прогнозування поведінки параметрів фактично обмежений. Вони не надають інструментів для застосування методів OLAP чи Data Mining, що унеможливує комплексну оцінку впливу параметрів мікроклімату на результати виробництва.

### **1.3 Огляд наукових праць**

Також був проведений пошук наукових праць за темою магістерської роботи. Нижче наведено короткий аналіз обраних робіт.

У статті “Multivariate analysis for data mining to characterize poultry house environment in winter” було зібрано великий масив даних про параметри мікроклімату з 60 різних точок у промисловому пташнику протягом кількох фаз вирощування птиці. Дослідження включало вимірювання температури, вологості, концентрації CO<sub>2</sub> та NH<sub>3</sub>, пилових частинок різної фракції, а також швидкості повітря в клітках і проходах. Було застосовано методи Spearman-кореляції для визначення взаємозв'язків між показниками та PCA (метод головних компонент) для виявлення груп параметрів, що мають найбільший вплив на якість мікроклімату. Далі автори використали три алгоритми кластеризації (k-means, k-medoids і FCM) для розподілу внутрішнього простору пташника на зони з різними умовами повітря. У результаті було встановлено, що центральна зона пташника має гірші показники якості повітря, а ключову роль відіграють відносна вологість та швидкість руху повітря. Дослідження показало практичну можливість просторового зонування пташника з використанням Data Mining, що може стати основою для адаптивного керування вентиляцією [7].

У роботі “A Heuristic and Data Mining Model for Predicting Broiler House Environment Suitability” виконано побудову системи прогнозування умов утримання птиці на основі дерева рішень. Для цього були зібрані дані з чотирьох комерційних пташників, включаючи показники вологості, температури, концентрації аміаку, швидкості вентиляції та вік птиці. Спочатку проведено описовий аналіз, за допомогою якого дослідники визначили вплив аміаку як критичного фактора, що найбільш суттєво впливає на продуктивність. Далі було сформовано навчальну вибірку з розподілом станів умов вирощування на чотири класи: «Відмінні», «Добрі», «Помірні» та «Неприйнятні». На основі цієї вибірки побудовано дві моделі — на основі випадкового дерева та класичного дерева рішень. Точність моделі рішення виявилась суттєво вищою: вона коректно ідентифікувала критичні режими та дозволила сформулювати чіткі правила виду «якщо–то», які можуть застосовуватись безпосередньо в системах підтримки прийняття рішень, і показала, що вік бройлерів, відносна вологість та концентрація аміаку відіграють вирішальну роль у забезпеченні належних умов вирощування. Таким чином, у роботі досягнуто побудови практичної моделі для автоматизованого визначення оптимальних параметрів мікроклімату з урахуванням вікових фаз вирощування птиці [8].

Автори статі “Algorithm for Autonomous Management of a Poultry Farm by a Cyber-Physical System” розробили та реалізували кіберфізичну систему автономного контролю мікроклімату пташника. Система побудована на основі контрольного модуля Raspberry Pi та мережі сенсорів, підключених до OpenHAB 3.0 як системи автоматизації. У межах системи реалізовано алгоритми, що автоматично регулюють температуру, вологість та швидкість вентиляції залежно від встановлених параметрів. Було досягнуто повної автономності підтримання мікроклімату — система працює без участі працівника і лише надсилає сповіщення у випадках критичних відхилень. Основним досягненням є створення доступної та економічної системи автоматичного контролю, яка мінімізує вплив людського фактора [9].

Таким чином, існує потреба у розробці інтелектуальної системи моніторингу та аналізу даних, яка не лише здійснюватиме контроль параметрів середовища, але й забезпечуватиме накопичення інформації, її багатовимірний аналіз, виявлення прихованих закономірностей та формування обґрунтованих управлінських рішень для підвищення ефективності вирощування птиці. Жодне з проаналізованих рішень не реалізує повний замкнений цикл від збору та зберігання даних до аналітичної обробки і підтримки прийняття рішень, що підкреслює актуальність розробки даної системи.

#### **1.4 Постановка завдання**

Необхідно розробити систему моніторингу параметрів мікроклімату у промисловому пташнику, що забезпечує збір, зберігання та аналітичну обробку даних для підвищення ефективності процесу вирощування птиці.

Система повинна надавати можливість отримувати відповіді на певні запитання, необхідні для контролю й оцінки стану середовища, зокрема:

- які екстремуми параметрів мікроклімату спостерігалися у пташнику за певний період часу;
- кількість отриманої продукції за вказаний період часу;
- які кореляційні зв'язки існують між показниками мікроклімату та рівнем смертності або продуктивності;
- які параметри найбільше впливають на якість утримання та потребують коригування.

Для реалізації зазначених можливостей необхідно:

- визначити перелік параметрів мікроклімату, що підлягають моніторингу, та сформулювати вимоги до системи збору даних;
- встановити структуру зберігання інформації та побудувати сховище даних, здатне накопичувати показники за тривалий період;

- забезпечити механізми візуалізації та аналізу показників у вигляді таблиць, графіків та звітів;
- реалізувати обробку даних для виявлення залежностей між параметрами мікроклімату та кінцевими результатами вирощування птиці;
- надати можливість порівняння даних між різними періодами, партіями вирощування або пташниками;
- сформувати функціональні засоби для своєчасного виявлення критичних відхилень та факторів ризику.

Отримані результати мають забезпечити підвищення контролю за умовами утримання птиці та слугувати основою для прийняття керуючих рішень щодо стабілізації та оптимізації мікроклімату у виробничому середовищі.

## 2 МОДЕЛЮВАННЯ ПРОЦЕСІВ ВИРОЩУВАННЯ ПТИЦІ У ПТАШНИКУ

### 2.1 Основні концепції

Моделювання системи є одним із ключових етапів розроблення програмного забезпечення, оскільки воно дозволяє формалізувати структуру, логіку та поведінку системи до початку її реалізації.

Основною метою моделювання є створення узагальненої моделі, що описує функціонування системи моніторингу параметрів мікроклімату у промисловому пташнику, її основні елементи, взаємозв'язки та сценарії взаємодії користувачів із системою.

Для опису системи застосовуються два підходи — функціональний і об'єктно-орієнтований.

Функціональне моделювання використовується для опису логіки процесів системи, послідовності дій та потоків даних між її складовими. У межах цього підходу визначаються основні функції системи, вхідні та вихідні дані, а також взаємозв'язки між підсистемами.

Для відображення функціональних аспектів системи застосовуються такі типи діаграм:

1. Діаграма прецедентів (Use Case Diagram) — відображає взаємодію користувачів (акторів) із системою та визначає основні сценарії її використання.
2. Діаграма активності (Activity Diagram) — описує послідовність дій при виконанні процесів моніторингу та аналізу даних, від моменту надходження показників до формування результатів аналізу.

Завдяки функціональному підходу визначаються межі системи, її зовнішні взаємодії та основні бізнес-процеси.

Об'єктно-орієнтований підхід орієнтований на опис структури системи, взаємозв'язків між об'єктами та способів їхньої взаємодії. У цьому підході система розглядається як сукупність об'єктів, що мають власні атрибути та методи.

Для представлення об'єктної моделі використовуються такі діаграми:

1. Діаграма класів (Class Diagram) — показує основні сутності системи та зв'язки між ними (асоціації, композиції, наслідування).
2. Діаграма послідовності (Sequence Diagram) — демонструє взаємодію між об'єктами під час виконання окремих сценаріїв.
3. Діаграма розгортання (Deployment Diagram) — відображає фізичну архітектуру системи: сервер бази даних, клієнтські додатки, аналітичні сервіси та сенсорні пристрої.

Об'єктно-орієнтоване моделювання дозволяє формалізувати архітектуру системи та підготувати основу для її програмної реалізації.

Таким чином, функціональний підхід визначає що саме робить система, а об'єктно-орієнтований — як саме вона це робить. Їх поєднання забезпечує комплексне бачення майбутньої системи: від функціональної поведінки користувачів до структурної побудови компонентів і механізмів їхньої взаємодії [10].

## 2.2 Діаграма прецедентів

Діаграма прецедентів — це один із видів діаграм UML, який використовується для моделювання функціональних вимог системи. Вона демонструє взаємодію зовнішніх користувачів (акторів) із системою через набір прецедентів. Вона складається з акторів та прецедентів.

Актор — це зовнішня сутність, яка взаємодіє з системою для досягнення певної мети.

Прецедент — це опис дії або функції, яку виконує система у взаємодії з актором.

На діаграмі можуть існувати такі зв'язки:

1. Асоціація - вказує, що актор взаємодіє з певним прецедентом.
2. Зв'язок «include» - один прецедент включає в себе інший, тобто викликає його під час виконання.
3. Зв'язок «extend» - додаткова поведінка, яка виконується лише за певних умов.
4. Генералізація - один актор або прецедент є узагальненням для інших.

Для побудови діаграми прецедентів розроблюваної системи перш за все необхідно визначити акторів, які взаємодіють з нею [11]. На таблиці 1 показано перелік акторів, які взаємодіють з системою.

Таблиця 1

Актори діаграми прецедентів

<b>Актор</b>	<b>Опис</b>
Оператор	Спостерігає за значеннями параметрів мікроклімату у пташнику, за необхідності регулює параметри вручну, опрацьовує виміряні значення, вносить дані про продуктивність птиці
Аналітик	Проводить аналіз роботи пташника та формує аналітичну звітність
Керівник	Приймає рішення щодо управління пташником
Сенсор	Вимірює значення параметрів мікроклімату

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. 4.

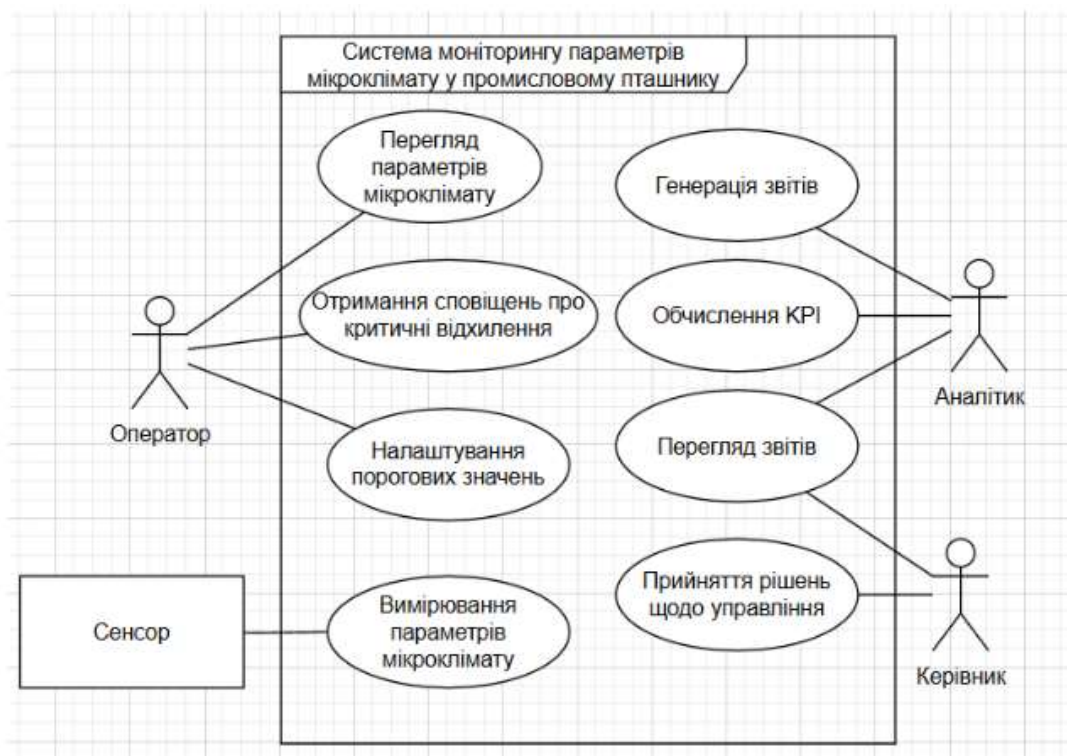


Рис. 4 Діаграма прецедентів

Діаграма зображує взаємодію трьох основних користувачів системи: оператора, аналітика та керівника, а також показує роль сенсора, який забезпечує збір даних.

Оператор є ключовою ланкою в роботі системи. Він переглядає параметри мікроклімату, отримує сповіщення про критичні відхилення у разі виявлення проблем і має можливість налаштовувати порогові значення для відстеження параметрів. Сенсор, як автономний елемент, здійснює вимірювання параметрів і передає їх до системи, де вони стають доступними для перегляду.

Аналітик використовує дані системи для створення звітів, обчислення KPI та аналізу отриманих результатів. Його робота допомагає керівнику приймати обґрунтовані управлінські рішення. Керівник, у свою чергу, отримує доступ до звітів, що надають йому інформацію для оцінки стану системи та прийняття рішень щодо управління процесами у пташнику.

## 2.3 Діаграма класів

Діаграма класів відображає структуру системи моніторингу параметрів мікроклімату у промисловому пташнику, показуючи основні класи, їх атрибути, методи та зв'язки між ними. Вона ілюструє логічну модель системи, яка поєднує елементи збору даних, контролю, обробки та взаємодії з користувачами. Діаграму зображено на рис. 5.

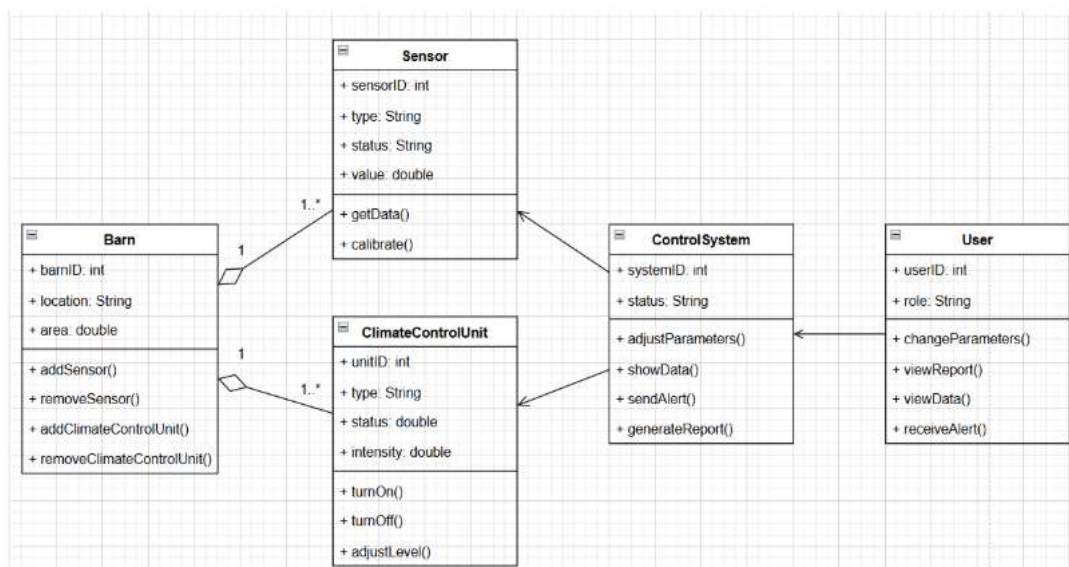


Рис. 5 Діаграма класів

На даній діаграмі представлено 5 класів, розглянемо їх детальніше.

**Barn** (Пташник) — клас, що описує фізичне приміщення пташника, у якому відбувається контроль мікроклімату.

- атрибути: `barnID`, `location`, `area`;
- методи: `addSensor()`, `removeSensor()`, `addClimateControlUnit()`, `removeClimateControlUnit()`.

**Barn** має агрегаційний зв'язок із класами **Sensor** і **ClimateControlUnit**, оскільки сенсори та блоки клімат-контролю є частиною пташника, але можуть функціонувати незалежно.

**Sensor** (Сенсор) — клас, що відповідає за вимірювання параметрів мікроклімату (температура, вологість, концентрація  $\text{CO}_2$ ,  $\text{NH}_3$  тощо).

- атрибути: sensorID, type, status, value;
- методи: getData(), calibrate().

Клас Sensor агрегується у Barn, що означає його логічну приналежність до пташника.

ClimateControlUnit (Блок клімат-контролю) — клас, який керує системами підтримання параметрів мікроклімату.

- атрибути: unitID, type, status, intensity;
- методи: turnOn(), turnOff(), adjustLevel().

Цей клас має агрегаційний зв'язок із Barn, оскільки входить до його складу, але може працювати автономно.

ControlSystem (Система управління) — центральний компонент, що координує роботу сенсорів і кліматичних блоків, виконує аналітичні обчислення й формує звіти.

- атрибути: systemID, status;
- методи: adjustParameters(), showData(), sendAlert(), generateReport().

Клас ControlSystem асоційований із ClimateControlUnit для керування кліматичними блоками, а також із Sensor для збору даних.

User (Користувач) — клас, що моделює користувачів системи (технологів, операторів, ветеринарів).

- атрибути: userID, name, role;
- методи: changeParameters(), viewReport(), viewData(), receiveAlert().

