

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних.

СД – сховище даних.

СППР – система підтримки прийняття рішень.

СУБД – система управління базами даних.

SQL – SQL Server Analysis Services (SSAS).

SSAS – SQL Server Integration Services.

SSIS – SQL Server Integration Services.

SSRS – SQL Server Reporting Services.

OLAP – On-Line Analytical Processing

DSV - Data Source View.

DF – Data Flow.

Зміст

ВСТУП	4
1 АНАЛІЗ ПОСТАНОВКИ ЗАВДАННЯ ТА ПРЕДМЕТНОЇ ОБЛАСТІ.....	5
1.1 Постановка завдання	5
1.2 Діаграма прецедентів.....	6
1.3 Архітектура системи СППР керівництвом тепличного господарства промислового типу	8
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....	10
2.1 Структура БД тепличного господарства	10
2.2 Загальні поняття з напрямку OLAP-технології	12
2.3 Структура сховища даних.....	14
3 РОЗРОБКА СИСТЕМИ АНАЛІЗУ	17
3.1 Механізм вилучення, обробки і передачі даних.....	17
3.1.1 Побудова розгорнутого куба з допомогою служби SSAS	17
3.1.2 Реалізація отримання даних за допомогою Data Flow	22
3.2 Побудова звітності в середовищі ВІ	26
3.3 Розрахунок КРІ	30
ВИСНОВКИ.....	34
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	35
ДОДАТКИ.....	37
ДОДАТОК А.....	38

ВСТУП

Актуальність теми. Якісне й ефективне вирощування продуктів дуже важливе, адже це безпосередньо впливає на наше здоров'я, а для виробників допомагає нормалізувати витрати на сам процес. Цей процес залежить від детального вивчення та аналізу всіх умов, необхідних для забезпечення нормального росту рослин. Під час вирощування рослин дуже важливо враховувати оптимальні значення мікроклімату, щоб отримати більший та якісний врожай. Саме тому ефективним є використання програмних засобів для збереження, моніторингу та аналізу всіх важливих показників. Такі системи можуть постійно контролювати стан теплиці, надавати інформацію у режимі реального часу, проводити аналіз на основі внесених даних.

На основі всіх існуючих даних виробник може аналізувати всі ключові показники, їхні зміни та вплив та приймати відповідні рішення. Проте створені системи з часом розширюються, а відповідно інформація в них теж, то стає незручно та неефективно аналізувати внесені дані. Тому у такому випадку існує необхідність у створенні системи, яка буде проводити аналіз роботи на основі накопичених даних.

Об'єкт дослідження: вирощування овочів у теплиці.

Предмет дослідження: система прийняття рішень, яка допоможе керівництву теплиці підвищити ефективність у вирощуванні продукції шляхом аналізу показників, які впливають на цей процес.

Мета дослідження: підвищити ефективність вирощування овочів у спорудах закритого ґрунту.

Структура роботи. Курсовий проєкт складається зі вступу, трьох розділів (аналізу предметної області, інформаційного забезпечення, розробки системи аналізу), висновків, списку використаних джерел та додатків.

Записка складається з 39-ти сторінок та 1-го додатку. У роботі використано 11 джерел.

1 АНАЛІЗ ПОСТАНОВКИ ЗАВДАННЯ ТА ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Постановка завдання

Для аналізу тепличного господарства необхідно забезпечити зберігання таких даних:

- інформація про теплиці, а саме:
 - назва теплиці;
 - площа;
- інформація про сорт:
 - назва сорту;
 - культура, до якої даний сорт відноситься;
- оптимальні умови:
 - діапазон оптимальної температури;
 - діапазон оптимальної вологості;
- інформація про врожайність:
 - врожай.
- інформація про зафіксовану температуру;
- інформація про зафіксовану вологість.

Для ефективної роботи системи необхідно забезпечити внесення в оперативну БД вище зазначеної інформації, а також розробити структуру СД для подальшої передачі даних у неї.

Перелік питань, на які необхідно дати відповідь під час аналізу:

- 1) Який сорт має найбільшу врожайність у розрізі вказаного часового періоду?
- 2) Яка врожайність сорту за вказаний часовий період у розрізі температури та вологості?

1.2 Діаграма прецедентів

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

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

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

У діаграмі прецедентів можуть бути такі варіанти взаємодії:

- використання;
- включення;
- розширення;
- вимога;
- схожість;
- рівнозначність.

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

Між прецедентами можуть бути такі зв'язки:

- Включення, або *include* – це взаємозв'язок, коли залежний прецедент обов'язково виконується на основі базового.
- Розширення, або *extend* – це різновид взаємозв'язку, коли базовий прецедент не обов'язково використовує дочірній [1, 2].

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

Актори, які діють у системі

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

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. 1.

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

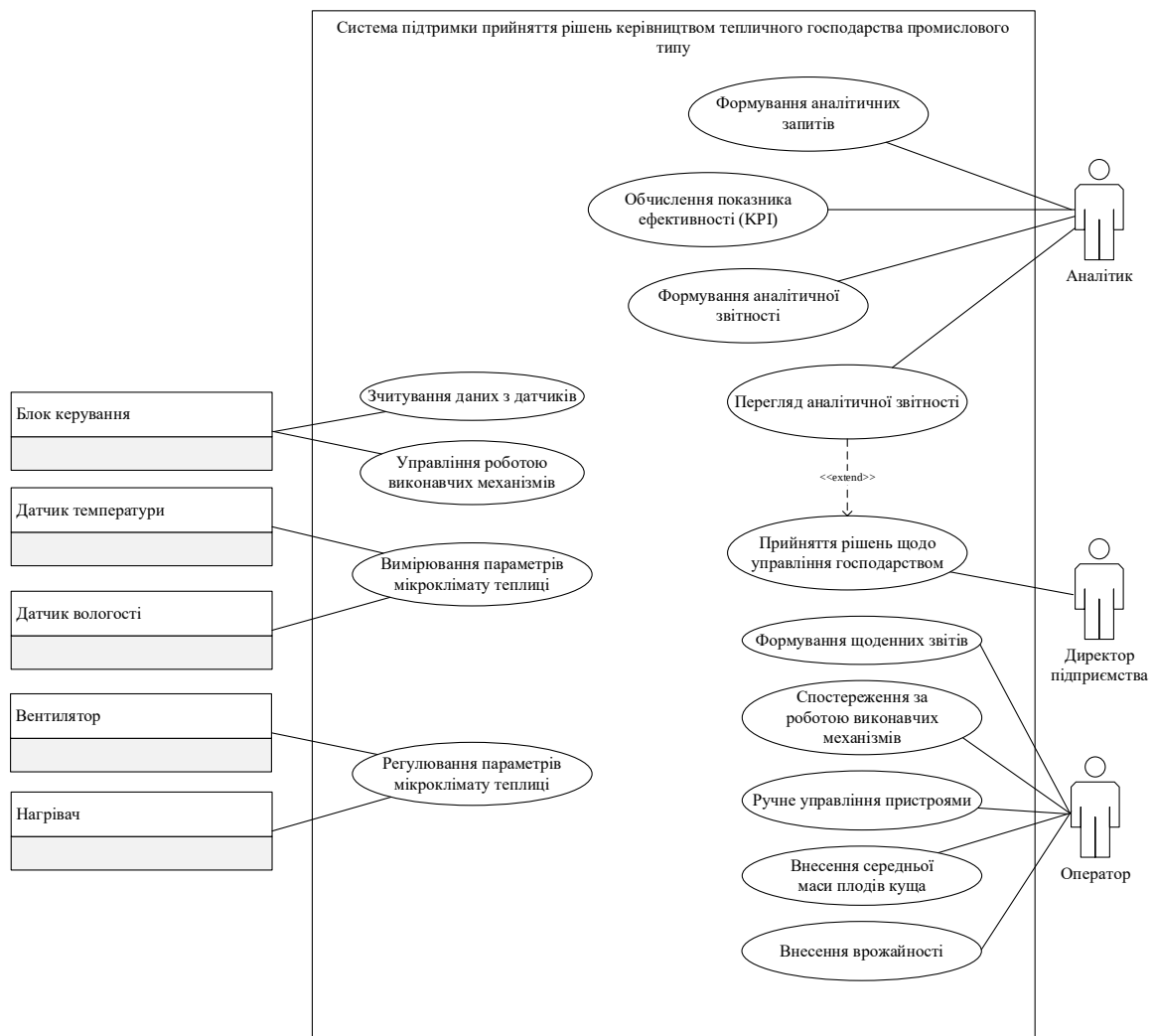


Рис. 1 Діаграма прецедентів до розроблюваної СППР

1.3 Архітектура системи СППР керівництвом тепличного господарства промислового типу

На рис. 2 зображено архітектуру розроблюваної системи. Основними фізичними вузлами є:

- робоча станція теплиці:
 - модуль внесення даних в оперативну БД;
 - оперативна БД теплиці;
 - блок керування;
 - датчики температури та вологості;
- сервер сховища даних;
- робоча станція аналітика.

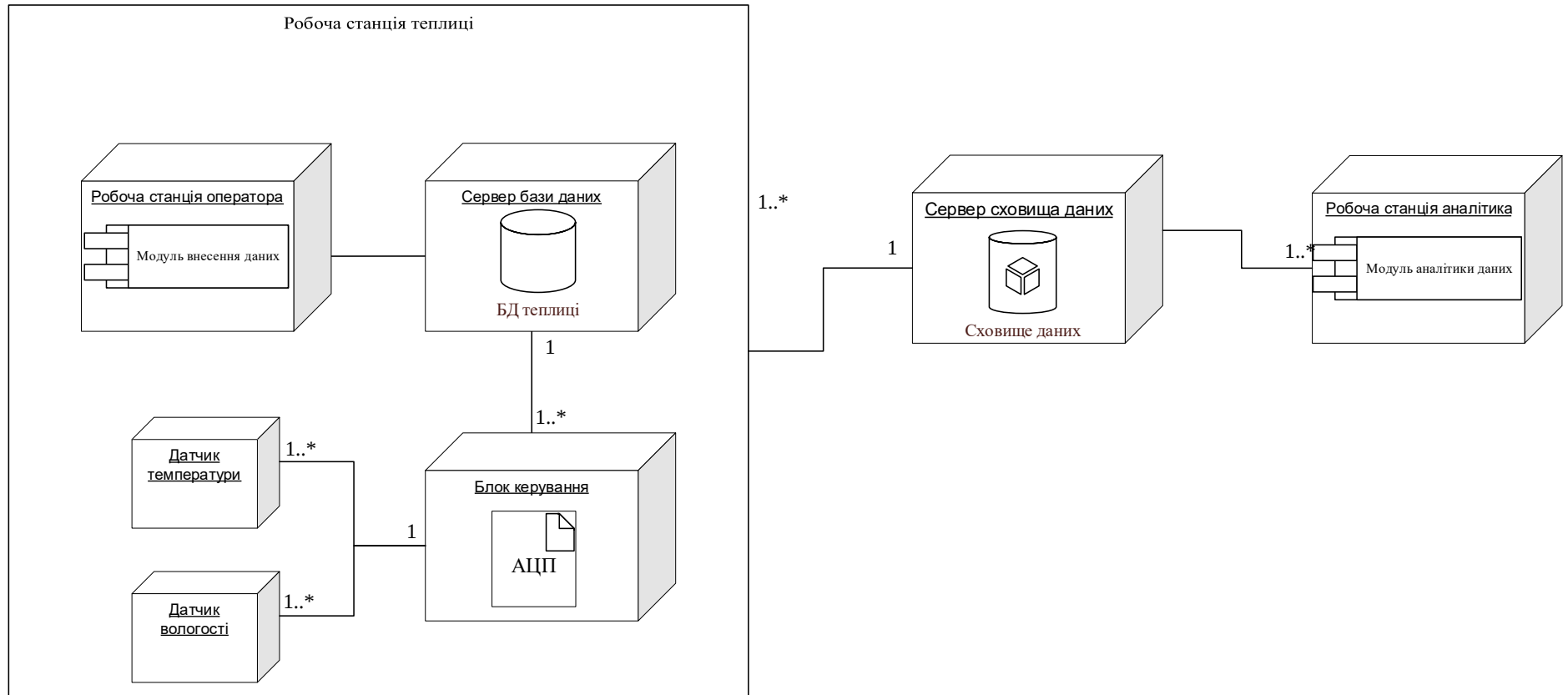


Рис. 2 Архітектура системи

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ Структура БД тепличного господарства

СППР розроблюється на основі розроблюваної підсистеми моніторингу автоматизованої системи управління мікрокліматом в теплиці. На рис. 3 представлена логічна схема оперативної бази даних.

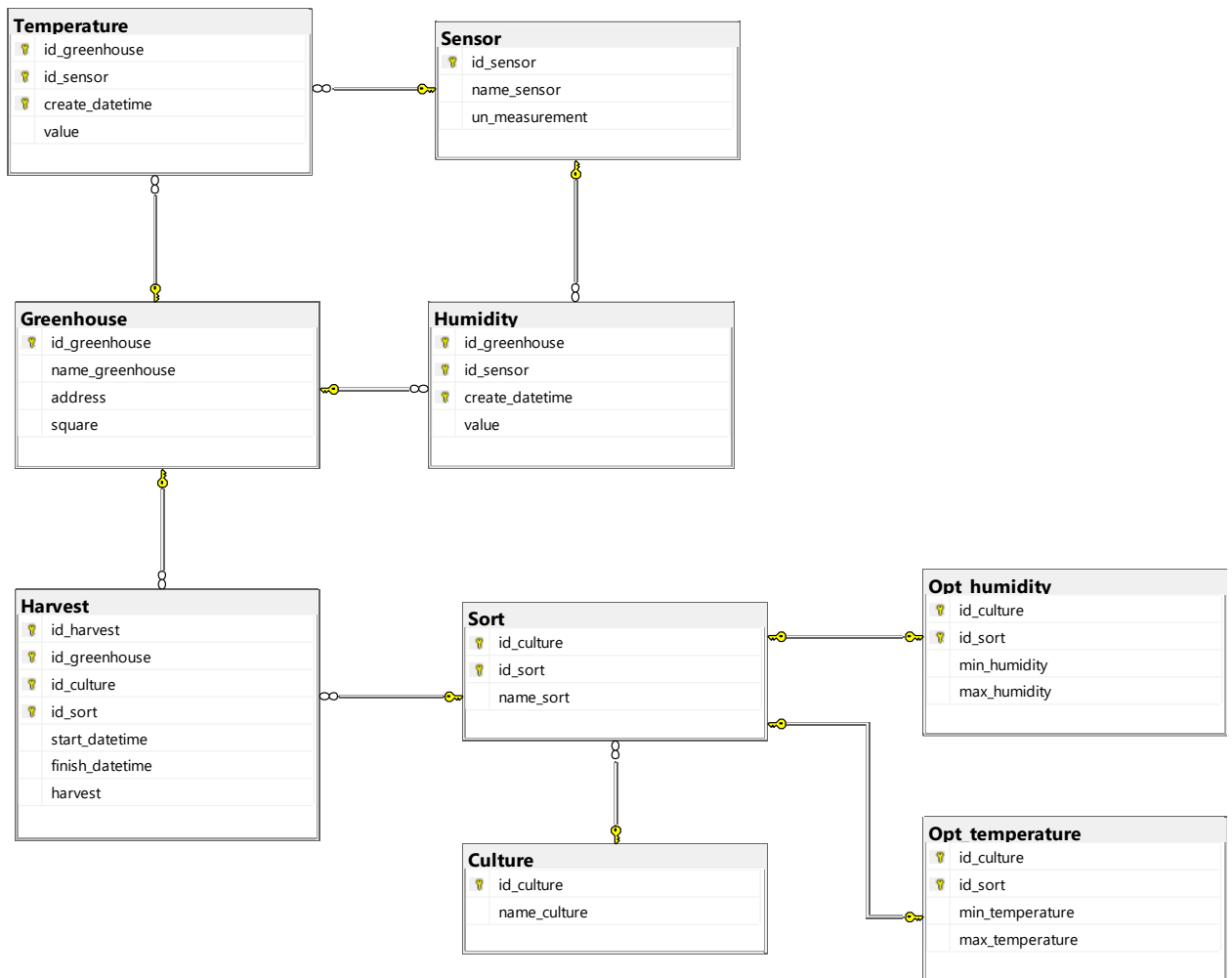


Рис. 3 Логічна модель БД тепличного господарства

На етапі розробки логічної моделі даних для системи моніторингу було виділено 9 сутностей:

- «Sensor» – інформація про датчик: містить ключовий атрибут код датчика (id_sensor) та два неключові, а саме назву датчика (name) та одиницю вимірювання (un_measurement);

- «Greenhouse» – інформація про теплицю: містить ключовий атрибут код теплиці (id_greenhouse) та три неключові – назву (name), адресу (address) та площу (square).
- «Temperature» – інформація про вимірювання температури у теплиці: містить час (datetime), код датчика (id_sensor), код теплиці (id_greenhouse) як ключові атрибути та значення (value) як неключовий;
- «Humidity» – інформація про вимірювання вологості у теплиці: містить час (datetime), код датчика (id_sensor), код теплиці (id_greenhouse) як ключові атрибути та значення (value) як неключовий;
- «Harvest» – інформація про врожайність у теплиці: містить код врожаю (id_harvest), код сорту (id_sort), код культури (id_culture) та код теплиці (id_greenhouse) як ключові атрибути та врожайність (harvest), день посадки (start_datetime) та день збирання врожаю (finish_datetime) як неключові атрибути;
- «Culture» – інформація про культуру: містить один ключовий атрибут – код культури (id_culture) та один неключовий – назву (name);
- «Sort» – інформація про сорт певної культури: містить один ключовий атрибут – код сорту (id_sort) та два неключові – назву сорту (name) та код культури (id_culture), до якої відноситься вказаний сорт;
- «Opt_temperature» – діапазон оптимальної температури для сорту: містить один ключовий атрибут – код сорту (id_sort), код культури (id_culture) та два неключові – нижній (min_temperature) та верхній (max_temperature) поріг оптимального показника температури;
- «Opt_humidity» – діапазон оптимальної вологості для сорту: містить один ключовий атрибут – код сорту (id_sort), код

культури (id_culture) та два неключові – нижній (min_humidity) та верхній (max_humidity) поріг оптимального показника вологості.

БД створена у СУБД MS SQL Server за допомогою SQL запитів. Приклад створення таблиці наведений на рис. 4.

```
CREATE TABLE Culture
(
    id_culture          integer IDENTITY ( 1, 1 ) ,
    name_culture        nvarchar(50) NOT NULL
)
go

ALTER TABLE Culture
    ADD CONSTRAINT XPKCulture PRIMARY KEY CLUSTERED (id_culture ASC)
go
```

Рис. 4 Створення таблиці «Culture»

2.2 Загальні поняття з напрямку OLAP-технології

Бази даних онлайн-аналітичної обробки (OLAP) полегшують запити бізнес-аналітики. OLAP — це технологія бази даних, оптимізована для запитів і звітів, а не для обробки транзакцій. Вихідними даними для OLAP є бази даних з онлайновою обробкою транзакцій (OLTP), які зазвичай зберігаються в сховищах даних. Дані OLAP отримуються з цих історичних даних і об'єднуються в структури, які дозволяють проводити складний аналіз. Дані OLAP також організовані ієрархічно і зберігаються в кубах замість таблиць. Це складна технологія, яка використовує багатовимірні структури для забезпечення швидкого доступу до даних для аналізу. Завдяки цій організації у звіті зведеної таблиці або звіту зі зведеною діаграмою легко відображатися підсумки високого рівня, наприклад загальні обсяги продажів у всій країні чи регіоні, а також відображатися деталі для сайтів, де продажі є особливо високими або слабкими.

Бази даних OLAP призначені для прискорення пошуку даних. Оскільки сервер OLAP, а не Microsoft Office Excel, обчислює підсумовані

значення, під час створення або зміни звіту в Excel потрібно надсилати менше даних. Цей підхід дозволяє працювати з набагато більшими обсягами вихідних даних, ніж ви могли б, якби дані були організовані в традиційній базі даних, де Excel витягує всі окремі записи, а потім обчислює підсумовані значення.

Бази даних OLAP містять два основних типи даних: міри, які є числовими даними, величини та середні значення, які використовуються для прийняття обґрунтованих бізнес-рішень, і виміри, які є категоріями, які ви використовуєте для організації цих показників. Бази даних OLAP допомагають організувати дані за багатьма рівнями деталізації, використовуючи для аналізу даних необхідні критерії [3].