User асоційований із ControlSystem, оскільки користувач взаємодіє із системою, отримує інформацію та надсилає команди.

Типи зв'язків, використаних на діаграмі [12]:

1. Асоціація — використовується між User і ControlSystem, ControlSystem і ClimateControlUnit, ControlSystem і Sensor. Цей тип зв'язку відображає взаємодію між об'єктами без залежності життєвого циклу.

2. Агрегація — використовується між `Varn` і класами `Sensor` та `ClimateControlUnit`. Вона показує відношення «ціле–частина», де частини можуть існувати самостійно.

Таким чином, діаграма класів відображає логічну структуру системи моніторингу параметрів мікроклімату. Вона демонструє, як основні сутності системи взаємодіють між собою, забезпечуючи збір, аналіз і управління даними у пташнику.

## 3 РОЗРОБКА СИСТЕМИ

### 3.1 Архітектура системи

Діаграма розгортання (рис. 6) відображає фізичні компоненти системи моніторингу параметрів мікроклімату та канали взаємодії між ними. У системі використовується локальна інфраструктура пташника для збору та відображення параметрів середовища, а також окремий серверний сегмент, призначений для зберігання та аналітичної обробки накопичених даних.

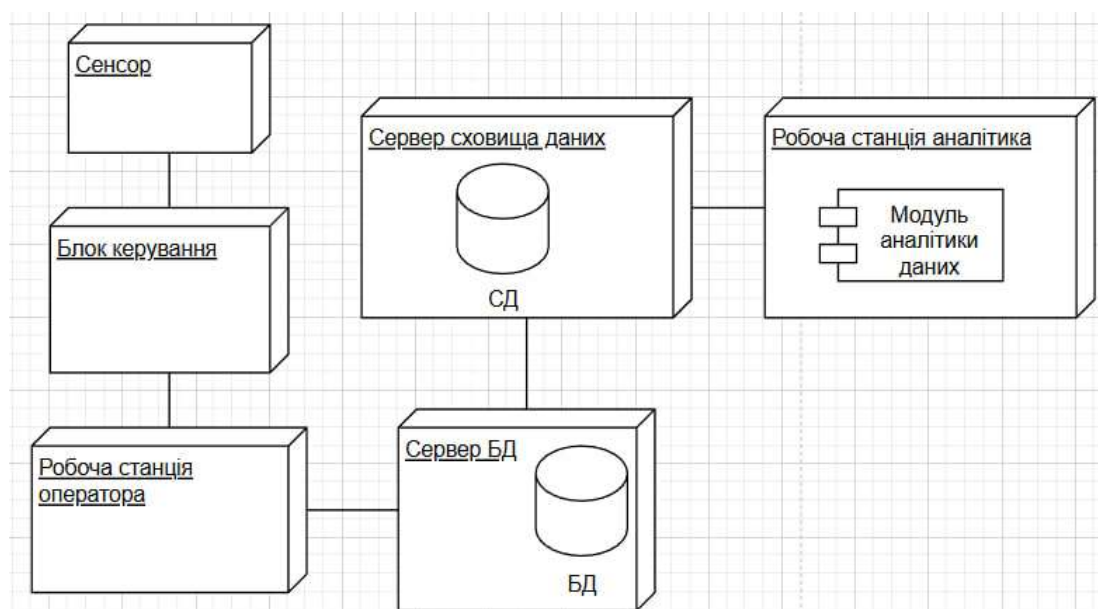


Рис. 6 Діаграма розгортання

На схемі зображено робочу станцію оператора, яка виконує функції локального інтерфейсу управління. Це персональний комп'ютер або панель із сенсорним екраном, встановлена безпосередньо в пташнику. Через неї відображаються поточні значення температури, вологості, концентрації газів, статистика змін у часі, журнали подій та попереджень. Оператор може вручну коригувати режими вентиляції, опалення або охолодження, а також налаштовувати порогові значення для автоматичних alarm-сповіщень. Робоча

станція підключена до локального сервера бази даних через мережу Ethernet або Wi-Fi.

Сервер бази даних використовується для зберігання інформації, що надходить у режимі реального часу. Він фіксує фактичні показники сенсорів, стан обладнання та історію управлінських дій оператора. Така локальна фіксація дозволяє швидко реагувати на зміни мікроклімату та забезпечує автономність у випадку тимчасової втрати зовнішнього зв'язку.

Дані до сервера надходять від сенсорів, розташованих у пташнику. Сенсори вимірюють температуру, відносну вологість, концентрацію CO<sub>2</sub> і NH<sub>3</sub>, а також тиск і швидкість руху повітря (у разі наявності відповідних датчиків). Отримані значення передаються в цифровій формі на блок керування, який аналізує показники та формує сигнали для керування обладнанням. Блок керування автоматично регулює роботу вентиляторів, клапанів, нагрівачів, системи зволоження чи охолодження, підтримуючи встановлені параметри середовища.

Для довгострокового зберігання історичних даних та виконання аналітичних запитів використовується сервер сховища даних. Він отримує дані з локального сервера бази даних, структурує їх та забезпечує можливість накопичення значних часових вибірок. Сховище виступає джерелом для виконання багатовимірного аналізу, формування звітів та визначення закономірностей у зміні параметрів мікроклімату.

Доступ до сховища даних здійснюється через робочу станцію аналітика. На ній розташоване програмне забезпечення для аналітичної обробки: OLAP-модуль для побудови багатовимірних зрізів, засоби візуалізації та статистичного аналізу, а також інструменти Data Mining для пошуку кореляцій і трендів. Аналітик може виконувати оцінку стабільності умов утримання, виявляти вплив окремих параметрів на продуктивність птиці та формувати рекомендації щодо оптимізації мікроклімату.

### 3.2 Опис джерела даних

Схема оперативної бази даних, яка знаходиться у вузлі «Сервер БД» і з якої постачаються дані до сховища, зображена на рис. 7.

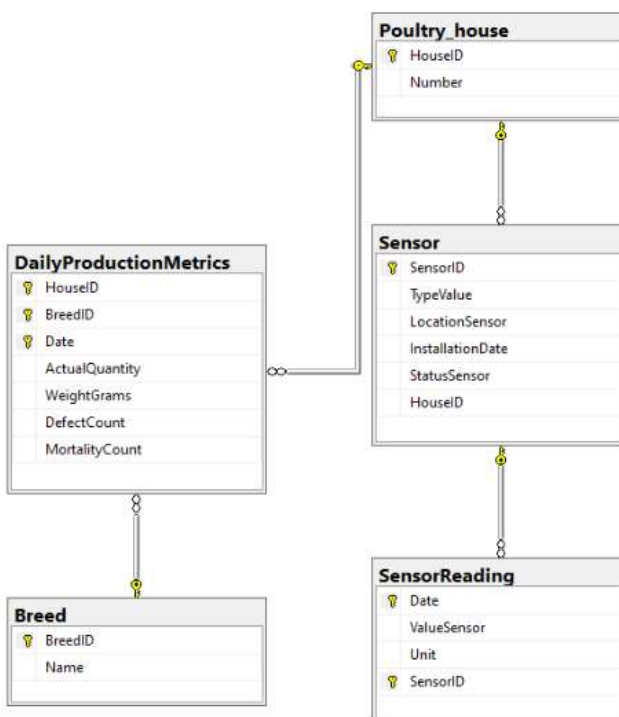


Рис. 7 Схема оперативної бази даних

Вона складається з таких таблиць:

1. **Poultry\_house** — ця таблиця містить інформацію про пташники. Вона пов'язана з іншими таблицями, такими як **Sensor** та **DailyProductionMetrics**, що дає можливість отримувати дані про сенсори, що знаходяться у пташниках, а також про щоденні виробничі метрики, пов'язані з ними.
2. **Sensor** — таблиця, що містить дані про сенсори, встановлені в пташниках для вимірювання різних параметрів мікроклімату. Її поля включають *SensorID*, *TypeValue* (тип показника), *LocationSensor* (місце розташування), *InstallationDate* (дата встановлення) та *StatusSensor* (статус датчика). Вона пов'язана з таблицею **SensorReading** через *SensorID*, де зберігаються фактичні показники

- цих сенсорів. Кожен датчик належить конкретному пташнику, що відображається через зв'язок з Poultry\_house за допомогою HouseID.
3. Breed — ця таблиця зберігає дані про породи птахів. Основним полем є BreedID, а також Name породи. Вона пов'язана з таблицею DailyProductionMetrics через BreedID, що дає змогу визначити, до якої породи належать щоденні показники виробництва.
  4. DailyProductionMetrics — ця таблиця містить щоденні показники виробництва. Її первинний ключ є композитним і складається з HouseID, BreedID та Date. Інші поля включають ActualQuantity (фактична кількість), WeightGrams (вага в грамах), DefectCount (кількість дефектів) та MortalityCount (кількість смертей). Вона пов'язана з таблицею Poultry\_house через HouseID та з таблицею Breed через BreedID, що дозволяє аналізувати продуктивність пташників залежно від породи та умов утримання за певний день.
  5. SensorReading — таблиця для зберігання показників сенсорів. Вона зберігає дані про кожне вимірювання, включаючи Date (дату/час вимірювання), ValueSensor (виміряне значення) та Unit (одиночку вимірювання). Ця таблиця має композитний первинний ключ, що складається з Date та SensorID, і зв'язана з таблицею Sensor через SensorID, що дозволяє ідентифікувати, з якого сенсора були отримані ці дані.

### 3.3 Проєктування сховища даних

Сховище даних — це спеціалізована система для зберігання та організації великих обсягів даних, призначена для ефективного аналізу та обробки інформації. Воно забезпечує централізоване зберігання даних з різних джерел і підтримує доступ до них для прийняття рішень та виконання аналітичних запитів.

Структура сховища даних визначає, як саме дані зберігаються та організовуються в системі для ефективного їх оброблення та аналізу. Вона охоплює різноманітні компоненти, такі як бази даних, таблиці, індекси, а також засоби для зберігання історичних даних. Основним завданням є забезпечення зручного доступу до даних для користувачів та аналітичних систем, а також підтримка високої швидкості обробки великих обсягів інформації [13].

Структура сховища даних, яке знаходиться у вузлі «Сервер сховища даних», для розроблюваної системи зображена на рис. 8.

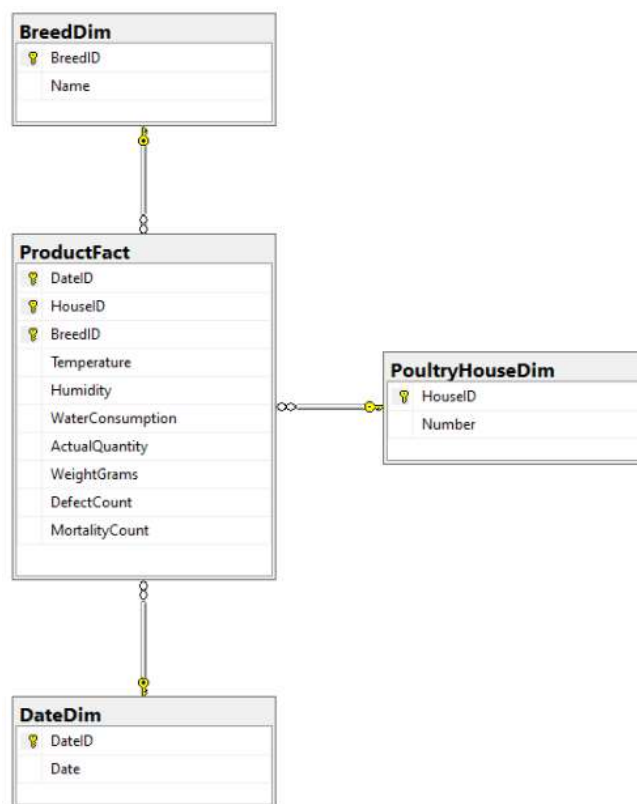


Рис. 8 Структура сховища даних

Сховище даних побудоване на основі зіркової схеми (star schema), що є оптимальним підходом для аналітичних цілей, багатовимірного аналізу та ефективної роботи з OLAP-кубами. У цій архітектурі наявна центральна таблиця фактів та декілька таблиць вимірів, які забезпечують контекст для даних.

Центральною таблицею є ProductFact. Вона містить кількісні дані про різні параметри середовища у пташниках, а також про виробництво продукції. Ця таблиця пов'язує всі виміри через свої зовнішні ключі (DateID, HouseID,

BreedID), що дозволяє проводити багатовимірний аналіз продуктивності та екологічних умов у птахівництві.

Таблиця ProductFact зв'язана з трьома вимірними таблицями, які надають деталізовану інформацію для аналізу:

1. BreedDim – ця вимірна таблиця містить довідкову інформацію про породи птиці. Вона включає їх унікальний ідентифікатор (BreedID) та Name (назву породи). Це дозволяє аналізувати показники фактів у розрізі різних порід.
2. PoultryHouseDim – цей вимір описує пташники. Він включає HouseID (унікальний ідентифікатор пташника) та Number (номер пташника). Це дозволяє відстежувати дані щодо продуктивності та умов середовища для кожного окремого пташника.
3. DateDim – цей вимір забезпечує часовий контекст для аналізу даних. Він включає DateID (унікальний ідентифікатор дати) та Date (повну дату). Це дозволяє аналізувати дані за днями, місяцями та роками, що є особливо важливим для відстеження змін у виробництві та параметрах середовища з часом.

### **3.4 Механізм вилучення, обробки і передачі даних**

Перед тим як дані потрапляють до сховища даних, вони певним чином обробляються. Цей процес має назву ETL (extract, transform, load), що означає вилучення, перетворення та завантаження. Нижче наведено процес створення багатовимірного кубу даних та виконання ETL.

SSAS є складовою частиною SQL Server і дозволяє створювати багатовимірні куби даних, які використовуються для зберігання та обробки великих обсягів даних. SSAS підтримує дві основні моделі роботи: OLAP та Data Mining. Далі буде розглянуто побудову OLAP-кубу у середовищі Microsoft Visual Studio.

Для створення нового проекту SSAS в Visual Studio слід обрати шаблон Analysis Services Project. Після створення проекту з'являється рішення, яке містить компоненти для налаштування джерел даних, вимірів та кубів.

Наступним кроком є додавання джерела даних (Data Source), що визначає базу даних, з якої буде отримуватися інформація для побудови куба. Для цього необхідно підключитися до реляційної бази даних, у моєму випадку це SQL Server, та вибрати таблиці або подання, які містять необхідні дані. На рис. 9 зображено додавання джерела даних.

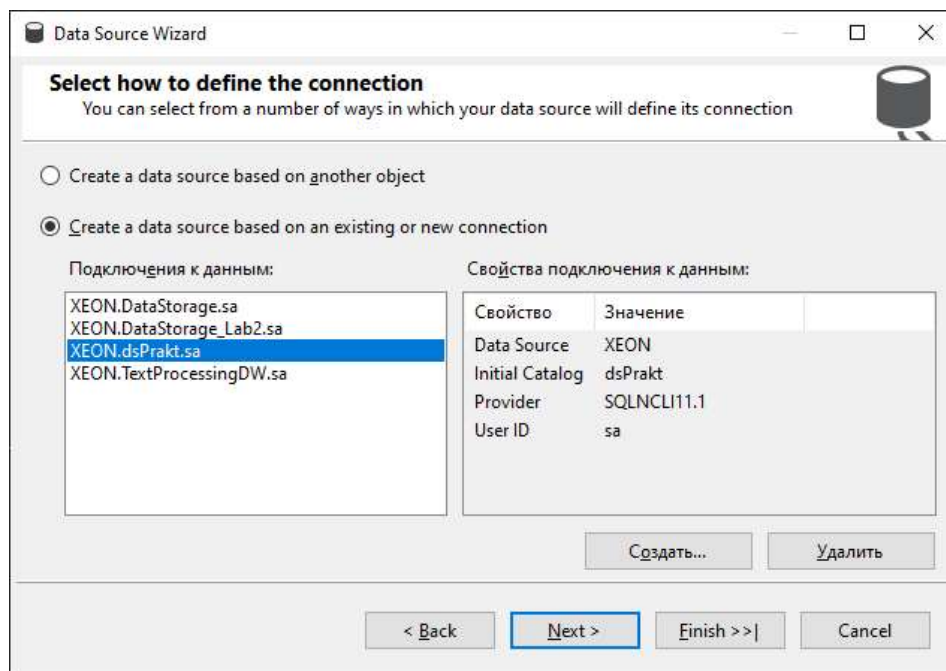


Рис. 9 Додавання джерела даних

Далі потрібно створити уявлення джерела даних. Data Source View (DSV) є важливою частиною побудови куба. Це віртуальна схема, яка визначає, як дані з джерела будуть використані в процесі побудови куба. У DSV можна вибрати таблиці та визначити відносини між ними. Це дозволяє об'єднати дані з різних джерел і спростити процес аналізу. На рис. 10 - 11 зображено створення уявлення джерела даних.