В основному існує три типи OLAP:

- MOLAP (багатовимірний OLAP) – це OLAP для багатовимірних індексів баз даних.
- ROLAP (Relational OLAP) – це OLAP, який здійснює динамічний багатовимірний аналіз на збережених даних реляційних баз даних.
- HOLAP (Hybrid OLAP) – це різноманітна інтеграція ROLAP та MOLAP. Він використовується для розробки ємності даних ROLAP, а MOLAP – найкраща можливість обробки для виконання вимог обробки [4].

Для розробки курсового проєкту було використано архітектуру MOLAP. Багатовимірний OLAP (MOLAP) — це класичний OLAP, який полегшує аналіз даних за допомогою багатовимірного куба даних. Дані попередньо обчислюються, повторно підсумовуються та зберігаються в MOLAP (основна відмінність від ROLAP). Використовуючи MOLAP, користувач може використовувати дані багатовимірного перегляду з різними аспектами.

Багатовимірний аналіз даних також можливий, якщо використовується реляційна база даних. Для цього знадобиться запитувати

дані з кількох таблиць. Навпаки, MOLAP має всі можливі комбінації даних, які вже зберігаються в багатовимірному масиві. MOLAP може отримати доступ до цих даних безпосередньо. Таким чином, MOLAP швидше в порівнянні з Relational Online Analytical Processing (ROLAP) [5].

2.3 Структура сховища даних

Сховище даних (data warehouse) — це тип системи керування даними, яка призначена для забезпечення та підтримки діяльності бізнес-аналітики (BI). Сховища даних призначені виключно для виконання запитів та аналізу і часто містять великі обсяги історичних даних. Дані в сховищі даних зазвичай отримують з широкого діапазону джерел, таких як файли журналів програм і програми транзакцій.

Сховище даних централізує та консолідує великі обсяги даних із кількох джерел. Його аналітичні можливості дозволяють організаціям отримувати цінну бізнес-ідею зі своїх даних для покращення процесу прийняття рішень [6].

Під час розробки системи було створено сховище даних, яке дозволить проводити аналіз у різних розрізах. Структура сховища даних зображена на рис. 5. Скрипти запитів створення всіх таблиць СД наведено у додатку А.

Для збереження необхідних даних у СД були розроблені такі таблиці:

- SortDim – вимір, що містить інформацію про сорт з такими полями:
 - id_culture – ідентифікатор культури;
 - id_sort – ідентифікатор сорту;
 - name_culture – назва культури;
 - name_sort – назва сорту;
- OptConditionDim – вимір, що зберігає оптимальні показники для сорту
 - id_condition – ідентифікатор умови;
 - id_culture – ідентифікатор культури;

- id_sort – ідентифікатор сорту;
- min_temperature – мінімальна оптимальна температура для сорту;
- max_temperature – максимальна оптимальна температура для сорту;
- min_humidity – мінімальна оптимальна вологість для сорту;
- max_humidity – максимальна оптимальна вологість для сорту;
- GreenhouseDim – вимір, що містить інформацію про теплицю:
 - id_greenhouse – ідентифікатор теплиці;
 - name_greenhouse – назва теплиці;
 - address – адреса, де знаходиться теплиця;
 - square – площа теплиці;
- DateDim – вимір, що зберігає часовий проміжок:
 - id_date – ідентифікатор дати;
 - year – рік;
 - month – місяць;
 - day – день.
- HarvestFact – таблиця фактів про зафіксовану врожайність з такими полями:
 - id_date – ідентифікатор дати;
 - id_greenhouse – ідентифікатор теплиці;
 - id_culture – ідентифікатор культури;
 - id_sort – ідентифікатор сорту;
 - avg_temperature – середня зафіксована температура за періоду вирощування врожаю;
 - avg_humidity – середня зафіксована вологість за періоду вирощування врожаю;
 - min_temperature – мінімальна зафіксована температура за періоду вирощування врожаю;

- max_temperature – максимальна зафіксована температура за періоду вирощування врожаю;
- min_humidity – мінімальна зафіксована вологість за періоду вирощування врожаю;
- max_humidity –максимальна зафіксована вологість за періоду вирощування врожаю;
- harvest – кількість врожаю за період вирощування.



Рис. 5 СД тепличного господарства

Так як інформація у таблицях-вимірах є відносно постійною, тому дані для цих таблиць заповнювались за допомогою SQL запитів, один з яких наведений на рис. 6.

```
BEGIN TRANSACTION
INSERT INTO GreenhouseDim ([name_greenhouse], [address], [square]) VALUES
('Цех №1', 'Київська обл., Фастівський район, м. Боярка', 5.0),
('Цех №2', 'Київська обл., Фастівський район, м. Боярка', 3.5),
('Цех №3', 'Київська обл., Фастівський район, м. Боярка', 4.0);
GO
COMMIT
GO
```

Рис. 6 Внесення умовно-постійної інформації про теплиці

Results		Messages		
	id_greenhouse	name_greenhouse	address	square
1	1	Цех №1	Київська обл., Фастівський район, м. Боярка	5.00000
2	2	Цех №2	Київська обл., Фастівський район, м. Боярка	3.50000
3	3	Цех №3	Київська обл., Фастівський район, м. Боярка	4.00000

Рис. 7 Результат виконання внесення інформації

3 РОЗРОБКА СИСТЕМИ АНАЛІЗУ

3.1 Механізм вилучення, обробки і передачі даних

3.1.1 Побудова розгорнутого куба з допомогою служби SSAS

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

Для розробки кубу було використано середовище Visual Studio з розширенням SSAS. На першому етапі необхідно визначити джерело даних – база даних OLAP або сховище даних. На основі визначеного джерела даних будуть імпортуватись необхідні дані. На рис. 7 зображено формування підключення джерела даних за допомогою модуля Data Source Wizard, де ми обираємо попередньо створене сховище даних.

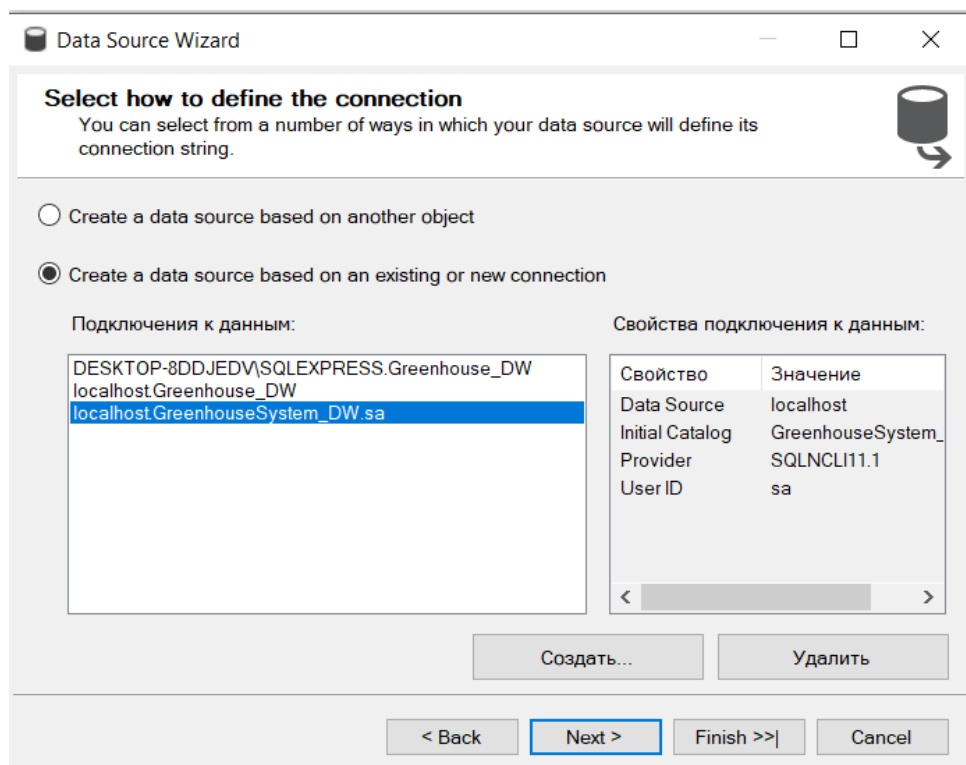


Рис. 7 Додавання джерела даних

Наступним етапом у розробці куба є створення уявлення джерела даних, яке являє собою абстракцію реляційного джерела даних, яка стає основою кубів і вимірів, які ви створюєте в багатовимірному проєкті. Метою DSV є надання контролю над структурами даних, які використовуються у проєкті, і працювати незалежно від базових джерел даних (наприклад, можливість перейменувати або об'єднати стовпці без безпосередньої зміни вихідного джерела даних). У процесі розробки можна створити кілька представлень джерел даних у проєкті або базі даних Analysis Services на одному або кількох джерелах даних і створити кожне з них, щоб задовольнити вимоги до іншого рішення [8].

На рис. 8-9 зображено додавання уявлення за допомогою Data Source View Wizard. У розроблюваному проєкті уявленням виступає створене СД.

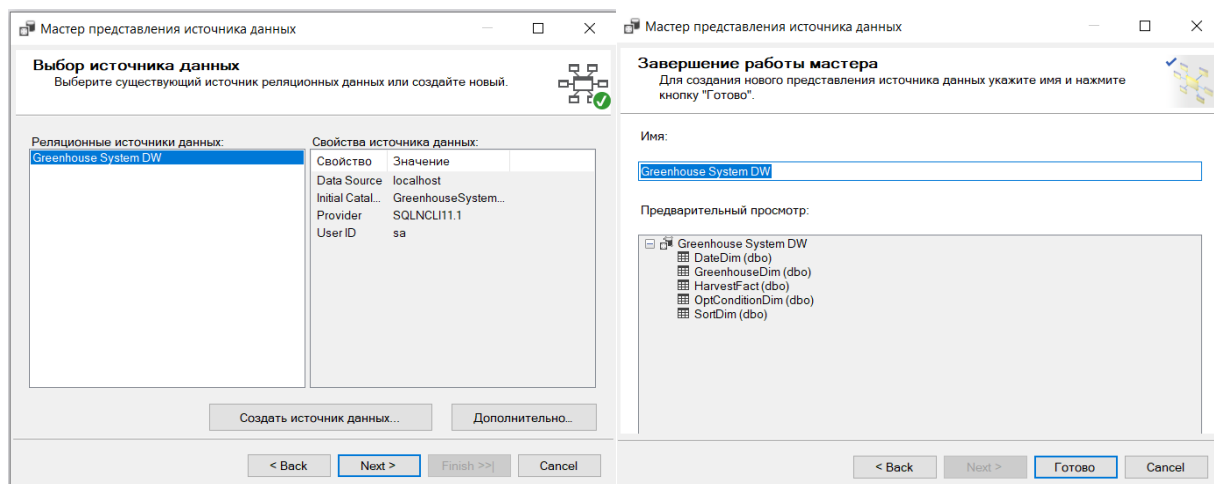


Рис. 8-9 Створення уявлення на основі СД GreenhouseSystem_DW

На основі створеного джерела та уявлення джерелах даних необхідно створити виміри для OLAP куба. Дивлячись на структуру створеного сховища у пункті 2.3 куб матиме чотири виміри.

На рис. 10 зображено процес створення часового виміру, яке буде дозволяти проводити аналіз у розрізі часу: року, місяця, дня.

The screenshot shows a window titled "Мастер измерений" (Master of Measurements). The main heading is "Завершение работы мастера" (Master work completion). Below it, the text says: "Укажите имя нового измерения, проверьте его структуру, а затем нажмите кнопку 'Готово', чтобы сохранить его." (Specify the name of the new measurement, check its structure, and then click the 'Ready' button to save it). There is a 3D cube icon with a green checkmark in the top right corner.

The "Имя:" (Name) field contains "Date Dim 1".

The "Предварительный просмотр:" (Preview) section shows a tree structure:

- [-] Date Dim 1
 - [-] Атрибуты (Attributes)
 - Id Date
 - Year
 - Month
 - Day

At the bottom, there are four buttons: "< Back", "Next >", "Готово" (Ready), and "Cancel". The "Готово" button is highlighted with a blue border.

Рис. 10 Створення часового виміру (DateDim)

Наступним виміром є вимір теплиці, створення якого зображено на рис. 11. Надалі даний вимір буде використовуватись для аналізу врожайності у розрізі конкретної теплиці. Аналітик може переглянути наскільки ефективним є вирощування конкретного сорту у певній теплиці.

The screenshot shows a window titled "Мастер измерений" (Master of Measurements). The main heading is "Завершение работы мастера" (Master work completion). Below it, the text says: "Укажите имя нового измерения, проверьте его структуру, а затем нажмите кнопку 'Готово', чтобы сохранить его." (Specify the name of the new measurement, check its structure, and then click the 'Ready' button to save it). There is a 3D cube icon with a green checkmark in the top right corner.

The "Имя:" (Name) field contains "Greenhouse Dim 1".

The "Предварительный просмотр:" (Preview) section shows a tree structure:

- [-] Greenhouse Dim 1
 - [-] Атрибуты (Attributes)
 - Id Greenhouse
 - Name Greenhouse
 - Square

At the bottom, there are four buttons: "< Back", "Next >", "Готово" (Ready), and "Cancel". The "Готово" button is highlighted with a blue border.

Рис. 11 Створення виміру GreenhouseDim

Наступним виміром є вимір сорту, який є одним з ключових вимірів. У подальшому можна проводити аналіз у розрізі сорту та культури, до якої відносяться певні сорти.

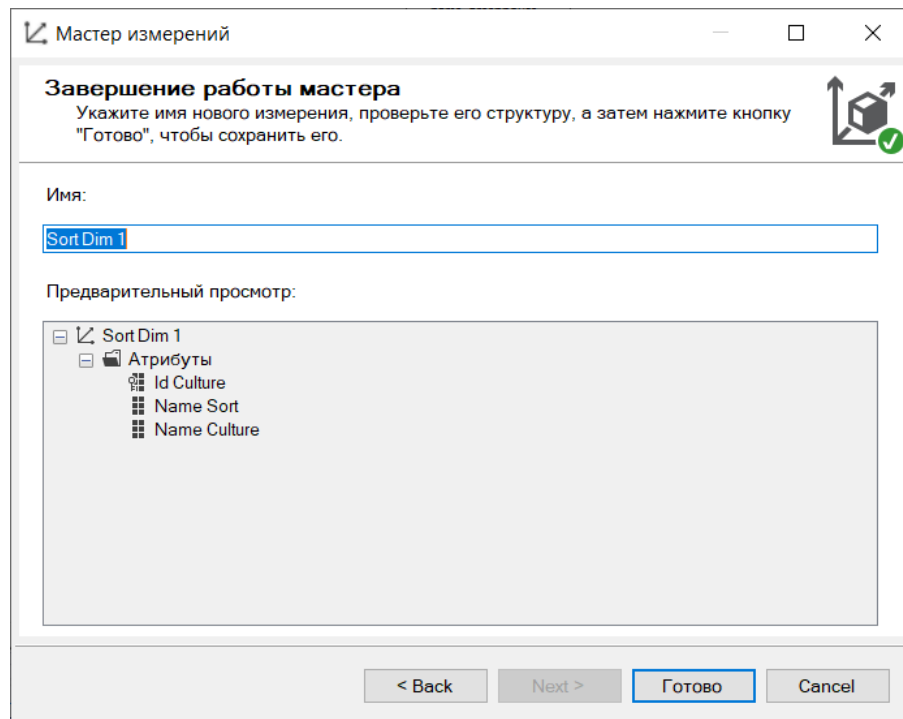


Рис. 12 Створення виміру сорту (SortDim)

Останнім виміром є вимір оптимальних умов, яка характеризує діапазон оптимальної температури та вологості.

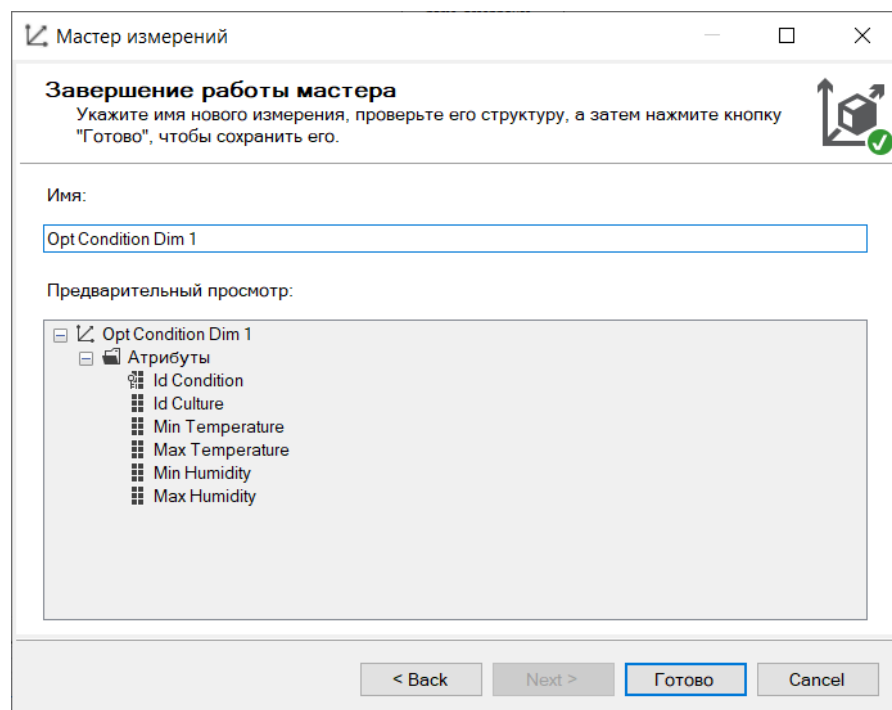


Рис. 13 Створення виміру оптимальних умов (OptConditionDim)

Після створення всіх необхідних вимірів наступним кроком є створення кубу OLAP за допомогою майстра кубів (Cube Wizard). На рис. 14 зображено створення кубу. На першому етапі необхідно обрати таблицю фактів, яка виступає основою для створення кубу.

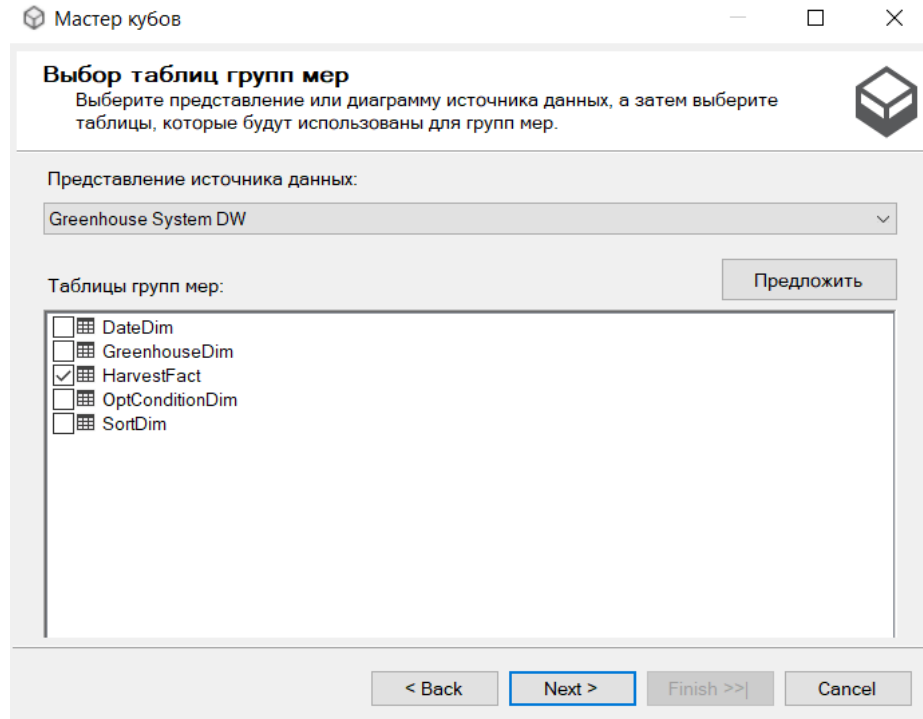


Рис. 14 Вибір таблиці для створення кубу

Наступним етапом іде вибір полів та вимірів, які формуватимуть куб.