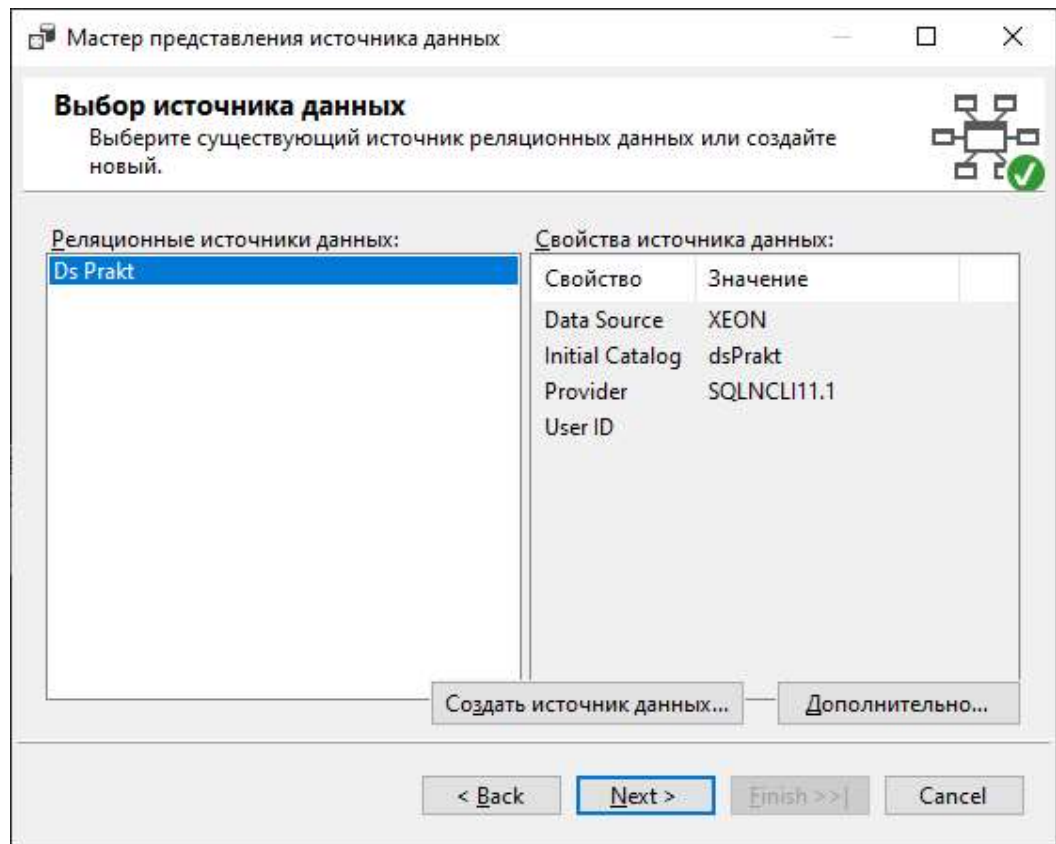


Рис. 10 Створення уявлення джерела даних ч.1

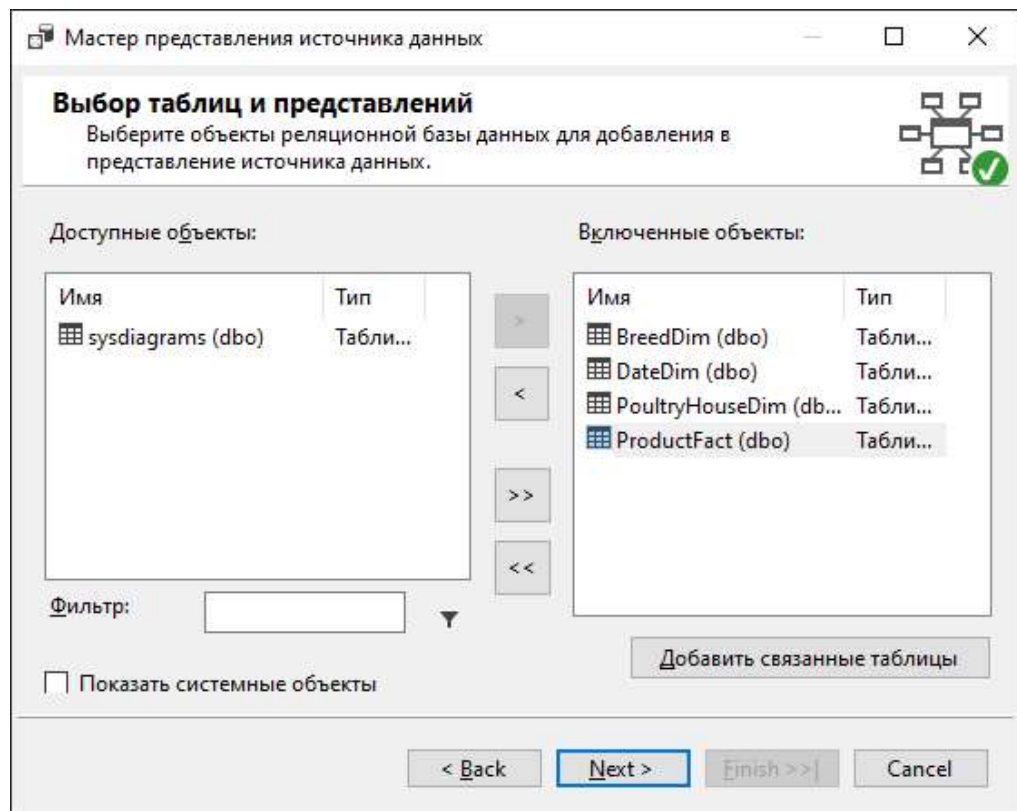


Рис. 11 Створення уявлення джерела даних ч.2

Після налаштування увявлення джерела даних можна переходити до створення вимірів та куба.

Виміри є основою для багатовимірного аналізу даних. Вимір — це категорія, за якою здійснюється групування або агрегація даних. Для кожного виміру створюються окремі таблиці, які містять інформацію, таку як час, географічне місце, клієнти тощо. Вимір включає в себе кілька атрибутів, які забезпечують деталізацію даних. На рис. 12 зображено процес створення виміру BreedDim.

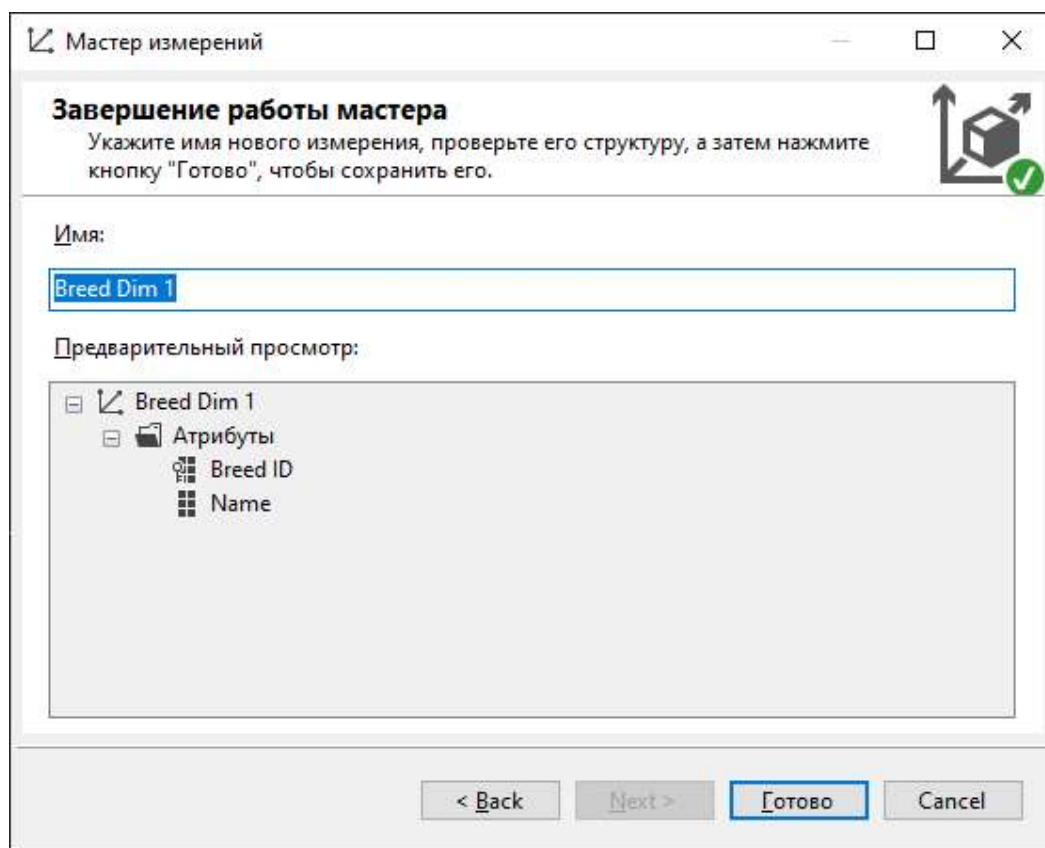


Рис. 12 Створення виміру BreedDim

Наступним кроком є створення кубу OLAP. При створенні куба спочатку вибирається Data Source View, що містить джерела даних для куба. Потім визначаються Measure Groups, які містять факти (числові дані), а також додаються Dimensions, що містять категорії для агрегації даних. Після налаштування всіх вимірів і груп, куб компілюється і готовий до використання для аналізу даних. На рис. 13 - 16 зображено створення куба.

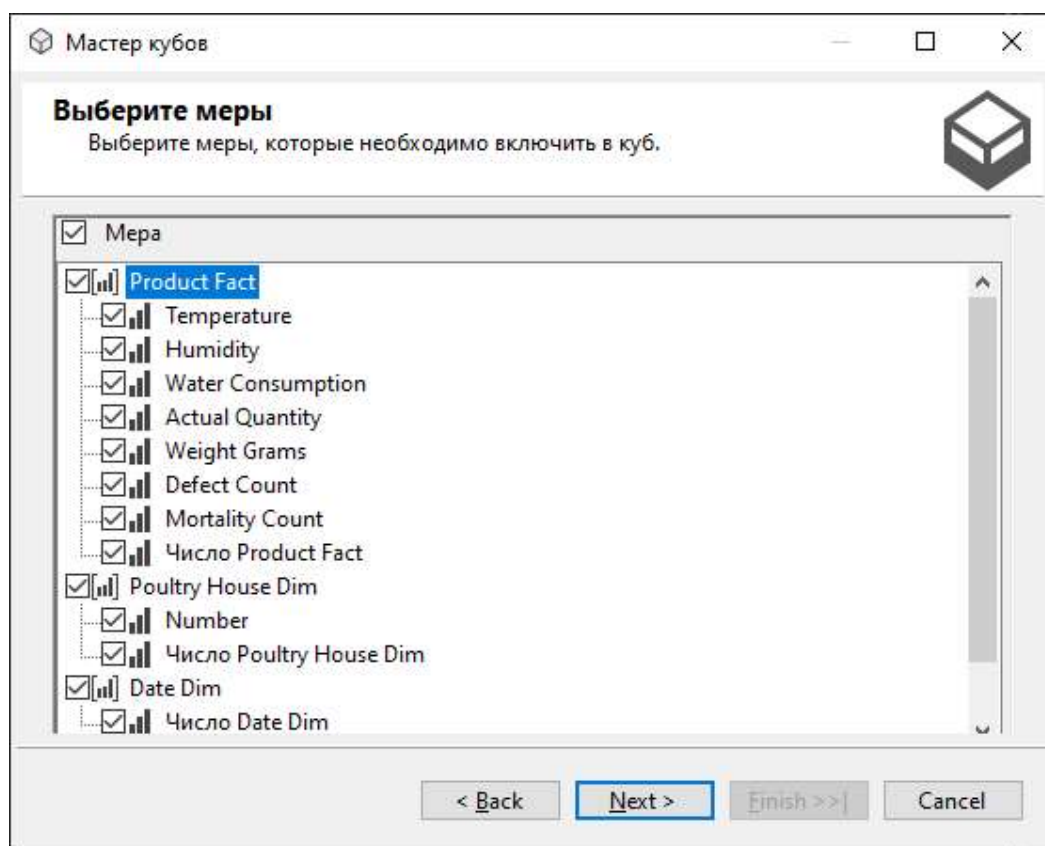
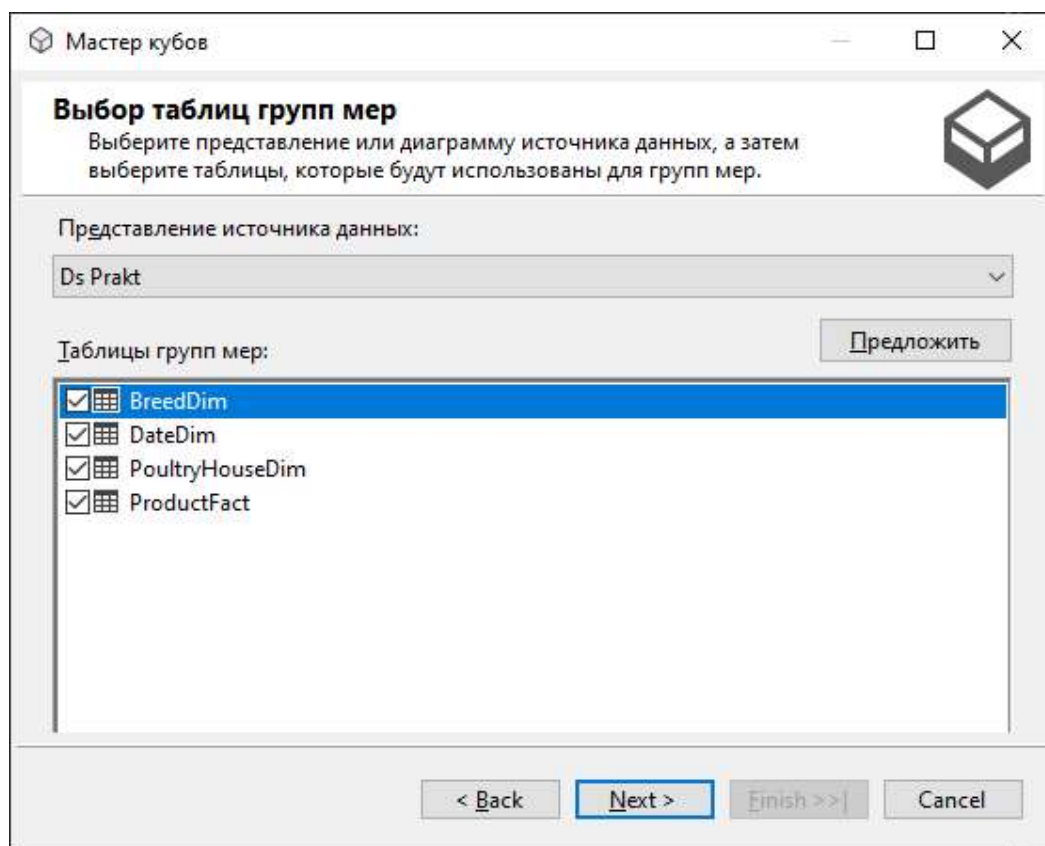