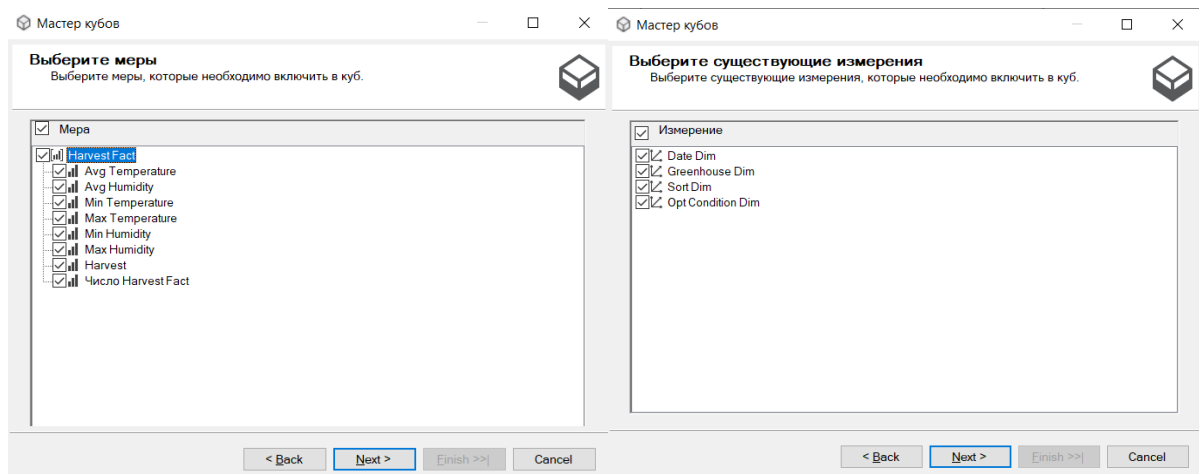


Рис. 15-16 Формування кубу

Після виконання всіх попередніх етапів результатом є створений Greenhouse System DW, що зображений на рис. 17.

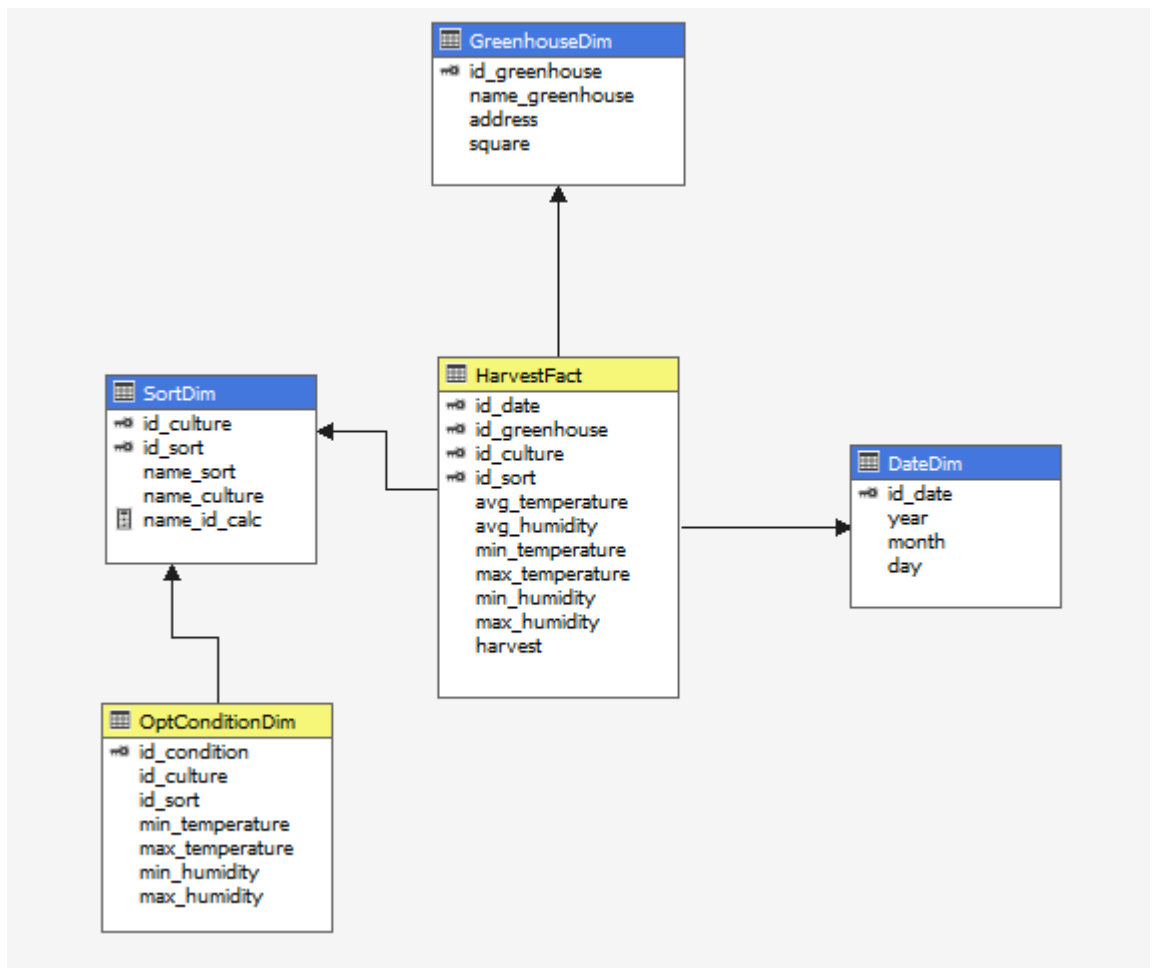


Рис. 17 Створений куб Greenhouse System DW

3.1.2 Реалізація отримання даних за допомогою Data Flow

Процес передачі даних було реалізовано за допомогою служби SQL Server Integration Services. SSIS – це інструмент, що дозволяє у зручному вигляді реалізувати інтеграцію, тобто. реалізувати процес перенесення даних з одного джерела до іншого. Цей процес іноді називають ETL (від англ. Extract, Transform, Load – дослівно «вилучення, перетворення, завантаження») [9].

В інструменті SSIS є служба Data Flow, за допомогою якої було проведено заповнення таблиць вимірів та фактів.

Наповнення сховища даних (СД) відбувається на основі оперативної БД, описаної у пункті 2.1 і поділено на 2 етапи (рис. 18).



Рис. 18 Потоки даних для наповнення СД

На першому етапі ми заповнюємо таблиці вимірів, дані для яких беруться, як було зазначено вище, з оперативної БД. На рис. 19 зображено потоки, які реалізують передачу даних з БД у СД.

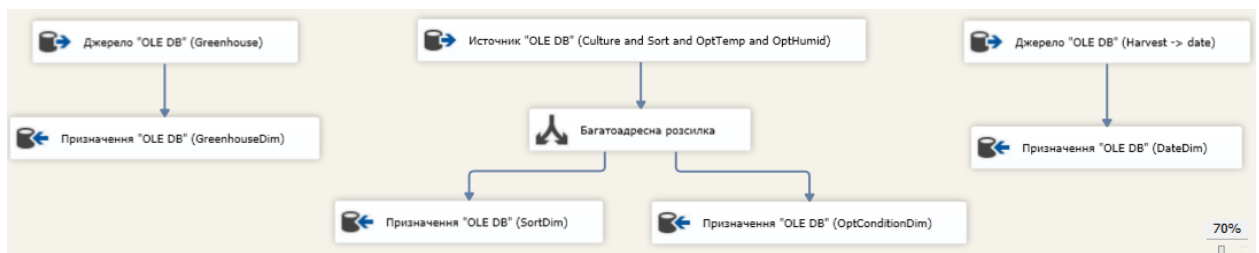


Рис. 19 Перший крок наповнення (таблиць-вимірів)

До прикладу нижче вказано процес заповнення DateDim.