Рис. 13-14 Створення кубу ч. 1-2

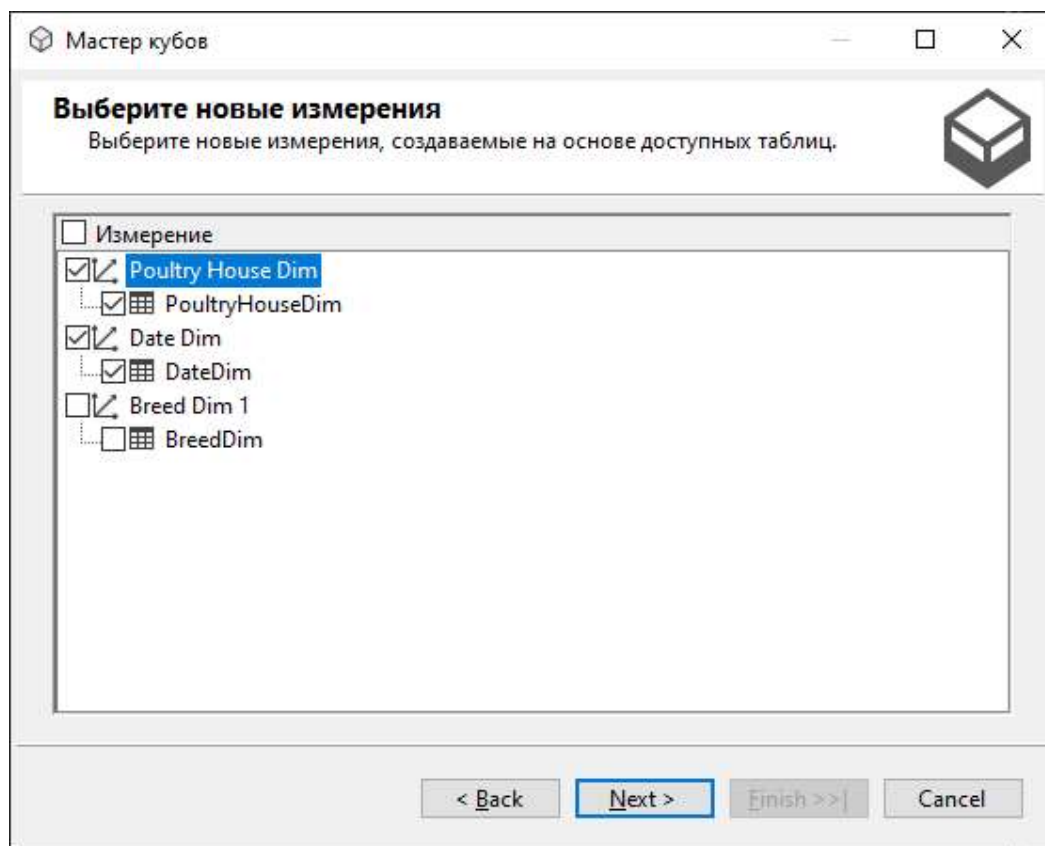
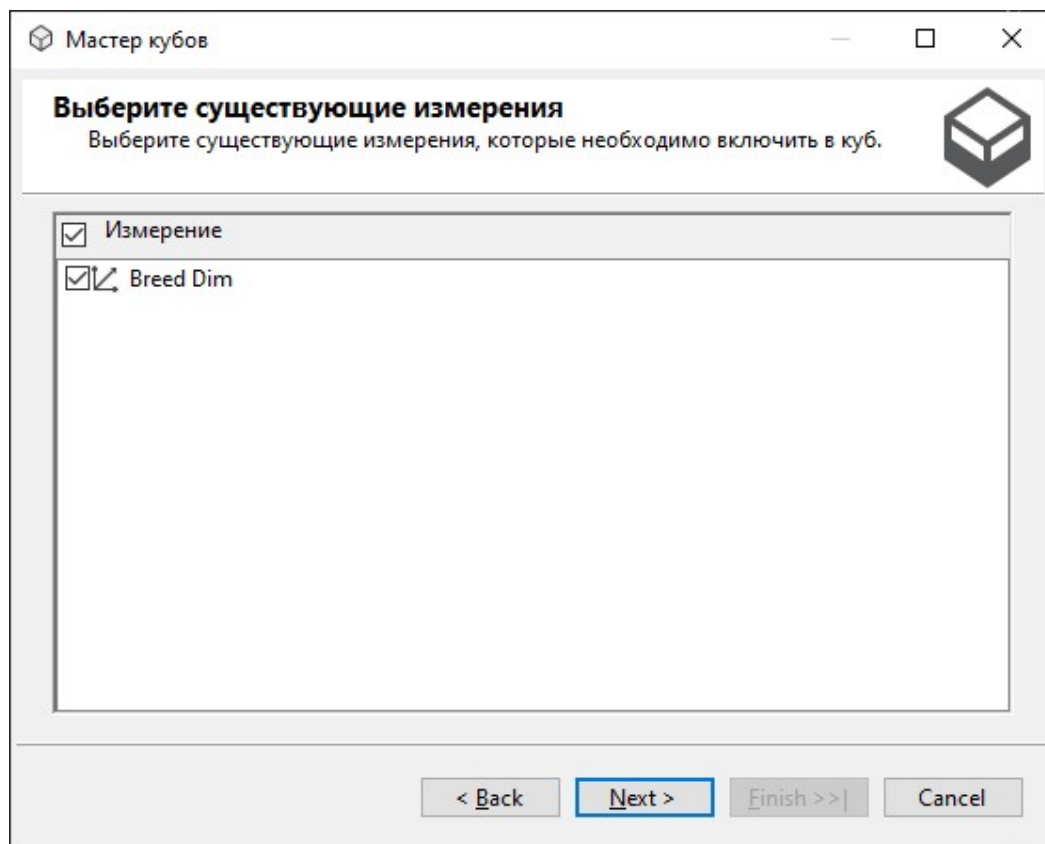


Рис. 15-16 Створення кубу ч. 3-4

Результат створення куба зображено на рис. 17.

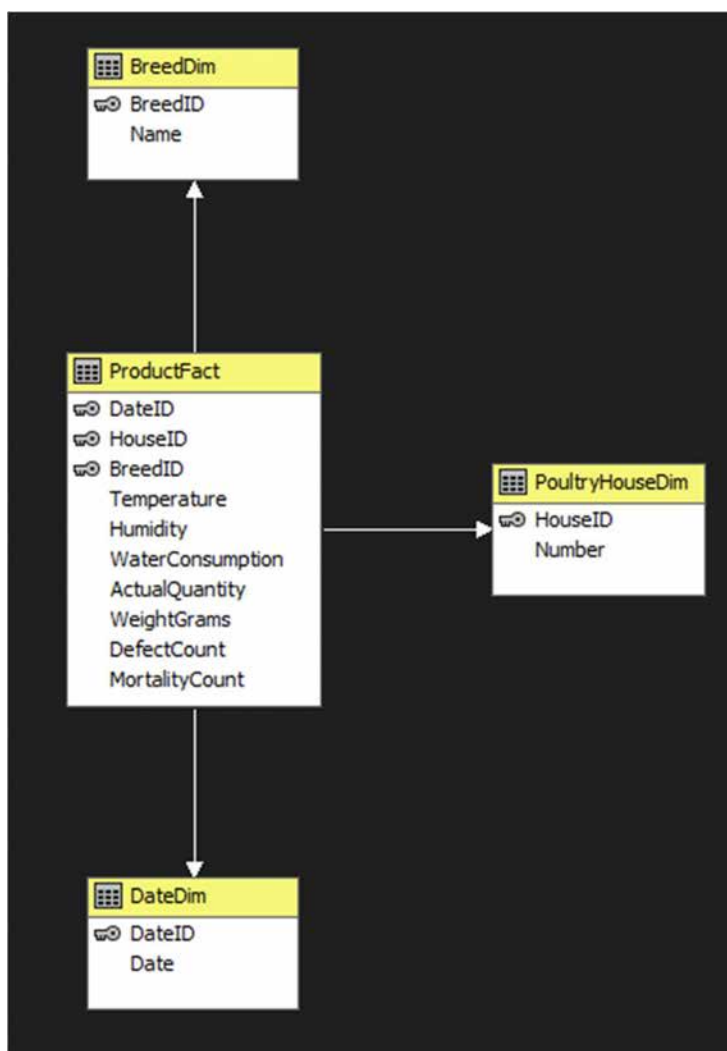


Рис. 17 Створений куб

Процес передачі даних було реалізовано за допомогою служби SQL Server Integration Services. SQL Server Integration Services (SSIS) — це потужний інструмент для інтеграції та трансформації даних у середовищі SQL Server. Він дозволяє виконувати завдання з імпорту, експорту, обробки та трансформації даних між різними джерелами. SSIS підтримує автоматизацію процесів інтеграції даних та їх переміщення, забезпечуючи гнучкість і продуктивність.

Наповнення сховища даних відбувається на основі оперативної бази даних і поділено на два етапи (рис. 18).



Рис. 18 Етапи заповнення СД

На першому етапі йде заповнення вимірів. На рис. 19 зображено потоки, які реалізують передачу даних з БД у СД.

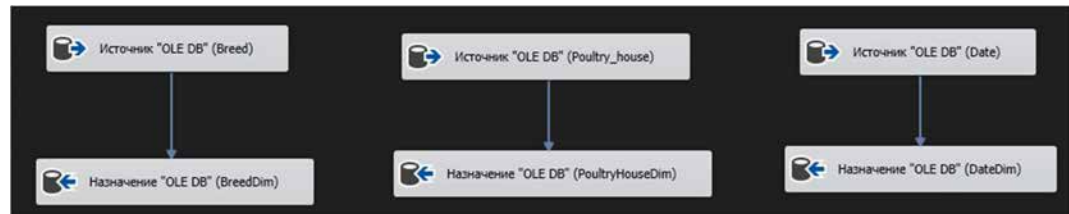


Рис. 19 Потоки заповнення вимірів

На останньому етапі йде заповнення таблиці фактів. На рис. 20-21 зображено потоки цього етапу (SQL код надано у Додатку А).



Рис. 20 Заповнення фактів ч.1

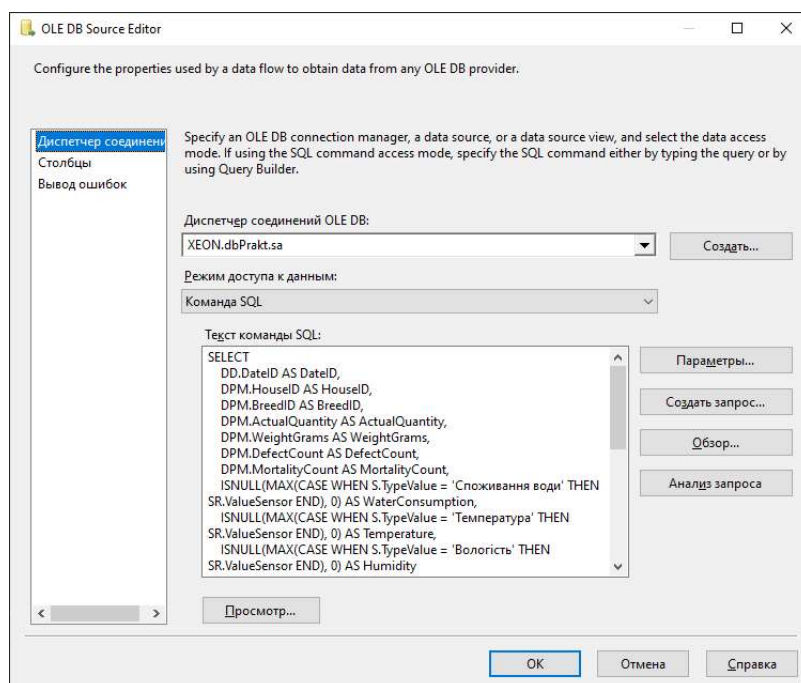


Рис. 21 Заповнення фактів ч.2

У результаті виконання всіх потоків у таблицях СД наявні дані, які перенесені з БД. На рис. 22-25 зображено результат виконання процесу ETL.

DateID	Date
1	2024-01-13
2	2024-01-14
3	2024-01-15
4	2024-01-16
5	2024-01-17
6	2024-01-18
7	2024-01-19
8	2024-01-20
9	2024-01-21

HouseID	Number
1	2
2	3

BreedID	Name
1	COBB - 500

Рис. 22-24 Дані, занесені у СД ч.1

DateID	HouseID	BreedID	Temperature	Humidity	WaterConsumption	ActualQuantity	WeightGrams	DefectCount	MortalityCount
1	1	1	33.50000	45.00000	0.00000	35000	43.00	0	0
2	1	2	33.50000	45.00000	0.00000	35000	43.00	0	0
3	2	1	33.20000	50.00000	1060.00000	35000	0.00	0	40
4	2	1	33.30000	45.00000	924.00000	35000	0.00	0	38
5	3	1	32.50000	50.00000	1140.00000	34960	0.00	0	33
6	3	2	32.40000	44.00000	1170.00000	34962	0.00	0	48
7	4	1	32.10000	54.00000	1460.00000	34927	0.00	0	25
8	4	2	31.90000	45.00000	1480.00000	34914	0.00	0	49
9	5	1	31.60000	61.00000	1900.00000	34902	110.00	0	28
10	5	2	31.70000	48.00000	1920.00000	34865	108.00	0	53
11	6	1	31.40000	63.00000	2080.00000	34874	0.00	0	34
12	6	2	31.30000	51.00000	2190.00000	34812	0.00	0	17
13	7	1	30.20000	56.00000	2160.00000	34840	0.00	0	18
14	7	2	30.60000	55.00000	2240.00000	34795	0.00	0	15
15	8	1	27.50000	53.00000	2270.00000	34822	180.00	0	15
16	8	2	30.90000	54.00000	2340.00000	34780	182.00	0	13
17	9	1	30.20000	55.00000	2650.00000	34807	0.00	0	10
18	9	2	29.70000	55.00000	2570.00000	34767	0.00	0	11
19	10	1	28.80000	58.00000	3150.00000	34797	0.00	0	12

Рис. 25 Дані, занесені у СД ч.2

### 3.5 Модуль аналітики даних

Доступ до сховища даних здійснюється через робочу станцію аналітика. На ній розгорнуто спеціалізований модуль аналітичної обробки, який реалізує механізми багатовимірного аналізу та інтелектуального виявлення закономірностей у даних. Цей модуль забезпечує можливість перегляду агрегованих показників, деталізації інформації за окремими часовими інтервалами, порівняння умов утримання у різних виробничих циклах та оцінки впливу окремих параметрів мікроклімату на продуктивність птиці.

Оскільки обсяг даних, що надходить із пташника, є значним, і він має часову структуру, для їх ефективної обробки та аналізу в системі використовується поєднання технологій OLAP та Data Mining.

OLAP — це технологія обробки даних, яка дозволяє користувачам здійснювати складний аналітичний обробіток великих обсягів даних у реальному часі. Технологія OLAP дає змогу здійснювати швидкий доступ до даних і отримувати на їх основі інформацію для підтримки ухвалення управлінських рішень. Вона підтримує багатовимірну модель даних, що дозволяє ефективно працювати з великими масивами даних, зокрема для задач бізнес-аналізу та звітності.

OLAP-системи можна класифікувати так [14]:

1. ROLAP (Relational OLAP) — використовує реляційні бази даних для виконання запитів і агрегації даних.
2. MOLAP (Multidimensional OLAP) — використовує спеціалізовані багатовимірні сховища для зберігання агрегованих даних.
3. HOLAP (Hybrid OLAP) — поєднує переваги ROLAP та MOLAP, дозволяючи зберігати частину даних у реляційній формі, а інші — у багатовимірному форматі.

OLAP-технології базуються на багатовимірній моделі, де дані організовані таким чином, щоб забезпечити зручне та швидке отримання аналітичних результатів. Найбільш важливі особливості OLAP:

- висока швидкість обробки запитів: завдяки попередній агрегації даних і зберіганню їх у багатовимірному вигляді, OLAP забезпечує швидкий доступ до результатів запитів;
- підтримка складних запитів: OLAP дозволяє виконувати складні запити для аналізу даних на різних рівнях деталізації, в тому числі з використанням таких операцій, як агрегація, розрізання, діагоналізація та поворот;
- можливість аналізу даних в різних вимірах: технологія дає змогу організувати дані за різними вимірами (наприклад, час, географія, категорії товарів тощо), що дозволяє здійснювати аналіз з різних точок зору.

Однією з основних особливостей OLAP є багатовимірна модель даних, де інформація зберігається у вигляді кубів. У такій моделі дані представлені як багатовимірні матриці, де кожен вимір є однією з осей куба. Наприклад, у бізнесі для аналізу продажів дані можуть бути організовані за вимірами часу (рік, місяць, день), регіоном (країна, місто), продуктом (категорія, марка) та іншими.

Куб OLAP — це основна структура для зберігання та обробки даних в OLAP-системах. Куб містить мірки і виміри, що дозволяє проводити аналіз даних в різних контекстах. Наприклад, куб для аналізу продажів може містити такі мірки, як сума продажів і кількість одиниць, а також виміри, такі як час, регіон і продукт.

У рамках OLAP кубу є основним інструментом для агрегування і фільтрації даних, що дозволяє швидко отримувати потрібні аналітичні результати.

Мірки (Measures) — це кількісні показники, які можна виміряти та аналізувати. Прикладом може бути прибуток, обсяг продажів, кількість товарів

тощо. Мірки є основними елементами аналізу в OLAP, оскільки вони містять значення, з якими користувачі працюють під час аналізу даних.

Виміри (Dimensions) — це категорії, за якими можна групувати або організувати дані. Виміри визначають різні аспекти чи характеристики, що допомагають описати мірки. Наприклад, для аналізу даних про продажі виміри можуть включати:

- час (місяць, квартал, рік);
- продукти (категорія, бренд, тип товару);
- географія (країна, регіон, місто).

Використання вимірів допомагає здійснювати аналіз за різними аспектами, наприклад, порівняння продажів у різних країнах або по різних місяцях [15].

Основними перевагами застосування OLAP у даній системі є:

- можливість агрегування показників мікроклімату за вибраними часовими або просторовими періодами;
- швидкий перегляд статистичних підсумків у різних розрізах без потреби виконання складних SQL-запитів;
- підтримка візуалізації даних у вигляді динамічних графіків, зведених таблиць і діаграм;
- інтеграція з аналітичними звітами в середовищі SQL Server Reporting Services (SSRS).

Для реалізації OLAP-рішень у системі було обрано платформу Microsoft SSAS, яка забезпечує побудову багатовимірних моделей, оптимізацію запитів і гнучке керування доступом до даних.

Data Mining — це процес виявлення корисних, раніше невідомих, неочевидних та потенційно значущих закономірностей, залежностей і трендів у великих обсягах даних. Цей процес базується на поєднанні методів статистики, машинного навчання, штучного інтелекту та технологій управління базами даних.

Основною метою Data Mining є перетворення сирих даних у зрозумілу та структуровану інформацію, яка може бути використана для прийняття зважених рішень, точного прогнозування майбутніх подій або оптимізації бізнес-процесів. Наприклад, в маркетингу це може бути виявлення схильності певної групи клієнтів до придбання конкретного продукту, а в медицині — прогноз розвитку захворювання на основі аналізу історичних даних пацієнтів.

На відміну від традиційного аналізу даних, який зазвичай передбачає формулювання гіпотези та подальшу її перевірку, Data Mining орієнтований на автоматизований пошук знань без чітко заданих припущень. Алгоритми здатні самостійно досліджувати великі масиви інформації, виявляючи:

- стійкі закономірності і кореляції між ознаками (наприклад, взаємозв'язок між віком клієнта та його кредитною історією);
- аномальні або нетипові споживацькі поведінки, що можуть свідчити про шахрайство чи збої в процесах;
- приховані групи чи кластери об'єктів, що відрізняються за набором характеристик;
- правила асоціації, які описують ймовірність одночасної появи різних подій або атрибутів.

Процес Data Mining зазвичай включає кілька послідовних етапів:

1. Збір і інтеграція даних. Дані можуть надходити з різних джерел: реляційних баз, XML-файлів, сенсорних мереж чи CRM-систем. Інтеграція забезпечує узгодженість форматів та єдину структуру для аналізу.
2. Попередня обробка — критично важливий етап, що включає очищення (видалення дублікатів і виправлення помилок), заповнення пропусків, нормалізацію та відбір релевантних ознак.
3. Вибір методів та алгоритмів залежно від завдання: класифікація (для прогнозування категорій), регресія (для непрямих числових

значень), кластеризація (для групування даних без заданих міток), асоціативний аналіз, виявлення аномалій тощо.

4. Моделювання полягає у навчанні обраної моделі на тренувальному наборі даних із подальшим налаштуванням гіперпараметрів.
5. Оцінка здійснюється за допомогою метрик (точність, повнота, F1-міра, MSE, ROC-AUC), перевірки на валідаційному та тестовому наборах.
6. Інтерпретація та впровадження знань — результати аналізу перекладають у зрозумілі бізнес-правила, інтегрують у ВІ-дашборди або автоматизовані системи прийняття рішень.

Технологія Data Mining є складовою ширшого процесу Knowledge Discovery in Databases (KDD), що охоплює весь шлях від розуміння бізнес-задачі до розгортання моделей у робочому середовищі. Сьогодні її можливості активно використовуються в банківській справі для скорингу позичальників, в електронній комерції для рекомендацій товарів, у телекомунікаціях для прогнозування відтоку абонентів, у медицині для діагностики та персоналізації лікування, а також у промисловості для передиктивного обслуговування обладнання. Завдяки поєднанню гнучких алгоритмів і масивів обчислювальних ресурсів, Data Mining відкриває шлях до глибоких інсайтів у будь-якій предметній області [16].

У межах дослідження використовуються такі алгоритми Data Mining:

1. 1-Rule (OneR) — простий, але ефективний метод класифікації, що дозволяє встановити, які параметри мікроклімату найбільше впливають на рівень смертності або продуктивності птиці.
2. Naive Bayes — алгоритм байєсівської класифікації, який дає можливість оцінювати ймовірність виникнення певних подій (наприклад, погіршення мікроклімату) на основі історичних даних.
3. Асоціативні правила (Association Rules) — використовуються для виявлення спільних закономірностей між параметрами середовища

(наприклад, підвищення вологості часто супроводжується збільшенням концентрації CO<sub>2</sub>).

4. Кластеризація (Clustering) — дозволяє групувати пташники або періоди часу за подібністю умов мікроклімату, що допомагає виявити типові сценарії або відхилення.

## 4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

### 4.1 Методи Data Mining

**4.1.1 Алгоритм 1-Rule.** Першим методом який було застосовано для інтелектуального аналізу, є алгоритм 1-Rule.

1-Rule (One Rule) — це один із найпростіших алгоритмів машинного навчання, який використовується для задач класифікації. Основна ідея цього методу полягає у створенні одного простого правила для кожного з вхідних атрибутів, після чого обирається те правило, яке забезпечує найменшу кількість класифікаційних помилок. Правило будується шляхом аналізу частоти появи класів цільової змінної при кожному значенні конкретного атрибута. Таким чином, кожен запис у наборі даних буде віднесений до того класу, який найчастіше зустрічається для відповідного значення ознаки.

Незважаючи на свою простоту, алгоритм 1-Rule є достатньо ефективним у задачах, де один з атрибутів має сильний вплив на цільову змінну. Метод дозволяє швидко отримати базову модель, яка може слугувати як для оцінки важливості ознак, так і для побудови інтерпретованої системи правил, придатної для прийняття рішень. У багатьох випадках 1-Rule може досягати точності, співставної з більш складними моделями, особливо коли дані містять чітко виражену кореляцію між атрибутами і класами [17].

У рамках реалізації класифікаційного аналізу в даній роботі було застосовано підхід 1-Rule для оцінки рівня смертності птахів (код виконання надано у Додатку Б). Метою є побудова простої, але інформативної моделі, яка б дозволила на основі одного параметра (або пташника, або класу кліматичних умов) класифікувати його на категорії смертності. Такий підхід дозволяє отримати швидкий аналітичний зріз по найбільш значущих показниках, не використовуючи складні методи машинного навчання.

Було визначено два цільові класи для класифікації рівня смертності:

1. Клас "Висока смертність" — якщо значення смертності більше за середнє значення по всіх записах.
2. Клас "Низька смертність" — якщо значення смертності менше за середнє значення по всіх записах.

Залежна змінна: клас смертності.

Незалежні змінні: пташник та кліматичні умови.

Алгоритм реалізується у кілька кроків:

1. Попередня класифікація: спочатку для кожного окремого запису даних визначається його "фактичний" клас смертності ("Висока смертність" або "Низька смертність") шляхом порівняння конкретного значення смертності із загальним середнім значенням смертності по всьому набору даних.
2. Формування правил для кожного атрибута:
  - для кожного унікального пташника підраховується, скільки разів він зустрічався з класом "Висока смертність" і скільки з класом "Низька смертність". Правилком для цього пташника стає той клас, який зустрічався частіше;
  - аналогічно, для кожного унікального класу кліматичних умов (Гарні, Нормальні, Погані) підраховується, скільки разів він зустрічався з класом "Висока смертність" і скільки з класом "Низька смертність". Правилком для цього класу умов стає той клас, який зустрічався частіше.
3. Вибір найкращого правила (1-Rule): після формування наборів правил для кожного з незалежних атрибутів (пташник, кліматичні умови), обирається той атрибут, чиї правила показують найменшу кількість помилок класифікації на всьому навчальному наборі даних. Цей атрибут і його правила складають "1-Rule" модель.

Таким чином, "1-Rule" модель дозволяє визначити, чи є "пташник" або "кліматичні умови" більш значущим фактором для прогнозування рівня смертності, і створює просте правило для кожного значення цього фактора.

На рис. 26 зображено результат виконання алгоритму.

1-Rule		Байес				
	PoultryHouseNum	MortalityLessOverall	MortalityGreaterOve	OverallAvgMortality	Classification	
▶	2	30	15	20.211	Низька смертні...	
	3	27	18	20.211	Низька смертні...	

	ConditionClass	RecordCount	MortalityLessOverall	MortalityGreaterOve	OverallAvgMortality	Classification
▶	t < 25.0M    t > 3...	44	23	21	20.211	Низька смертні...
	Other	26	18	8	20.211	Низька смертні...
	t >= 28.0M && t ...	20	16	4	20.211	Низька смертні...

Рис. 26 Результат виконання 1-Rule

Для визначення цільових класів "Висока смертність" та "Низька смертність" було встановлено загальне середнє значення смертності по всіх записах, яке склало 20.211.

При аналізі даних за номером пташника були отримані такі результати:

1. Пташник "2": для цього пташника зафіксовано 30 випадків, коли смертність була меншою за загальне середнє значення, і 15 випадків, коли смертність перевищувала середнє. На основі цих даних пташник "2" був класифікований як такий, що має "Низьку смертність".
2. Пташник "3": для пташника "3" було виявлено 27 випадків смертності нижче середнього показника та 18 випадків смертності вище середнього. Відповідно, пташник "3" також був класифікований як такий, що має "Низьку смертність".

При аналізі даних за класом кліматичних умов (визначених на основі показників температури та вологості) були отримані такі класифікації:

1. "Погані умови" (t < 25.0M || t > 35.0M || hum < 40.0M || hum > 80.0M): загалом зареєстровано 44 записи для цього класу умов. З них 23 випадки мали смертність нижче загального середнього, а 21 випадок

— вище середнього. Цей клас умов був класифікований як такий, що асоціюється з "Низькою смертністю".

2. "Нормальні умови" (Other): для нормальних умов зафіксовано 26 записів. 18 з них демонстрували смертність нижче середнього, тоді як 8 випадків — вище середнього. Клас "Нормальні умови" також був класифікований як такий, що має "Низьку смертність".
3. "Гарні умови" ( $t \geq 28.0M$  &&  $t \leq 32.0M$  &&  $hum \geq 50.0M$  &&  $hum \leq 70.0M$ ): цей клас включав 20 записів. 16 випадків мали смертність нижче середнього, і лише 4 — вище середнього. "Гарні умови" були класифіковані як такі, що асоціюються з "Низькою смертністю".

Аналіз результатів виконання алгоритму 1-Rule показує, що для обох незалежних змінних (номер пташника та клас кліматичних умов) та їхніх унікальних значень, переважаючим класом смертності є "Низька смертність". Це свідчить про те, що у більшості спостережень рівень смертності був нижчим за загальний середній показник, встановлений для всього набору даних.

Це означає, що для реальної оцінки стану пташників і прийняття управлінських рішень недоцільно обмежуватися використанням лише одного параметра — необхідно застосовувати методи, що враховують взаємодію декількох ознак. Для цього у даному дослідженні наступним етапом було використано метод Наївного Байєса.

**4.1.2 Метод Наївного Байєса.** Це один з найпростіших та водночас ефективних статистичних методів машинного навчання, який ґрунтується на використанні теореми Байєса з припущенням незалежності ознак. Його часто застосовують для задач класифікації завдяки високій швидкості роботи, низьким вимогам до обчислювальних ресурсів та хорошим результатами навіть при обмеженій кількості даних. Метод добре масштабується на великі обсяги інформації та легко інтерпретується, що робить його популярним у різних галузях, таких як спам-фільтрація, аналіз текстів, медична діагностика тощо.

У даній роботі метод Наївного Байєса використовується для розв'язання задачі класифікації птахів за рівнем їхньої смертності (код виконання надано у

Додатку В). Для цього критерію в наборі даних визначено два цільові класи: "Висока смертність" (вище за середнє значення смертності по всіх записах) та "Низька смертність" (нижче за середнє значення смертності по всіх записах). Класифікація дозволяє автоматично визначати, до якого з цих класів належить кожен птах на основі заданих ознак. Як вхідні характеристики (незалежні змінні) для класифікації було обрано пташник, у якому утримуються птахи, та кліматичні умови (температура, вологість) у цьому пташнику.

Метод Наївного Байеса реалізується у кілька послідовних кроків. На першому етапі проводиться підготовка та перетворення даних: із бази даних за допомогою ADO.NET витягуються значення ознак (ідентифікатор пташника, температура, вологість) та цільової змінної (рівень смертності). Далі для кожного запису даних визначається його "фактичний" клас смертності ("Висока смертність" або "Низька смертність") шляхом порівняння конкретного значення смертності із загальним середнім значенням смертності по всьому набору даних. Після цього всі записи групуються відповідно до їхнього фактичного класу.

Після підготовки даних виконується обчислення ймовірностей:

1. Априорні ймовірності: Для кожного цільового класу ("Висока смертність" та "Низька смертність") визначається априорна ймовірність — частка об'єктів (записів) у навчальному наборі, що належать до цього класу.
2. Умовні ймовірності: Для кожного унікального значення кожної ознаки (наприклад, для кожного ідентифікатора пташника, або для кожного класу кліматичних умов — "Гарні", "Нормальні", "Погані") розраховується умовна ймовірність появи цієї ознаки у межах кожного класу ("Висока смертність" або "Низька смертність"). Для уникнення нульових ймовірностей (які можуть виникнути, якщо певна комбінація ознаки та класу не зустрічалася у навчальних даних) застосовується згладжування Лапласа (Laplace smoothing). Таким чином формується таблиця умовних ймовірностей, яка

відображає, наскільки характерна та чи інша ознака для кожної категорії смертності.

Далі алгоритм проходить по всіх можливих комбінаціях значень незалежних ознак (кожному пташнику та кожному класу кліматичних умов) та для кожної такої комбінації обчислює апостеріорну ймовірність належності до кожного класу смертності ("Висока смертність" та "Низька смертність") на основі раніше обчислених апріорних та умовних ймовірностей (за формулою Байєса). Комбінація пташника та кліматичних умов класифікується до того класу смертності, для якого отримана більша апостеріорна ймовірність [18].

На рис. 27 зображено результат виконання алгоритму.

1-Rule		Байєс			
	PoultryHouse	ConditionClass	P(H Features)	P(L Features)	Predicted_Class
▶	2	Нормальні умови	0.28452	0.71548	L
	2	Гарні умови	0.19802	0.80198	L
	2	Погані умови	0.43489	0.56511	L
	3	Нормальні умови	0.34333	0.65667	L
	3	Гарні умови	0.24507	0.75493	L
	3	Погані умови	0.50292	0.49708	H
*					

Рис. 27 Результат виконання методу Наївного Байєса

Відповідно до наданого зображення, що демонструє результати виконання методу Наївного Байєса, було здійснено класифікацію рівня смертності ("H" - висока, "L" - низька) на основі комбінації пташника та кліматичних умов.

Для пташника "2" спостерігаються такі прогнози:

1. У "Нормальних умовах" ймовірність високої смертності ( $P(H|Features)$ ) становить 0.28452, тоді як ймовірність низької смертності ( $P(L|Features)$ ) значно вища — 0.71548. Отже, прогнозований клас смертності для пташника "2" у нормальних умовах — низький ("L").
2. У "Гарних умовах" ймовірність високої смертності падає до 0.19802, а ймовірність низької смертності зростає до 0.80198. Це також

призводить до прогнозу низької смертності ("L") для пташника "2" у гарних умовах.

3. Навіть у "Поганих умовах" для пташника "2" ймовірність високої смертності (0.43489) залишається нижчою за ймовірність низької смертності (0.56511), що вказує на прогноз низької смертності ("L").

Для пташника "3" результати відрізняються:

1. У "Нормальних умовах" для пташника "3" ймовірність високої смертності становить 0.34333, а ймовірність низької смертності — 0.65667. Прогнозований клас смертності — низький ("L").
2. У "Гарних умовах" ймовірність високої смертності для пташника "3" становить 0.24507, тоді як ймовірність низької смертності значно вища — 0.75493. Це також призводить до прогнозу низької смертності ("L").
3. Проте, у "Поганих умовах" для пташника "3" спостерігається переломний момент: ймовірність високої смертності становить 0.50292, що трохи перевищує ймовірність низької смертності (0.49708). Таким чином, для пташника "3" у поганих умовах прогнозується висока смертність ("H").

Аналіз результатів Наївного Байєса показує, що для пташника "2" у всіх типах кліматичних умов (нормальні, гарні, погані) прогнозується низький рівень смертності. Водночас, для пташника "3" рівень смертності також прогнозується низьким у нормальних та гарних умовах, але підвищується до високого рівня при несприятливих кліматичних умовах. Це вказує на потенційну вразливість пташника "3" до погіршення кліматичних умов порівняно з пташником "2".

Такий результат є більш інформативним для прийняття управлінських рішень, оскільки дозволяє визначити не лише рівень ризику, але й умови, за яких цей ризик зростає. Отже, метод Наївного Байєса забезпечує більш точну та обґрунтовану класифікацію порівняно з простими однофакторними підходами.

Разом з тим, отримані результати не дають відповіді на питання про конкретні комбінації параметрів, які найчастіше зустрічаються разом із

підвищеною смертністю та можуть вважатися критичними. Для виявлення таких повторюваних закономірностей у наборі даних доцільно застосувати метод асоціативних правил.

**4.1.3 Метод асоціативних правил.** Це важливий підхід у галузі інтелектуального аналізу даних, який дозволяє виявити цікаві залежності, закономірності та структури у великих обсягах даних. Його головною метою є знаходження зв'язків типу «якщо–то» між об'єктами або ознаками, що часто трапляються разом у даних. Наприклад, у сфері роздрібної торгівлі це може бути виявлення правил виду: «якщо покупець придбав хліб і масло, то він, ймовірно, придбає й молоко».

Метод базується на аналізі транзакцій або прикладів, у кожному з яких зафіксовано наявність певних атрибутів (ознаки або характеристики). В результаті аналізу будується набір асоціативних правил, кожне з яких має вигляд "якщо X, то Y", де X — ліва частина правила (*antecedent*), а Y — права частина (*consequent*). Правило інтерпретується як: якщо в прикладі присутні всі елементи з множини X, то з великою ймовірністю в ньому також будуть присутні елементи з Y.