Рис. 20 Формування запиту на основі джерела оперативної БД

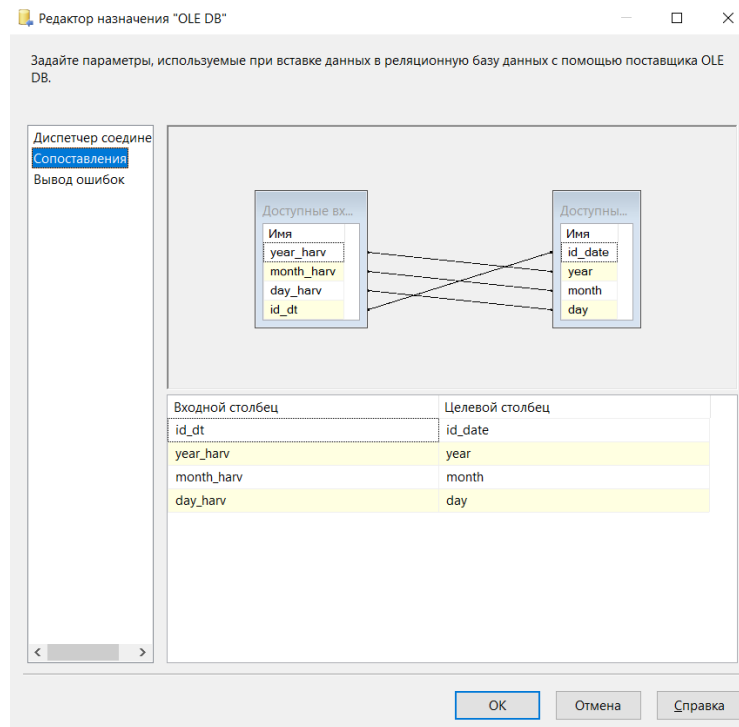


Рис. 21 Процес передачі даних на основі сформованої вибірки

На другому етапі проводиться заповнення таблиці фактів врожайностей. Запит, наведений нижче проводить вибірку врожайності з оперативної БД та дати, занесеної у СД на першому етапі.

```
SELECT GreenhouseSystem.dbo.Harvest.id_greenhouse,
GreenhouseSystem.dbo.Harvest.id_culture, GreenhouseSystem.dbo.Harvest.id_sort,
GreenhouseSystem.dbo.Harvest.harvest, GreenhouseSystem_DW.dbo.DateDim.id_date,
MIN(Temperature.[value]) AS min_temp, MAX(Temperature.[value]) AS max_temp,
AVG(Temperature.[value]) AS avg_temp,
MIN(Humidity.[value]) AS min_humid, MAX(Humidity.[value]) AS max_humid,
AVG(Humidity.[value]) AS avg_humid
FROM GreenhouseSystem.dbo.Harvest
JOIN GreenhouseSystem.dbo.Greenhouse ON Harvest.id_greenhouse =
Greenhouse.id_greenhouse
JOIN GreenhouseSystem.dbo.Temperature ON Greenhouse.id_greenhouse =
Temperature.id_greenhouse
JOIN GreenhouseSystem.dbo.Humidity ON Greenhouse.id_greenhouse =
Humidity.id_greenhouse
JOIN GreenhouseSystem_DW.dbo.DateDim ON DateDim.[year] =
CAST(YEAR(GreenhouseSystem.dbo.Harvest.finish_datetime) AS NVARCHAR(4))
AND DateDim.[month] = CAST(MONTH(GreenhouseSystem.dbo.Harvest.finish_datetime) AS
NVARCHAR(2))
AND DateDim.[day] = CAST(DAY(GreenhouseSystem.dbo.Harvest.finish_datetime) AS
NVARCHAR(2))
WHERE Temperature.create_datetime >= Harvest.start_datetime
AND Temperature.create_datetime <= Harvest.finish_datetime
AND Humidity.create_datetime >= Harvest.start_datetime
AND Humidity.create_datetime <= Harvest.finish_datetime
GROUP BY Harvest.id_greenhouse, Harvest.id_culture, Harvest.id_sort,
Harvest.finish_datetime, Harvest.harvest, DateDim.id_date
```

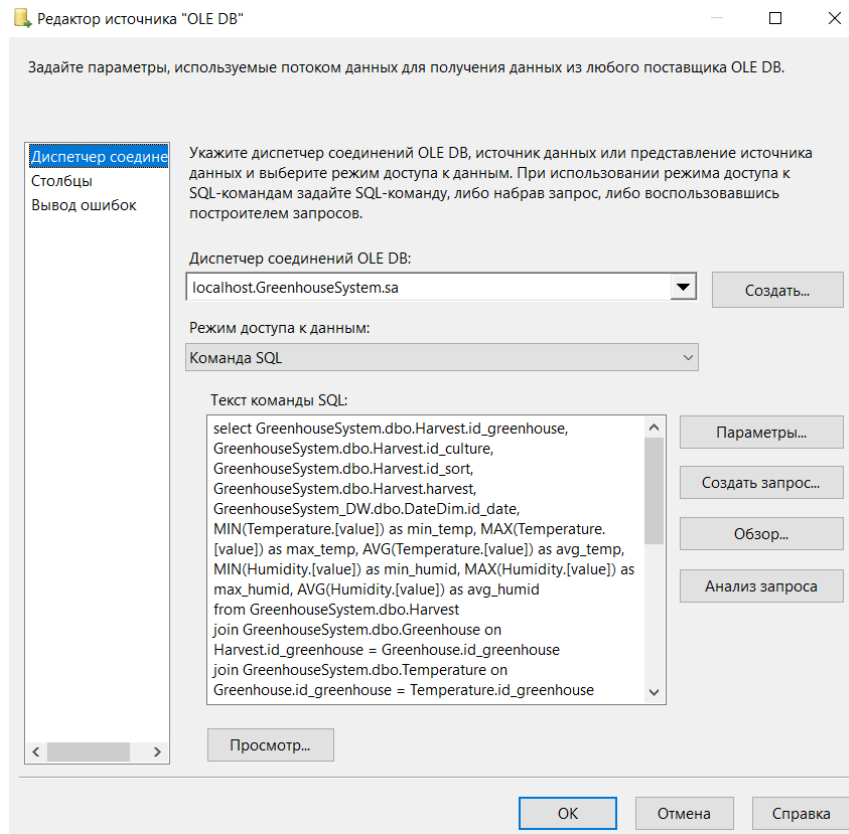


Рис. 22 Формування вибірки для фактів врожайності

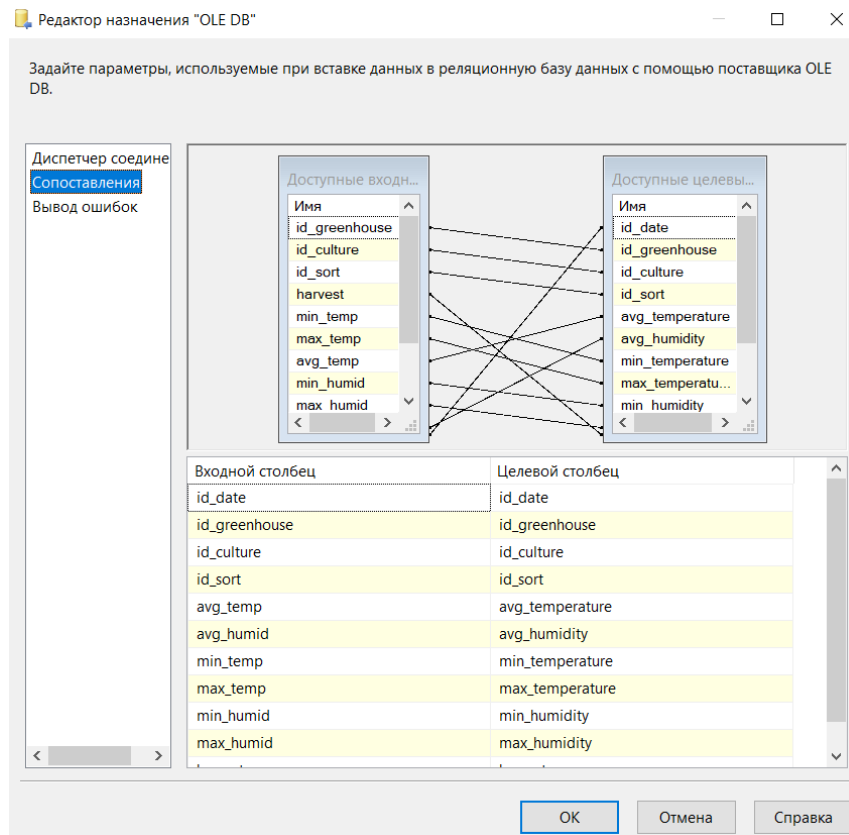


Рис. 23 Процесс передачи даних на основі сформованої вибірки

У результаті виконання всіх потоків у таблицях СД наявні дані, які перенесені з БД. На рис. 24-27 зображено результат виконання потоків.

Results Messages				
	id_greenhouse	name_greenhouse	address	square
1	1	Цех №1	Київська обл., Фастівський район, м. Боярка	5.00000
2	2	Цех №2	Київська обл., Фастівський район, м. Боярка	3.50000
3	3	Цех №3	Київська обл., Фастівський район, м. Боярка	4.00000

Results Messages							
	id_condition	id_culture	id_sort	min_temperature	max_temperature	min_humidity	max_humidity
1	11	1	1	20.00000	24.00000	65.00000	70.00000
2	12	1	2	22.00000	25.00000	60.00000	65.00000
3	13	1	3	18.00000	22.00000	60.00000	70.00000
4	24	2	4	18.00000	27.00000	75.00000	80.00000
5	25	2	5	18.00000	20.00000	75.00000	80.00000
6	26	2	6	22.00000	24.00000	70.00000	80.00000

Results Messages				
	id_culture	id_sort	name_sort	name_culture
1	1	1	Волове серце	Помідор
2	1	2	Черрі	Помідор
3	1	3	Корнійський	Помідор
4	2	4	Фенікс	Огірок
5	2	5	Амур	Огірок
6	2	6	Гарнец	Огірок

	id_date	id_greenhouse	id_culture	id_sort	avg_temperature	avg_humidity	min_temperature	max_temperature	min_humidity	max_humidity	harvest
1	2020820	1	1	1	22.33333	67.66666	22.00000	23.00000	66.00000	70.00000	4.10000
2	2020826	2	2	4	25.33333	78.00000	25.00000	26.00000	77.00000	79.00000	4.20000
3	2021420	1	1	1	19.00000	65.00000	17.00000	21.00000	64.00000	66.00000	3.40000
4	2021427	2	2	4	18.00000	75.66666	17.00000	19.00000	75.00000	76.00000	3.60000
5	2021715	2	2	4	20.00000	74.33333	20.00000	20.00000	73.00000	76.00000	3.50000
6	2021730	1	1	1	21.33333	65.33333	21.00000	22.00000	64.00000	67.00000	4.00000
7	20201118	1	1	1	22.00000	66.33333	21.00000	23.00000	65.00000	68.00000	3.90000
8	20201118	2	2	4	24.33333	78.00000	23.00000	26.00000	77.00000	79.00000	4.00000
9	20211025	2	2	4	21.66666	75.66666	21.00000	22.00000	74.00000	76.00000	3.00000

Рис. 24-27 Дані, занесені у СД після виконаних потоків

3.2 Побудова звітності в середовищі BI

Під час проведення аналізу даних одним з найважливіших етапів є формування звітності. Саме за допомогою звітів аналітик та підприємство можуть побачити цілісну картину розвитку підприємства.

Для створення звітної інформації була використана служба SQL Server Reporting Services (SSRS).

Служби SQL Server Reporting Services (SSRS) надають набір локальних засобів та служб для створення та розгортання мобільних звітів та звітів з розбиттям на сторінки, а також управління ними.

SQL Server Reporting Services пропонує оновлений набір продуктів:

- "Традиційні" звіти з розбиттям на сторінки були оновлені, щоб можна було створювати звіти, що виглядають по-сучасному, за допомогою оновлених засобів і нових функцій.
- Нові мобільні звіти з гнучким макетом, що адаптується під різні пристрої та способи їх утримання.
- Сучасний веб-портал, який можна відкрити у будь-якому сучасному браузері. На новому порталі можна впорядковувати та переглядати мобільні звіти, звіти з розбивкою на сторінки та ключові показники ефективності Reporting Services. Також можна зберігати книги Excel на порталі.

Процес створення звітів починається з підключення джерела даних. Далі за допомогою майстра створюємо звіт, використовуючи запити, які формуватимуть вибірку. У режимі конструктора можна редагувати зовнішній вигляд звіту та додати необхідні діаграми, які будуть наочно демонструвати результати вибірки.