Для оцінки сили та значущості правил використовують такі основні показники:

1. Підтримка (*support*) — частка записів у наборі даних, які одночасно містять X та Y. Вона показує, наскільки поширене правило в цілому.
2. Впевненість (*confidence*) — умовна ймовірність того, що Y буде присутнім, якщо вже відоме X. Це міра точності правила.
3. Захоплення (*lift*) — співвідношення довіри до ймовірності появи Y незалежно від X. Показує, наскільки сильний вплив має X на появу Y у порівнянні з випадковим розподілом.

Процес формування асоціативних правил складається з кількох етапів: підготовки даних, генерації частих наборів ознак, побудови асоціативних правил та їх подальшої фільтрації й оцінки [19]. У даному дослідженні ці етапи були реалізовані за допомогою бібліотеки *mlxtend* у середовищі Python, а вхідні дані

були завантажені безпосередньо з таблиці ProductFact (код виконання надано у Додатку Д).

Підготовка даних включала категоризацію вхідних параметрів. Було обчислено медіанні значення для кожного з мікрокліматичних параметрів: температура та вологість. На їх основі створено бінарні ознаки, які відображають перевищення або недосягнення медіанних рівнів. Аналогічним чином категоризовано параметри смертності птиці.

Після підготовки даних було створено булеву матрицю ознак, яка стала основою для побудови частих наборів елементів за допомогою алгоритму Apriori. Порог підтримки було встановлено на рівні 0.05, що дозволяє виявити всі набори ознак, які зустрічаються принаймні у 5% випадків.

На основі знайдених частих наборів було згенеровано асоціативні правила з мінімальним значенням довіри 0.5. Це дозволяє враховувати лише ті правила, які мають щонайменше 50% ймовірності виконання правої частини при наявності лівої. На завершальному етапі правила було відсортовано та відфільтровано відповідно до дослідницьких цілей.

На рис. 28 зображено сформовані асоціативні правила.

```

=== Асоціативні правила, які були знайдені ===
Правило: {'high_hum'} => {'high_mortality_count'}
Підтримка: 0.3111, Впевненість: 0.6667, Lift: 1.3333
---
Правило: {'low_hum'} => {'low_mortality_count'}
Підтримка: 0.2889, Впевненість: 0.6190, Lift: 1.2381
---
Правило: {'high_temp'} => {'low_mortality_count'}
Підтримка: 0.2778, Впевненість: 0.5556, Lift: 1.1111
---
Правило: {'low_temp'} => {'high_mortality_count'}
Підтримка: 0.2778, Впевненість: 0.5556, Lift: 1.1111
---
Правило: {'high_hum', 'low_temp'} => {'high_mortality_count'}
Підтримка: 0.2667, Впевненість: 0.6857, Lift: 1.3714
---
Правило: {'high_temp', 'low_hum'} => {'low_mortality_count'}
Підтримка: 0.2000, Впевненість: 0.5294, Lift: 1.0588
---
Правило: {'low_temp', 'low_hum'} => {'low_mortality_count'}
Підтримка: 0.0889, Впевненість: 1.0000, Lift: 2.0000
---

```

Рис. 28 Сформовані асоціативні правила

Були ідентифіковані сім ключових правил. Правило "висока вологість" (high\_hum) => "висока смертність" (high\_mortality\_count) має підтримку 0.3111

та впевненість 0.6667, що вказує на те, що у двох третинах випадків високої вологості спостерігається висока смертність, а показник Lift 1.3333 підтверджує позитивну кореляцію. З іншого боку, правило "низька вологість" (low\_hum) => "низька смертність" (low\_mortality\_count) показує підтримку 0.2889 та впевненість 0.6190, з Lift 1.2381, що свідчить про схожий позитивний зв'язок.

Щодо температури, виявлено, що "висока температура" (high\_temp) => "низька смертність" (low\_mortality\_count) має підтримку 0.2778 та впевненість 0.5556, з Lift 1.1111. Водночас, "низька температура" (low\_temp) => "висока смертність" (high\_mortality\_count) демонструє ідентичні показники підтримки (0.2778) та впевненості (0.5556), з Lift 1.1111, що підкреслює їхні протилежні, але схожі за силою впливи.

Аналіз комбінованих факторів показав, що правило "{high\_hum, low\_temp}" (висока вологість та низька температура) => "{high\_mortality\_count}" (висока смертність) має підтримку 0.2667, впевненість 0.6857 та Lift 1.3714, що є одним з найсильніших позитивних зв'язків. Правило "{high\_temp, low\_hum}" (висока температура та низька вологість) => "{low\_mortality\_count}" (низька смертність) показує підтримку 0.2000, впевненість 0.5294 та Lift 1.0588.

Найбільш виразним та показовим виявилось правило "{low\_temp, low\_hum}" (низька температура та низька вологість) => "{low\_mortality\_count}" (низька смертність). Хоча його підтримка є найнижчою (0.0889), що свідчить про відносну рідкість такої комбінації, його впевненість становить 1.0000 (100%), а показник Lift дорівнює 2.0000. Це означає, що при одночасній наявності низької температури та низької вологості завжди спостерігається низька смертність, і ймовірність низької смертності зростає вдвічі порівняно з випадковою подією.

Аналіз асоціативних правил дозволив не лише підтвердити вплив окремих параметрів мікроклімату на рівень смертності, а й виявити комбінації умов, за яких ризик суттєво змінюється. Отримані правила показали, що висока вологість у поєднанні з низькою температурою є найбільш несприятливим фактором, який стійко асоціюється з підвищенням смертності. Навпаки, комбінація низької вологості та високої температури корелює з більш сприятливими умовами та

зниженням ризику. Особливо цінним є те, що метод асоціативних правил дозволив виявити саме стійкі закономірності, які повторюються в різних частинах набору даних, а не лише загальні статистичні тенденції. Таким чином, результати цього етапу можуть бути використані для формування рекомендацій щодо оптимізації умов утримання.

Разом з тим, асоціативні правила не дають відповіді на питання про те, чи існує природне групування записів за характером їх параметрів і чи утворюють вони окремі типові стани середовища. Для цього доцільно застосувати методи кластеризації.

**4.1.4 Кластеризація.** Це метод безконтрольного навчання, який полягає в поділі даних на групи або кластери, в межах яких елементи є схожими між собою за певними характеристиками. Мета кластеризації — це виявлення структур у даних, де кожен кластер складається з об'єктів, які є подібними до одного одного, але відрізняються від об'єктів з інших кластерів. Кластеризація є важливим етапом у багатьох галузях, таких як машинне навчання, обробка зображень, біоінформатика, маркетинг та інші.

Основною метою кластеризації є виявлення природних структур у даних, що дозволяє значно зменшити складність задачі для подальшого аналізу. Наприклад, у задачах сегментації клієнтів в маркетингу кластеризація допомагає визначити групи схожих споживачів, що полегшує адаптацію стратегій продажів для кожної групи. В обробці зображень кластеризація може бути використана для виявлення патернів або текстур в зображеннях, що спрощує їх подальшу обробку [20].

Для реалізації кластеризації в даному дослідженні використовується алгоритм К-середніх (K-Means), який є одним з найпоширеніших і ефективних методів кластеризації.

Алгоритм К-середніх працює за принципом мінімізації суми квадратів відстаней між точками в кластері і його центром. Цей метод є особливо ефективним для задач, де кластери мають форму сферичних областей і дані добре розділені. Ключовою особливістю алгоритму є попереднє визначення

кількості кластерів ( $K$ ), що є одним з його основних обмежень. Однак, після визначення кількості кластерів, алгоритм виконує наступні кроки:

1. Ініціалізація: вибираються випадкові центри кластерів.
2. Призначення: кожна точка даних призначається до найближчого центру кластеру.
3. Оновлення: центри кластерів оновлюються як середнє значення точок, що належать кожному кластеру.
4. Повторення: кроки 2 та 3 повторюються до стабільності, коли центри кластерів перестають змінюватися.

Цей алгоритм швидкий і простий у реалізації, але його ефективність може бути обмежена при наявності кластерів складної форми або якщо дані містять багато шумових точок [21].

Перед застосуванням алгоритмів кластеризації було проведено попередню обробку даних: зі структури було вилучено ідентифікатори (BreedID, HouseID, DateID), що не несуть корисної інформації для кластеризації. Решта числових ознак були масштабовані за допомогою StandardScaler, що дозволяє привести всі показники до однакового масштабу, забезпечуючи коректну роботу методу  $K$ -середніх.

Для виконання кластеризації було використано алгоритм  $K$ -середніх (KMeans) із попереднім підбором оптимальної кількості кластерів на основі силуетного коефіцієнта (код виконання надано у Додатку Е). Для значень від 2 до 10 кластерів було обчислено відповідні коефіцієнти, і за їхнім аналізом було визначено, що оптимальною є кількість 6 (силуетний коефіцієнт = 0.4274). На рис. 29 зображено графік силуетного коефіцієнта при різній  $k$ -сть кластерів.

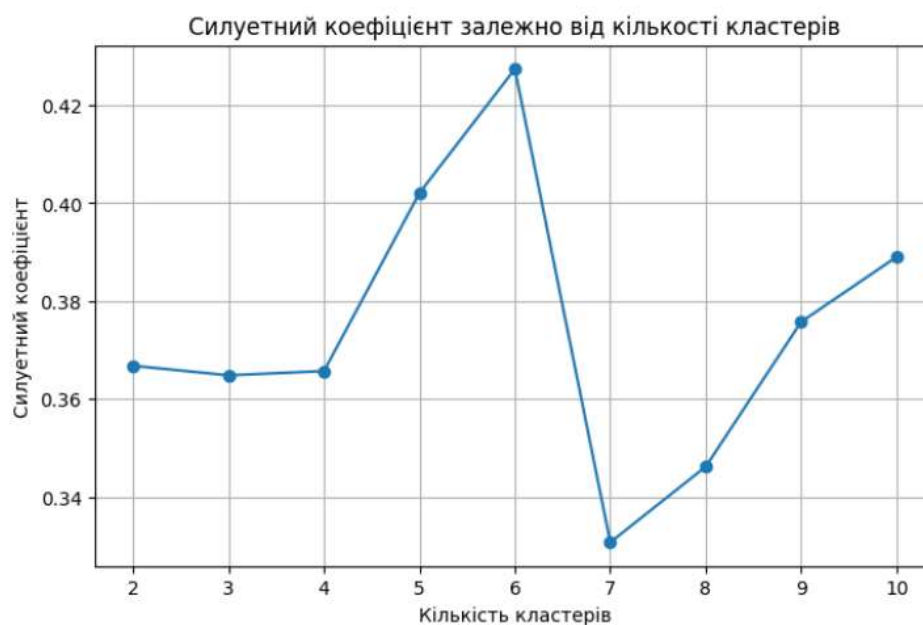


Рис. 29 Графік силуетного коефіцієнта

Після визначення кількості кластерів було здійснено остаточне групування об'єктів у шість кластерів. Для інтерпретації результатів було також виконано понижене відображення даних за допомогою PCA та побудовано графік з візуалізацією кластерів. На рис. 30 зображено графік з візуалізацією кластерів.

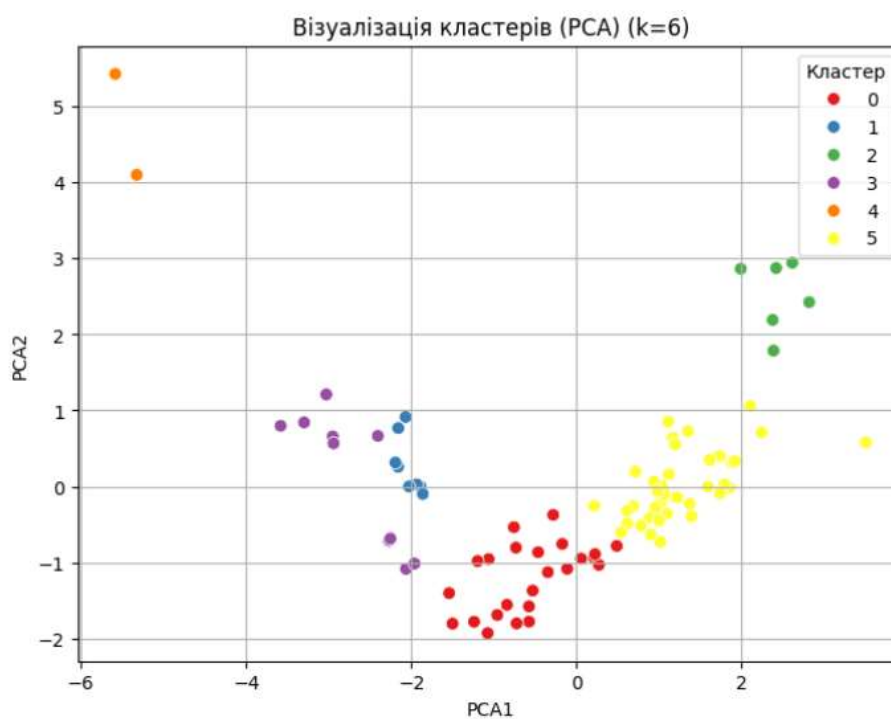


Рис. 30 Графік візуалізації кластерів

Візуалізація кластерів за допомогою PCA з двома основними компонентами (PCA1 та PCA2) підтверджує візуальне розділення даних на 6 груп. На графіку можна побачити, як точки даних, що належать до одного кластера, групуються разом, утворюючи відмінні скупчення, що свідчить про успішну ідентифікацію прихованих патернів у даних.

На рис. 31 зображено розподіл записів по кластерам.

Розподіл записів по кластерам:		Розподіл по кластерам:			
cluster	count	BreedID	HouseID	cluster	count
0	24	0	1	5	21
1	8	1	1	5	19
2	6	2	1	0	12
3	10	3	1	0	12
4	2	4	1	3	5
5	40	5	1	2	3
		6	1	2	2
		7	1	1	4
		8	1	2	1
		9	1	1	2
		10	1	1	4
		11	1	2	1

Name: count, dtype: int64

Рис. 31 Розподіл записів по кластерам

Розподіл записів за цими кластерами виявився наступним: кластер 5 є найбільшим і містить 40 записів, за ним іде кластер 0 з 24 записами, кластер 3 з 10 записами, кластер 1 з 8 записами, кластер 2 з 6 записами, і найменший кластер 4, що містить лише 2 записи.

Детальний розподіл записів по кластерах із зазначенням BreedID та HouseID показав, що, наприклад, 21 запис для BreedID 1 та HouseID 1 потрапив до кластера 5, а 19 записів для BreedID 1 та HouseID 2 також увійшли до кластера 5. Існують також менші групи, такі як 1 запис для BreedID 1 та HouseID 1 у кластері 4.

На рис. 32 зображено центроїди кластерів.

Центроїди кластерів (повністю):

	Temperature	Humidity	WaterConsumption	ActualQuantity	WeightGrams	\
0	-0.617169	0.505987	0.916036	0.373529	-0.240456	
1	-1.309387	0.676718	0.667261	-2.352228	-0.513476	
2	1.575451	-2.023947	-1.619755	0.511272	-0.490093	
3	-1.132765	1.049224	1.071293	-0.543392	2.224167	
4	-1.464601	0.676718	0.499288	-2.838073	2.924492	
5	0.752282	-0.431486	-0.732898	0.447389	-0.381784	

	DefectCount	MortalityCount
0	-0.208155	-0.482157
1	-0.208155	1.009805
2	-0.208155	2.154800
3	0.367248	0.517111
4	6.281812	1.784701
5	-0.208155	-0.454400

Рис. 32 Центроїди кластерів

Аналіз центроїдів кластерів дозволив інтерпретувати типовий профіль кожної групи, відображаючи середні стандартизовані значення ознак:

1. Кластер 0 (24 записи): Цей кластер об'єднує партії з відносно помірною температурою (-0.617169) та позитивною вологістю (0.505987). Вони характеризуються високим споживанням води (0.916036) та позитивною фактичною кількістю продукції (0.373529). Найважливішою ознакою є низький рівень смертності (-0.482157), що вказує на типові, добре керовані умови, які забезпечують низьку смертність.
2. Кластер 1 (8 записів): Характеризується відносно низькою температурою (-1.309387) та помірно високою вологістю (0.676718). Цей кластер відрізняється значно вищим рівнем смертності (1.009805) та негативною фактичною кількістю (-2.352228). Ймовірно, цей кластер об'єднує партії, які стикаються з несприятливими температурно-вологісними умовами, що призводить до значних втрат та низької продуктивності.
3. Кластер 2 (6 записів): Представляє партії з дуже високою температурою (1.575451) та вкрай низькою вологістю (-2.023947). Споживання води є низьким (-1.619755), а рівень смертності є найвищим серед усіх кластерів (2.154800). Цей кластер чітко вказує

на критично несприятливі, можливо, сухі та спекотні, умови, які максимально негативно впливають на виживання птиці.