Для відображення інформації було створено 3 звіти:

1. Екстремуми температури у розрізі врожайності сорту (рис. 28-30)

Створений запит для формування вибірки:

```
select GreenhouseDim.name_greenhouse, SortDim.name_culture, SortDim.name_sort,
DATEFROMPARTS(DateDim.[year], DateDim.[month], DateDim.[day]) as date_harvest,
HarvestFact.min_temperature, HarvestFact.max_temperature,
HarvestFact.min_humidity, HarvestFact.max_humidity
from HarvestFact
join GreenhouseDim on HarvestFact.id_greenhouse = GreenhouseDim.id_greenhouse
join SortDim on HarvestFact.id_sort = SortDim.id_sort
join DateDim on HarvestFact.id_date = DateDim.id_date
```

Екстремуми температури у розрізі врожайності сорту

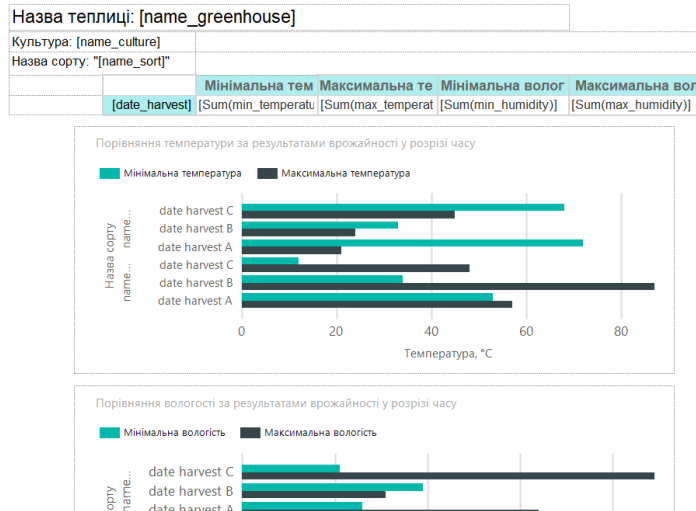


Рис. 28 Звіт у режимі конструктора

Екстремуми температури у розрізі врожайності сорту

Назва теплиці: Цех №1
Культура: Помідор
Назва сорту: "Болоньє серце"

	Мінімальна температура	Максимальна температура	Мінімальна вологість	Максимальна вологість
20.08.2020	22.00	23.00	66.00	70.00
18.11.2020	21.00	23.00	65.00	68.00
20.04.2021	17.00	21.00	64.00	66.00
30.07.2021	21.00	22.00	64.00	67.00
30.10.2021	22.00	24.00	66.00	69.00

Порівняння температури за результатами врожайності у розрізі часу



Порівняння вологості за результатами врожайності у розрізі часу



Назва теплиці: Цех №2

Культура: Огірок

Назва сорту: "Фенікс"

	Мінімальна температура	Максимальна температура	Мінімальна вологість	Максимальна вологість
26.08.2020	25.00	26.00	77.00	79.00
18.11.2020	23.00	26.00	77.00	79.00
27.04.2021	17.00	19.00	75.00	76.00
15.07.2021	20.00	20.00	73.00	76.00
25.10.2021	21.00	22.00	74.00	78.00

Порівняння температури за результатами врожайності у розрізі часу



Порівняння вологості за результатами врожайності у розрізі часу



Рис. 29-30 Результат формування звіту

2. Зміна врожайності у розрізі сорту та часу

Створений запит для формування вибірки:

```
select GreenhouseDim.name_greenhouse, SortDim.name_culture, SortDim.name_sort,
DATEFROMPARTS(DateDim.[year], DateDim.[month], DateDim.[day]) as date_harvest,
HarvestFact.harvest
from HarvestFact
join GreenhouseDim on HarvestFact.id_greenhouse = GreenhouseDim.id_greenhouse
join SortDim on HarvestFact.id_sort = SortDim.id_sort
join DateDim on HarvestFact.id_date = DateDim.id_date
```

Зміна врожайності у розрізі сорту

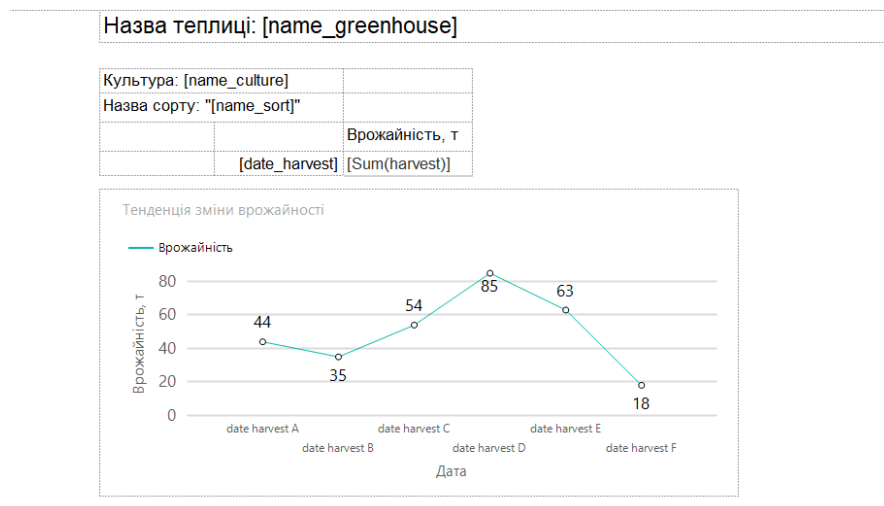
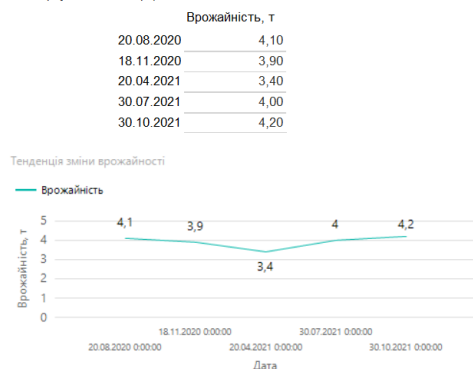


Рис. 31 Звіт у режимі конструктора

Зміна врожайності у розрізі сорту

Назва теплиці: Цех №1

Культура: Помідор
Назва сорту: "Волове серце"



Назва теплиці: Цех №2

Культура: Огірок
Назва сорту: "Фенікс"

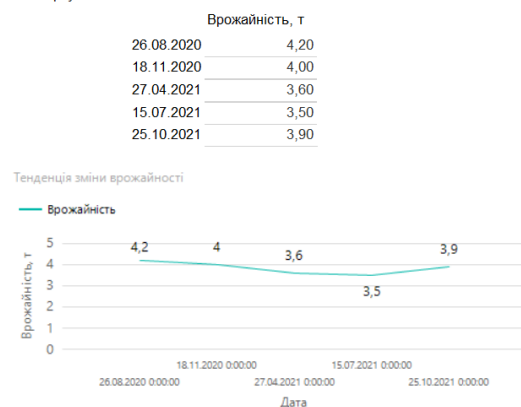


Рис. 32-33 Результат формування звіту

3. Аналіз температури та вологості у розрізі оптимальних показників

Створений запит для формування вибірки:

```
select GreenhouseDim.name_greenhouse, SortDim.name_culture, SortDim.name_sort,
DATEFROMPARTS(DateDim.[year], DateDim.[month], DateDim.[day]) as date_harvest,
HarvestFact.avg_temperature, HarvestFact.avg_humidity,
OptConditionDim.min_temperature, OptConditionDim.max_temperature,
OptConditionDim.min_humidity, OptConditionDim.max_humidity,
HarvestFact.harvest
from HarvestFact
join GreenhouseDim on HarvestFact.id_greenhouse = GreenhouseDim.id_greenhouse
join SortDim on HarvestFact.id_sort = SortDim.id_sort
join DateDim on HarvestFact.id_date = DateDim.id_date
join OptConditionDim on HarvestFact.id_sort = OptConditionDim.id_sort
```

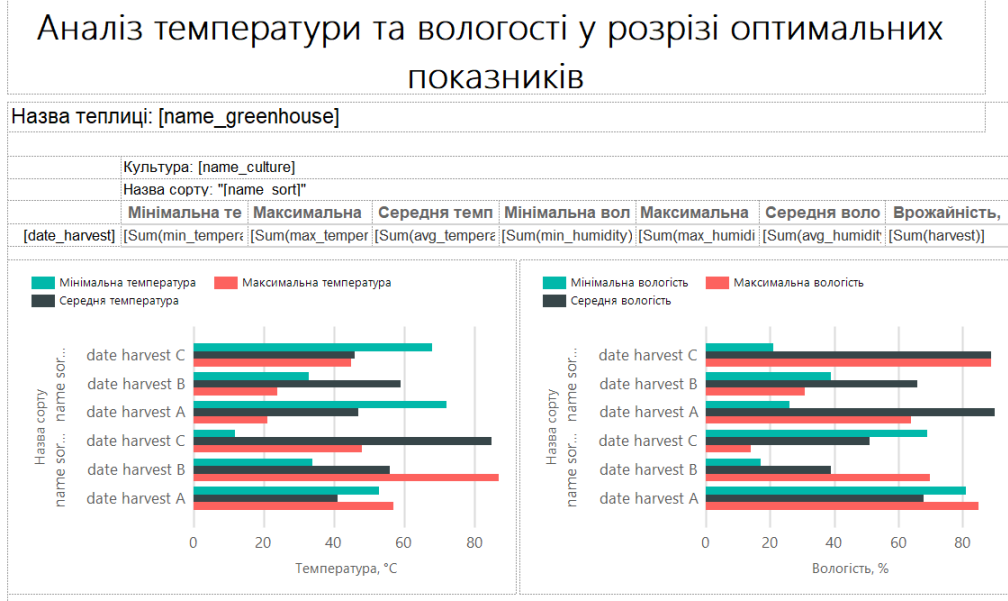
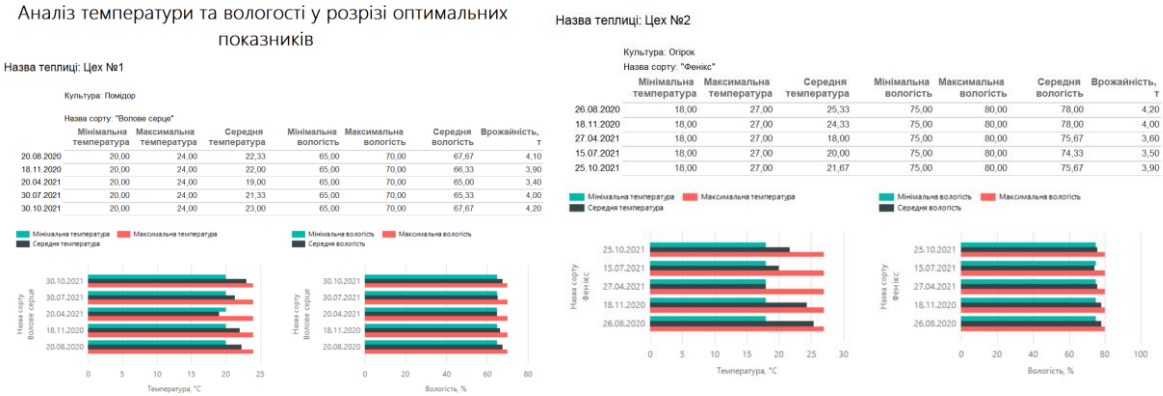