4. Кластер 3 (10 записів): Для цього кластера характерні низька температура (-1.132765) та висока вологість (1.049224). Відзначаються позитивні значення дефектів (0.367248) та відносно висока смертність (0.517111), незважаючи на значну вагу птиці (2.224167). Це може вказувати на умови, які, хоч і сприяють набору ваги, водночас провокують проблеми зі здоров'ям, що призводить до підвищеної смертності та дефектів.
5. Кластер 4 (2 записи): Цей найменший кластер виділяється екстремально високим показником дефектів (6.281812) та високою смертністю (1.784701). Температура відносно низька (-1.464601), а вологість помірна (0.676718). Цей кластер ідентифікує аномально проблемні партії з високим рівнем падежу та браку, що вимагають негайного втручання.
6. Кластер 5 (40 записів): Найбільший кластер, що характеризується відносно високою температурою (0.752282) та низькою вологістю (-0.431486). Рівень смертності у цьому кластері є низьким (-0.454400). Це представляє найбільш сприятливі або "оптимальні" умови утримання птиці, що призводять до мінімальних втрат.

Аналіз результатів кластеризації показав наявність чітких групових відмінностей у характері мікрокліматичних умов та їх зв'язку з продуктивністю і смертністю птиці. Виявлені кластери демонструють як сприятливі, так і проблемні типи середовищ утримання. Зокрема, кластер 5 та 0 відображають стабільні умови з низькою смертністю та оптимальними значеннями температури та вологості, що може розглядатися як орієнтир для підтримання технологічного режиму. Натомість кластери 1, 2, 3 та 4 вказують на комбінації факторів, які безпосередньо пов'язані з підвищеною смертністю та збільшенням дефектів, при цьому кластер 2 ідентифікує найбільш критичні умови, а кластер 4 — аномальні ситуації, що потребують окремої уваги та оперативного втручання.

Отримані результати свідчать про те, що умови утримання птиці формують не випадкові, а повторювані типові стани, кожен з яких має свій профіль ризику. Це дозволяє не лише оцінювати поточний стан пташників, а й прогнозувати можливі наслідки змін мікроклімату. Таким чином, кластеризація виступає інструментом групової діагностики, доповнюючи результати попередніх методів інтелектуального аналізу та дозволяючи здійснювати більш обґрунтоване планування коригувальних дій.

## 4.2 Методи OLAP

Оскільки OLAP ґрунтується на аналізі даних у багатовимірній формі, завершальним етапом роботи стало застосування відповідних методів до сформованого гіперкуба. Для дослідження були використані засоби побудови звітів, графічних візуалізацій та обчислення ключових показників ефективності.

Для формування звітів було використано SSRS, що дозволило створити порівняльний звіт по пташникам за обраний період. У звіті відображено узагальнену інформацію щодо кількості птиці, об'єму отриманої продукції, споживання води, рівня смертності та відсотка браку (запит створення звіту надано у Додатку Ж). Окремим блоком у звіті представлено значення параметрів мікроклімату (температури та вологості), що дозволяє порівняти умови утримання у різних пташниках. Такий формат звіту забезпечує можливість оперативного аналізу стану виробничих умов та виявлення проблемних зон. Побудований звіт наведено у Додатку И.

Для наочного представлення динаміки втрат було побудовано графік смертності для двох пташників у часі. Графік дозволяє простежити, як змінювався показник смертності у кожному пташнику протягом періоду спостережень. Візуальний аналіз показав, що для пташника №2 рівень смертності залишається стабільнішим і коливається у відносно вузькому діапазоні. У свою чергу, для пташника №3 спостерігаються більш різкі зміни та

періоди підвищеної смертності, що свідчить про меншу стабільність та вищу чутливість до змін умов. Побудований графік наведено на рис. 33.

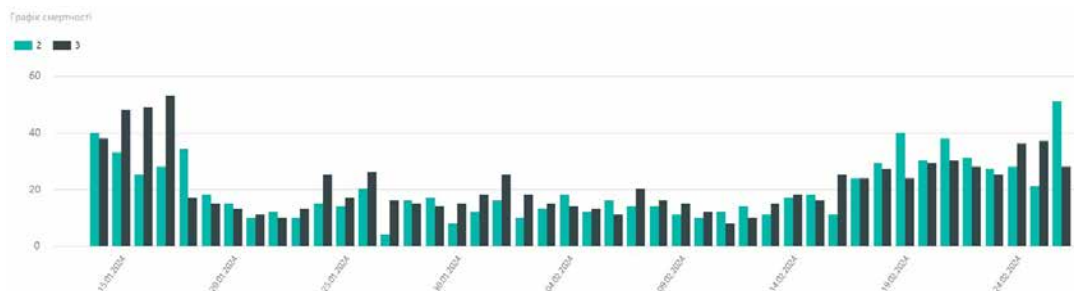


Рис. 33 Графік смертності

Для оцінки результативності функціонування пташників були визначені КРІ, розраховані на основі даних, що містяться у кубі. До таких показників віднесено: відсоток смертності для кожного пташника, відсоток браку продукції та сумарний обсяг отриманої продукції. Розраховані значення КРІ наведено на рис. 34.

KPI_DefectRate_House#2	97.84
KPI_DefectRate_House#3	98.31
KPI_MortalityRate_House#2	2.48
KPI_MortalityRate_House#3	2.72
KPI_ReceivedProduct_House#2	17416
KPI_ReceivedProduct_House#3	18158

Рис. 34 Результати розрахунку КРІ

Отримані результати OLAP-аналізу узгоджуються з висновками, отриманими на етапі Data Mining. Зокрема, згідно зі значеннями КРІ, пташник №3 демонструє вищі втрати (смертність та брак) у порівнянні з пташником №2, що відповідає результатам моделювання, де було показано підвищену чутливість пташника №3 до погіршення мікрокліматичних умов. Натомість, пташник №2 характеризується більш стабільними параметрами та нижчими показниками втрат, що підтверджує його відносну стійкість за різних кліматичних умов.

Таким чином, OLAP-аналіз не лише надав можливість агрегувати та візуалізувати дані для прийняття управлінських рішень, але й підтвердив ефективність виявлених раніше закономірностей за допомогою Data Mining, остаточно виділивши пташник №3 як більш вразливий до відхилень

мікроклімату, що потребує додаткового контролю та корекції режимів утримання.

## ВИСНОВКИ

У ході виконання магістерської роботи було проаналізовано та змодельовано предметну область моніторингу мікроклімату у промисловому пташнику, виконано огляд промислових рішень і сформовано вимоги до аналітичної системи. Спроектовано оперативну БД, сховище даних та налагоджено механізм його наповнення. Завершальним етапом стало проведення аналітики на даних зі сховища із застосуванням технологій OLAP та Data Mining для підтримки прийняття управлінських рішень у птахівництві.

Початковий інтелектуальний аналіз показав межі однофакторних правил: алгоритм 1-Rule не дав надійної класифікації рівня смертності, оскільки за більшістю значень ознак домінував клас «низька смертність». Модель Наївного Байєса дала конкретну відмінність між пташниками: для пташника №2 у всіх типах кліматичних умов прогнозувався клас «низька смертність», тоді як для пташника №3 за «поганих» умов — клас «висока смертність». Аналіз асоціативних правил встановив чіткі залежності між мікрокліматом і втратами: «висока вологість»  $\Rightarrow$  «висока смертність», «низька температура»  $\Rightarrow$  «висока смертність», а комбінація {висока вологість, низька температура}  $\Rightarrow$  «висока смертність». Кластеризація K-means з оптимальною кількістю шість кластерів виокремила групи з підвищеними втратами, які відповідають сценаріям «низька температура / висока вологість», що узгоджується з результатами Байєса та асоціативних правил.

Після цього виконано OLAP-узагальнення на сформованому гіперкубі MS SQL Server із використанням SSRS. Побудовано порівняльний звіт по пташниках за період з агрегованими показниками продукції, споживання води, смертності, браку та параметрів температури/вологості; побудовано графік динаміки смертності у часі; розраховано KPI для кожного пташника. За фактичними значеннями KPI встановлено: у пташника №3 смертність становить 2,72% проти

2,48% у №2; відсоток браку — 98,31% проти 97,84% у №2; сумарний обсяг отриманої продукції — 18 158 у №3 та 17 416 у №2. Графік смертності демонструє для пташника №3 більшу амплітуду коливань і сплески, тоді як для пташника №2 динаміка рівніша. Сукупно це підтвердило зроблені раніше висновки Data Mining: пташник №3 є більш чутливим до погіршення мікрокліматичних умов і потребує посиленого контролю параметрів середовища, насамперед щодо поєднання підвищеної вологості та знижених температур.

Отримані результати дають підстави стверджувати, що поєднання OLAP і Data Mining є доцільним для задач аналітичного моніторингу пташників. Практична цінність полягає у можливості відстежувати динаміку смертності та своєчасно виявляти відхилення мікроклімату.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Тепловий стрес у птахівництві: оптимальні умови для бройлерів. URL: <https://kormil.com.ua/teplovyy-stres-u-ptakhivnytstvi-optymalni-umovy-dlia-broyleriv-povnyu-posibnyk/> (дата звернення: 07.09.2025).
2. Climate in poultry houses. URL: <https://www.poultryhub.org/all-about-poultry/husbandry-management/climate-in-poultry-houses> (дата звернення: 07.09.2025).
3. Параметри, що вирішують усе. URL: <https://agrotimes.ua/article/parametry-shho-vyrishuyut-use/> (дата звернення: 07.09.2025).
4. BlueControl climate controller for poultry production. URL: <https://www.skov.com/en/products/climate-control/bluecontrol-climate-for-poultry/> (дата звернення: 07.09.2025).
5. Poultry Temperature Control & Production | ViperTouch | Big Dutchman. URL: <https://www.bigdutchmanusa.com/poultry-production/products/environmental/controls-and-software/vipertouch/> (дата звернення: 07.09.2025).
6. The most efficient poultry climate control system. URL: <https://www.fancom.com/blog/poultry-climate-control-system> (дата звернення: 07.09.2025).
7. Mingyang Li, Zilin Zhou, Qiang Zhang, Jie Zhang, Yunpeng Suo, Junze Liu, Dan Shen, Lu Luo, Yansen Li, Chunmei Li. Multivariate analysis for data mining to characterize poultry house environment in winter. Poultry Science. 2024. Vol. 103, Issue 5, Article 103633. URL: <https://pubmed.ncbi.nlm.nih.gov/38552343/> (дата звернення: 07.09.2025).
8. Angel Antonio Gonzalez Martinez, Irenilza de Alencar Nääs, Thayla Morandi Ridolfi de Carvalho-Curi, Jair Minoro Abe, Nilsa Duarte da Silva Lima. A Heuristic and Data Mining Model for Predicting Broiler House

- Environment Suitability. *Animals*. 2021. Vol. 11, Issue 10, Article 2780. URL: <https://www.mdpi.com/2076-2615/11/10/2780> (дата звернення: 07.09.2025).
9. Nayden Chivarov, Kristiyan Dimitrov, Stefan Chivarov. Algorithm for Autonomous Management of a Poultry Farm by a Cyber-Physical System. *Animals*. 2023. Vol. 13, Issue 20, Article 3252. URL: <https://www.mdpi.com/2076-2615/13/20/3252> (дата звернення: 07.09.2025).
10. Difference between Function Oriented Design and Object Oriented Design. URL: <https://www.geeksforgeeks.org/blogs/difference-between-function-oriented-design-and-object-oriented-design/> (дата звернення: 08.09.2025).
11. UML Use Case Diagram Tutorial. URL: <https://www.lucidchart.com/pages/tutorial/uml-use-case-diagram> (дата звернення: 08.09.2025).
12. UML Class Diagram Tutorial. URL: <https://www.lucidchart.com/pages/uml-class-diagram> (дата звернення: 08.09.2025).
13. What is a data warehouse?. URL: <https://www.ibm.com/think/topics/data-warehouse> (дата звернення: 09.09.2025).
14. Difference between ROLAP, MOLAP and HOLAP. URL: <https://www.geeksforgeeks.org/dbms/difference-between-rolap-molap-and-holap/> (дата звернення: 09.09.2025).
15. Overview of Online Analytical Processing (OLAP). URL: <https://support.microsoft.com/en-us/office/overview-of-online-analytical-processing-olap-15d2cdde-f70b-4277-b009-ed732b75fdd6> (дата звернення: 09.09.2025).
16. What is data mining?. URL: <https://www.ibm.com/think/topics/data-mining> (дата звернення: 09.09.2025).
17. Learn-One-Rule Algorithm. URL: <https://www.geeksforgeeks.org/machine-learning/learn-one-rule-algorithm/> (дата звернення: 10.09.2025).

18. Naive Bayes Classifiers. URL: <https://www.geeksforgeeks.org/machine-learning/naive-bayes-classifiers/> (дата звернення: 10.09.2025).
19. Association Rule. URL: <https://www.geeksforgeeks.org/machine-learning/association-rule/> (дата звернення: 10.09.2025).
20. Data Mining - Cluster Analysis. URL: <https://www.geeksforgeeks.org/data-analysis/data-mining-cluster-analysis/> (дата звернення: 10.09.2025).
21. Clustering. URL: <https://scikit-learn.org/stable/modules/clustering.html#k-means> (дата звернення: 10.09.2025).

## ДОДАТКИ

### Додаток А

#### Код заповнення таблиці фактів

```

SELECT
    DD.DateID AS DateID,
    DPM.HouseID AS HouseID,
    DPM.BreedID AS BreedID,
    DPM.ActualQuantity AS ActualQuantity,
    DPM.WeightGrams AS WeightGrams,
    DPM.DefectCount AS DefectCount,
    DPM.MortalityCount AS MortalityCount,
    ISNULL(MAX(CASE WHEN S.TypeValue = 'Споживання води' THEN SR.ValueSensor END), 0) AS
WaterConsumption,
    ISNULL(MAX(CASE WHEN S.TypeValue = 'Температура' THEN SR.ValueSensor END), 0) AS
Temperature,
    ISNULL(MAX(CASE WHEN S.TypeValue = 'Вологість' THEN SR.ValueSensor END), 0) AS Humidity
FROM
    DailyProductionMetrics DPM
JOIN
    Poultry_house PH ON DPM.HouseID = PH.HouseID
LEFT JOIN
    Sensor S ON PH.HouseID = S.HouseID
LEFT JOIN
    SensorReading SR ON S.SensorID = SR.SensorID AND DPM.[Date] = SR.[Date]
LEFT JOIN
    dsPrakt.dbo.DateDim DD ON DPM.[Date] = DD.[Date] -- Приєднання до таблиці DateDim з бази
даних dsPrakt
GROUP BY
    DPM.[Date], DD.DateID, PH.Number, DPM.HouseID, DPM.BreedID, DPM.ActualQuantity,
    DPM.WeightGrams, DPM.DefectCount, DPM.MortalityCount
ORDER BY
    DPM.HouseID;

```

## Код виконання 1-Rule

```

private void ClassifyMortalityByPoultryHouse()
{
    if (factTable == null || factTable.Rows.Count == 0 ||
        !factTable.Columns.Contains("MortalityCount"))
    {
        MessageBox.Show("Відсутні дані або стовпець 'MortalityCount' для класифікації  
за пташником та породою.");
        return;
    }

    var overallAvgMortality = factTable.AsEnumerable()
        .Where(r => !r.IsNull("MortalityCount"))
        .Average(r => Convert.ToDecimal(r["MortalityCount"]));

    var result = factTable.AsEnumerable()
        .GroupBy(row => new { HouseID = row.Field<int>("HouseID"), BreedID =
row.Field<int>("BreedID") })
        .Select(g =>
        {
            int houseId = g.Key.HouseID;
            string poultryHouseNum = poultryhouseNumbers.ContainsKey(houseId) ?
poultryhouseNumbers[houseId] : "Невідомо";

            var mortalityValues = g.Where(r => !r.IsNull("MortalityCount"))
                .Select(r =>
                Convert.ToDecimal(r["MortalityCount"]))
                .ToList();

            int lessOverallAvg = mortalityValues.Count(v => v < overallAvgMortality);
            int greaterOverallAvg = mortalityValues.Count(v => v >
overallAvgMortality);

            string classification = (greaterOverallAvg > lessOverallAvg) ? "Висока  
смертність" : "Низька смертність";

            return new
            {
                PoultryHouseNumber = poultryHouseNum,
                MortalityLessOverallAvg = lessOverallAvg,
                MortalityGreaterOverallAvg = greaterOverallAvg,
                OverallAvgMortality = Math.Round(overallAvgMortality, 3),
                Classification = classification
            };
        })
        .OrderBy(r => r.PoultryHouseNumber)
        .ToList();

    dataGridView1.DataSource = result;
}

```

```

private void ClassifyMortalityByConditions()

```

```

{
    if (factTable == null || factTable.Rows.Count == 0 ||
        !factTable.Columns.Contains("MortalityCount") ||
        !factTable.Columns.Contains("Temperature") ||
        !factTable.Columns.Contains("Humidity"))
    {
        MessageBox.Show("Відсутні дані або необхідні стовпці (MortalityCount,
Temperature, Humidity) для класифікації за умовами.");
        return;
    }

    var overallAvgMortality = factTable.AsEnumerable()
        .Where(r => !r.IsNull("MortalityCount"))
        .Average(r => Convert.ToDecimal(r["MortalityCount"]));

    var categorizedData = factTable.AsEnumerable()
        .Select(row =>
        {
            decimal temperature = Convert.ToDecimal(row["Temperature"]);
            decimal humidity = Convert.ToDecimal(row["Humidity"]);

            string conditionClass;
            if ((temperature < 25.0M || temperature > 35.0M) || (humidity < 40.0M ||
humidity > 80.0M))
            {
                conditionClass = "(t < 25.0M || t > 35.0M) || (hum < 40.0M || hum >
80.0M)";
            }
            else if ((temperature >= 28.0M && temperature <= 32.0M) && (humidity >=
50.0M && humidity <= 70.0M))
            {
                conditionClass = "(t >= 28.0M && t <= 32.0M) && (hum >= 50.0M && hum
<= 70.0M)";
            }
            else
            {
                conditionClass = "Other";
            }

            decimal currentMortality = row.IsNull("MortalityCount") ? 0M :
Convert.ToDecimal(row["MortalityCount"]);

            return new
            {
                ConditionClass = conditionClass,
                MortalityCount = currentMortality,
                IsMortalityLessOverallAvg = (currentMortality < overallAvgMortality) ?
1 : 0,
                IsMortalityGreaterOverallAvg = (currentMortality >
overallAvgMortality) ? 1 : 0
            };
        }).ToList();

    var result = categorizedData
        .GroupBy(row => row.ConditionClass)
        .Select(g =>
        {
            int count = g.Count();
            int lessOverallAvg = g.Sum(x => x.IsMortalityLessOverallAvg);
            int greaterOverallAvg = g.Sum(x => x.IsMortalityGreaterOverallAvg);

            string classification = (greaterOverallAvg > lessOverallAvg) ? "Висока
смертність" : "Низька смертність";

            return new

```

```
        {
            ConditionClass = g.Key,
            RecordCount = count,
            MortalityLessOverallAvg = lessOverallAvg,
            MortalityGreaterOverallAvg = greaterOverallAvg,
            OverallAvgMortality = Math.Round(overallAvgMortality, 3),
            Classification = classification
        };
    })
    .OrderByDescending(r => r.RecordCount)
    .ToList();

dataGridView2.DataSource = result;
}
```

## Код виконання методу Наївного Байєса

```

private class NaiveBayesTrainingRow
{
    public int HouseID { get; set; }
    public string ConditionClass { get; set; }
    public string ActualMortalityClass { get; set; }
}

private void ClassifyMortalityByNaiveBayes()
{
    if (factTable == null || factTable.Rows.Count == 0 ||
        !factTable.Columns.Contains("MortalityCount") ||
        !factTable.Columns.Contains("HouseID") ||
        !factTable.Columns.Contains("Temperature") ||
        !factTable.Columns.Contains("Humidity"))
    {
        MessageBox.Show("Відсутні дані або необхідні стовпці (MortalityCount, HouseID,
Temperature, Humidity) для класифікації Наївним Байєсом.");
        return;
    }

    var validMortalityCounts = factTable.AsEnumerable()
        .Where(r => !r.IsNull("MortalityCount"))
        .Select(r => Convert.ToDecimal(r["MortalityCount"]))
        .ToList();

    if (!validMortalityCounts.Any())
    {
        MessageBox.Show("Немає дійсних даних для MortalityCount для обчислення
середнього. Класифікація Наївним Байєсом неможлива.");
        return;
    }

    decimal overallAvgMortality = validMortalityCounts.Average();

    List<NaiveBayesTrainingRow> classifiedData = new List<NaiveBayesTrainingRow>();
    foreach (DataRow row in factTable.Rows)
    {
        if (row.IsNull("MortalityCount") || row.IsNull("Temperature") ||
row.IsNull("Humidity"))
        {
            continue;
        }

        int houseId = Convert.ToInt32(row["HouseID"]);
        decimal mortality = Convert.ToDecimal(row["MortalityCount"]);
        decimal temperature = Convert.ToDecimal(row["Temperature"]);
        decimal humidity = Convert.ToDecimal(row["Humidity"]);

        string mortalityClass = mortality >= overallAvgMortality ? "H" : "L";

        string conditionClass;
        if ((temperature < 25.0M || temperature > 35.0M) || (humidity < 40.0M ||
humidity > 80.0M))
        {
            conditionClass = "Погані умови";
        }
        else if ((temperature >= 28.0M && temperature <= 32.0M) && (humidity >= 50.0M
&& humidity <= 70.0M))
        {
            conditionClass = "Гарні умови";
        }
    }
}

```

```

else
{
    conditionClass = "Нормальні умови";
}

classifiedData.Add(new NaiveBayesTrainingRow
{
    HouseID = houseId,
    ConditionClass = conditionClass,
    ActualMortalityClass = mortalityClass
});
}

if (!classifiedData.Any())
{
    MessageBox.Show("Немає класифікованих даних для навчання Наївного Байєса.");
    return;
}

int totalRecords = classifiedData.Count;
double p_H = (double)classifiedData.Count(x => x.ActualMortalityClass == "H") /
totalRecords;
double p_L = (double)classifiedData.Count(x => x.ActualMortalityClass == "L") /
totalRecords;

List<int> uniqueHouseIDs = classifiedData.Select(x =>
x.HouseID).Distinct().ToList();
List<string> uniqueConditionClasses = classifiedData.Select(x =>
x.ConditionClass).Distinct().ToList();

Dictionary<int, double> p_House_H = new Dictionary<int, double>();
Dictionary<int, double> p_House_L = new Dictionary<int, double>();
foreach (int houseId in uniqueHouseIDs)
{
    int count_H = classifiedData.Count(x => x.ActualMortalityClass == "H");
    int count_L = classifiedData.Count(x => x.ActualMortalityClass == "L");

    p_House_H[houseId] = (double)(classifiedData.Count(x => x.ActualMortalityClass
== "H" && x.HouseID == houseId) + 1) / (count_H + uniqueHouseIDs.Count);
    p_House_L[houseId] = (double)(classifiedData.Count(x => x.ActualMortalityClass
== "L" && x.HouseID == houseId) + 1) / (count_L + uniqueHouseIDs.Count);
}

Dictionary<string, double> p_Condition_H = new Dictionary<string, double>();
Dictionary<string, double> p_Condition_L = new Dictionary<string, double>();
foreach (string conditionClass in uniqueConditionClasses)
{
    int count_H = classifiedData.Count(x => x.ActualMortalityClass == "H");
    int count_L = classifiedData.Count(x => x.ActualMortalityClass == "L");

    p_Condition_H[conditionClass] = (double)(classifiedData.Count(x =>
x.ActualMortalityClass == "H" && x.ConditionClass == conditionClass) + 1) / (count_H +
uniqueConditionClasses.Count);
    p_Condition_L[conditionClass] = (double)(classifiedData.Count(x =>
x.ActualMortalityClass == "L" && x.ConditionClass == conditionClass) + 1) / (count_L +
uniqueConditionClasses.Count);
}

DataTable resultTable = new DataTable();
resultTable.Columns.Add("PoultryHouse", typeof(string));
resultTable.Columns.Add("ConditionClass", typeof(string));
resultTable.Columns.Add("P(H|Features)", typeof(double));
resultTable.Columns.Add("P(L|Features)", typeof(double));
resultTable.Columns.Add("Predicted_Class", typeof(string));

```

```

foreach (int houseId in uniqueHouseIDs)
{
    foreach (string conditionClass in uniqueConditionClasses)
    {
        double numeratorH = p_House_H.ContainsKey(houseId) ? p_House_H[houseId] :
0;
        numeratorH *= p_Condition_H.ContainsKey(conditionClass) ?
p_Condition_H[conditionClass] : 0;
        numeratorH *= p_H;

        double numeratorL = p_House_L.ContainsKey(houseId) ? p_House_L[houseId] :
0;
        numeratorL *= p_Condition_L.ContainsKey(conditionClass) ?
p_Condition_L[conditionClass] : 0;
        numeratorL *= p_L;

        double totalProbability = numeratorH + numeratorL;
        double final_p_H = totalProbability > 0 ? numeratorH / totalProbability :
0;
        double final_p_L = totalProbability > 0 ? numeratorL / totalProbability :
0;

        string predictedClass = final_p_H >= final_p_L ? "H" : "L";

        resultTable.Rows.Add(
            poultryhouseNumbers.ContainsKey(houseId) ?
poultryhouseNumbers[houseId] : "Невідомо",
            conditionClass,
            Math.Round(final_p_H, 5),
            Math.Round(final_p_L, 5),
            predictedClass
        );
    }
}

dataGridView3.DataSource = resultTable;
}

```

## Код виконання методу асоціативних правил

```

import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
from sqlalchemy import create_engine
import warnings

warnings.filterwarnings('ignore')

connection_string = (
    "mssql+pyodbc://localhost/dsPrakt?driver=ODBC+Driver+17+for+SQL+Server"
)
engine = create_engine(connection_string)

query = "SELECT * FROM ProductFact"
try:
    df = pd.read_sql(query, engine)
    print("Дані успішно завантажено з ProductFact.")
except Exception as e:
    print(f"Помилка завантаження даних: {e}")
    df = pd.DataFrame(columns=['DateID', 'HouseID', 'BreedID', 'Temperature',
                              'Humidity', 'WaterConsumption',
                              'ActualQuantity', 'WeightGrams', 'DefectCount',
                              'MortalityCount'])

if df.empty:
    print("DataFrame порожній. Перевірте підключення до бази даних та наявність даних у ProductFact.")
else:
    temp_median = df['Temperature'].median()
    df['high_temp'] = df['Temperature'] > temp_median
    df['low_temp'] = df['Temperature'] < temp_median

    hum_median = df['Humidity'].median()
    df['high_hum'] = df['Humidity'] > hum_median
    df['low_hum'] = df['Humidity'] < hum_median

    mortality_median = df['MortalityCount'].median()
    df['high_mortality_count'] = df['MortalityCount'] > mortality_median
    df['low_mortality_count'] = df['MortalityCount'] < mortality_median

    all_bool_cols = [col for col in df.columns if col.startswith(('high_', 'low_'))]
    df_bool = df[all_bool_cols].astype(bool)

    frequent_itemsets = apriori(df_bool, min_support= 0.05, use_colnames= True)

    if frequent_itemsets.empty:
        print("Не знайдено частих наборів елементів із заданим min_support.")
    else:
        rules = association_rules(frequent_itemsets, metric= "confidence",
min_threshold= 0.5)

        rules_sorted = rules.sort_values(by='support', ascending=False)

        microclimate_params = [
            'high_temp', 'low_temp', 'high_hum', 'low_hum'
        ]
        production_params = [
            'high_mortality_count', 'low_mortality_count'
        ]

        rules_filtered = rules_sorted[

```

```

        rules_sorted['antecedents'].apply(lambda x: all(param in
microclimate_params for param in x)) &
        rules_sorted['consequents'].apply(lambda x: all(param in production_params
for param in x))
    ]

    if not rules_filtered.empty:
        print("\n=== Асоціативні правила, які були знайдені ===")
        for index, row in rules_filtered.iterrows():
            print(f"Правило: {set(row['antecedents'])} =>
{set(row['consequents'])}")
            print(f"Підтримка: {row['support']:.4f}, Впевненість:
{row['confidence']:.4f}, Lift: {row['lift']:.4f}")
            print(f"----")
    else:
        print("\nНемає знайдених асоціативних правил із заданими параметрами
мікроклімату та продукції.")

```

## Код виконання кластеризації

```

import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.decomposition import PCA
import pyodbc
import seaborn as sns
import matplotlib.pyplot as plt

connection_string = (
    "Driver={ODBC Driver 17 for SQL Server};"
    "Server=localhost;Database=dsPrakt;Trusted_Connection=yes;"
)

conn = pyodbc.connect(connection_string)
query = """
    SELECT *
    FROM ProductFact
    """
df = pd.read_sql(query, conn)
conn.close()

features = df.drop(columns=['BreedID', 'HouseID', 'DateID'])
scaler = StandardScaler()
X_scaled = scaler.fit_transform(features)

range_n_clusters = range(2, 11)
silhouette_scores = []
best_n = 2
best_score = -1

print("Силуєтні коефіцієнти для різної кількості кластерів:")
for n_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=n_clusters, random_state=42, n_init='auto')
    labels = kmeans.fit_predict(X_scaled)
    score = silhouette_score(X_scaled, labels)
    silhouette_scores.append(score)
    print(f"Кількість кластерів: {n_clusters}, силуєтний коефіцієнт: {score:.4f}")
    if score > best_score:
        best_score = score
        best_n = n_clusters

plt.figure(figsize=(8, 5))
plt.plot(range_n_clusters, silhouette_scores, marker='o')
plt.xlabel("Кількість кластерів")
plt.ylabel("Силуєтний коефіцієнт")
plt.title("Силуєтний коефіцієнт залежно від кількості кластерів")
plt.grid(True)
plt.show()

final_kmeans = KMeans(n_clusters=best_n, random_state=42, n_init='auto')
df['cluster'] = final_kmeans.fit_predict(X_scaled)

print(f"\nОптимальна кількість кластерів: {best_n}")
print("Розподіл записів по кластерам:")
print(df['cluster'].value_counts().sort_index())

result = df[['BreedID', 'HouseID', 'cluster']]
print("\nРозподіл по кластерам:")
print(result.value_counts().reset_index(name='count'))

```

```
centroids = pd.DataFrame(final_kmeans.cluster_centers_, columns=features.columns)
print("\nЦентроїди кластерів (повністю):")
print(centroids)

pca = PCA(n_components= 2)
pca_data = pca.fit_transform(X_scaled)

plt.figure(figsize= (8, 6))
sns.scatterplot(x=pca_data[:, 0], y=pca_data[:, 1], hue=df['cluster'], palette='Set1',
s=50)
plt.title(f"Візуалізація кластерів (PCA) (k={best_n})")
plt.xlabel("PCA1")
plt.ylabel("PCA2")
plt.grid(True)
plt.legend(title= "Кластер")
plt.show()
```

## Код запиту створення звіту

```

WITH InitialActualQuantityForPeriod AS (
    SELECT
        PF.HouseID,
        PF.BreedID,
        PF.ActualQuantity AS InitialActualQuantity,
        ROW_NUMBER() OVER (
            PARTITION BY PF.HouseID, PF.BreedID
            ORDER BY DD.[Date] ASC
        ) AS rn
    FROM
        ProductFact PF
    JOIN
        DateDim DD ON PF.DateID = DD.DateID
    WHERE
        DD.[Date] >= @StartDate AND DD.[Date] <= @EndDate
)
SELECT
    PH.Number AS [Номер Пташника],
    B.[Name] AS [Назва Породи],
    MAX(PF.ActualQuantity) AS [К-сть голів на початок періоду],
    SUM(PF.WeightGrams) AS [Сумарна Вага (г.)],
    SUM(PF.WaterConsumption) AS [Сумарне Споживання Води (л.)],
    CAST(SUM(PF.MortalityCount) AS DECIMAL(18, 5)) * 100.0 /
    NULLIF(IQP.InitialActualQuantity, 0) AS [Рівень Смертності за Період (%)],
    CAST(SUM(PF.DefectCount) AS DECIMAL(18, 5)) * 100.0 / NULLIF(IQP.InitialActualQuantity,
    0) AS [Рівень Браку за Період (%)],

    AVG(PF.Temperature) AS [Середня Температура (°C)],
    MIN(PF.Temperature) AS [Мін. Температура (°C)],
    MAX(PF.Temperature) AS [Макс. Температура (°C)],
    AVG(PF.Humidity) AS [Середня Вологість (%)],
    MIN(PF.Humidity) AS [Мін. Вологість (%)],
    MAX(PF.Humidity) AS [Макс. Вологість (%)]
FROM
    ProductFact PF
JOIN
    PoultryHouseDim PH ON PF.HouseID = PH.HouseID
JOIN
    BreedDim B ON PF.BreedID = B.BreedID
JOIN
    DateDim DD ON PF.DateID = DD.DateID
LEFT JOIN
    InitialActualQuantityForPeriod IQP ON PF.HouseID = IQP.HouseID
    AND PF.BreedID = IQP.BreedID
WHERE
    DD.[Date] >= @StartDate AND DD.[Date] <= @EndDate
    AND IQP.rn = 1
GROUP BY
    PH.Number,
    B.[Name],
    IQP.InitialActualQuantity
ORDER BY
    PH.Number,
    B.[Name];

```

## Створений звіт

## Порівняльний звіт по пташникам за 13.01.2024 - 26.02.2024

Номер Назва Пташника Породи	К сть голів на початок періоду	Сумарна Вага г	Сумарне Водил	Рівень Смертності за		Рівень Браку за		Температура			Середня Вологість		
				Період (%)	Період (%)	Температура С	Температура С	Температура С	Макс Температура С	Середня Вологість (%)	Мін Вологість (%)	Макс Вологість (%)	
2 СОВВ - 500	35000,00	17416,00	283070,00	2,48	97,84	24,96	17,30	33,50	57,98	45,00	66,00		
3 СОВВ - 500	35000,00	18158,00	280374,00	2,172	98,31	25,32	19,20	33,50	57,76	44,00	69,00		