Рис. 34 Звіт у режимі конструктора



Назва теплиці: Цех №2

Культура: Огірок

Назва сорту: "Фенікс"

Мінімальна температура

Максимальна температура

Середня температура

Мінімальна вологість

Максимальна вологість

Середня вологість

Врожайність, т

26.08.2020

18.11.2020

27.04.2021

15.07.2021

25.10.2021

18,00

27,00

25,33

75,00

80,00

78,00

4,20

25.10.2021

15.07.2021

27.04.2021

18.11.2020

26.08.2020

18,00

27,00

21,67

75,00

80,00

75,67

3,90

Назва сорту: "Фенікс"

Температура, °C

25.10.2021

15.07.2021

27.04.2021

18.11.2020

26.08.2020

0

5

10

15

20

25

30

Назва сорту: "Фенікс"

Вологість, %

25.10.2021

15.07.2021

27.04.2021

18.11.2020

26.08.2020

0

20

40

60

80

100

Рис. 35-36 Результат формування звіту

3.3 Розрахунок КРІ

Для визначення успішності роботи підприємства аналітики обчислюють коефіцієнт ефективності.

КРІ — це набір обчислень, пов'язаних з групою показників у кубі, які використовуються для оцінки успіху бізнесу. Зазвичай ці обчислення є комбінацією виразів багатовимірних виразів (MDX) або обчислюваних елементів. КРІ також мають додаткові метадані, які надають інформацію про те, як клієнтські програми повинні відображати результати обчислень КРІ.

KPI обробляє інформацію про поставлену ціль, фактичну формулу продуктивності, записану в кубі, і вимірювання, щоб показати тенденцію та статус ефективності.

Компоненти для розрахунку KPI:

- Ціль – числовий вираз MDX або обчислення, яке повертає цільове значення KPI.
- Значення – числовий вираз MDX, який повертає фактичне значення KPI.
- Статус – вираз MDX, який представляє стан KPI на певний момент часу. Вираз MDX стану має повертати нормалізоване значення від -1 до 1. Значення, рівні або менші за -1, будуть інтерпретуватися як "погані" або "низькі". Значення нуль (0) інтерпретується як «прийнятне» або «середнє». Значення, що дорівнюють або перевищують 1, будуть інтерпретуватися як «добре» або «високе» [11].

Для аналізу фактів, занесених у куб, було визначено такі KPI:

1. KPI_Sort1_Opt_Avg_Temp – визначення ефективності середньої температури за першим сортом.
2. KPI_Sort1_Opt_Avg_Humidity – визначення ефективності середньої вологості за першим сортом.
3. KPI_Sort4_Opt_Avg_Temp – визначення ефективності середньої температури за четвертим сортом.
4. KPI_Sort4_Opt_Avg_Humidity – визначення ефективності середньої температури за четвертим сортом.

Кожен з показників показує ефективність температури та вологості у розрізі сорту.

Для розрахунку KPI необхідно розраховане фактичне значення, цільове значення та зазначені умови для визначення статусу (рис. 37-39).

Выражение значения

```
(([Measures].[Avg Temperature], [Sort Dim].[Id Sort].&[1],[Date Dim].[Year].&[2021]) / ([Measures].[Count Harvest Fact], [Sort Dim].[Id Sort].&[1],[Date Dim].[Year].&[2021]))
```

Проблемы не найдены. Стр: 1 Симв: 173 Столб: 87 Пробелы CRLF

Рис. 37 Розрахунок фактичного значення

Целевое выражение

```
(([Measures].[Min Temperature - Opt Condition Dim], [Sort Dim].[Id Sort].&[1]) + ([Measures].[Max Temperature - Opt Condition Dim], [Sort Dim].[Id Sort].&[1]) ) / 2
```

Проблемы не найдены. Стр: 1 Симв: 166 Столб: 84 Пробелы CRLF

Рис. 38 Розрахунок цільового значення

Состояние

Признак состояния: Шкала

Выражение состояния:

```
CASE
WHEN KPIVALUE( "KPI_Sort1_Opt_Avg_Temp" ) = KPIGOAL( "KPI_Sort1_Opt_Avg_Temp" ) THEN 1
WHEN KPIVALUE( "KPI_Sort1_Opt_Avg_Temp" ) >= ([Measures].[Min Temperature - Opt Condition Dim], [Sort Dim].[Id Sort].&[1]) AND KPIVALUE( "KPI_Sort1_Opt_Avg_Temp" ) <= ([Measures].[Max Temperature - Opt Condition
```

Проблемы не найдены. Стр: 6 Симв: 4 Пробелы CRLF

Рис. 39 Визначення статусу за зазначеними умовами

Також для порівняння даних за попередній часовий проміжок можна розраховувати тренд (рис. 40).

Тренд

Признак тренда: Стандартная стрелка

Выражение тренда:

```
CASE
WHEN ([Measures].[Avg Temperature], [Sort Dim].[Id Sort].&[1],[Date Dim].[Year].&[2021]) / ([Measures].[Count Harvest Fact], [Sort Dim].[Id Sort].&[1],[Date Dim].[Year].&[2021]) > ([Measures].[Avg Temperature], [Sort Dim].[Id Sort].&[1],[Date Dim].[Year].&[2020]) / ([Measures].[Count Harvest Fact], [Sort Dim].[Id Sort].&[1],[Date Dim].[Year].&[2020])
```

Рис. 40 Розрахунок тренду

Результат розрахунку KPI зображено на рис. 41.

Отобразить структуру	Значение	Цель	Состояние	Тренд
KPI_Sort1_Opt_Avg_Humidity	65.9999966666667	67.5		↓
KPI_Sort1_Opt_Avg_Temp	21.11111	22		↓
KPI_Sort4_Opt_Avg_Humidity	75.2222166666667	77.5		↓
KPI_Sort4_Opt_Avg_Temp	19.8888866666667	22.5		↓

Рис. 41 Результати розрахунку KPI

На основі розрахованих даних можна зробити висновок, що середня температура та вологість за сортами знаходиться у межах норми, проте знизилась порівняно з минулим роком.

ВИСНОВКИ

У ході виконання курсового проєкту було розроблено систему аналізу тепличного господарства за допомогою OLAP технологій.

На першому етапі було сформовано постановку завдання, проаналізовано предметну область за допомогою діаграми прецедентів, сформовано список акторів та описано архітектуру розроблюваної системи.

На другому етапі було описано оперативну БД, з'ясовано поняття про OLAP технології та спроектовано сховище даних, яке наповнюється на основі даних з оперативної БД.

На третьому етапі було створено та розгорнуто гіперкуб на основі сховища даних, створеного у 2-му етапі, реалізовано передачу даних з оперативної БД за допомогою служби Data Flow. Також було сформовано звітну інформацію та розраховано ключовий показник ефективності підприємства.

Під час виконання проєкту було використано СУБД SQL Server Management Studio для створення БД та СД, середовище розробки Visual Studio з інтегрованими службами Integration Services, Integration Services, Reporting Services для аналізу даних.

У результаті виконання проєкту було створено програмне рішення, яке дозволяє реалізовувати потоки даних з БД у СД та аналізувати великий обсяг інформації для досконалого вивчення всіх особливостей вирощування культур.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Use Cases. – [Електронний ресурс]. – Режим доступу: <https://www.usability.gov/how-to-and-tools/methods/use-cases.html>
2. Use Case. – [Електронний ресурс]. – Режим доступу: <https://searchsoftwarequality.techtarget.com/definition/use-case>
3. Overview of Online Analytical Processing (OLAP). – [Електронний ресурс]. – Режим доступу: <https://support.microsoft.com/en-us/office/overview-of-online-analytical-processing-olap-15d2cdde-f70b-4277-b009-ed732b75fdd6>
4. Що таке OLAP? – [Електронний ресурс]. – Режим доступу: <https://uk.education-wiki.com/5528785-what-is-olap>
5. What is MOLAP (Multidimensional OLAP) in Data Warehouse? – [Електронний ресурс]. – Режим доступу: <https://www.guru99.com/multidimensional-online-analytical-processing.html>
6. What Is a Data Warehouse? – [Електронний ресурс]. – Режим доступу: <https://www.oracle.com/database/what-is-a-data-warehouse/>
7. SSAS Tutorial: What is SSAS Cube, Architecture & Types . – [Електронний ресурс]. – Режим доступу: <https://www.guru99.com/ssas-tutorial.html>
8. Data Source Views in Multidimensional Models. – [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/en-us/analysis-services/multidimensional-models/data-source-views-in-multidimensional-models?view=asallproducts-allversions>
9. SQL Server Integration Services (SSIS) для начинающих – часть 1. – [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/330618/>
10. Что такое службы SQL Server Reporting Services (SSRS)? – [Електронний ресурс]. – Режим доступу:

<https://docs.microsoft.com/ru-ru/sql/reporting-services/create-deploy-and-manage-mobile-and-paginated-reports?view=sql-server-ver15>

11. Key Performance Indicators (KPIs) in Multidimensional Models. – [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/en-us/analysis-services/multidimensional-models/key-performance-indicators-kpis-in-multidimensional-models?view=asallproducts-allversions>

ДОДАТКИ

ЗАПИТИ СТВОРЕННЯ ОБ'ЄКТІВ СД

```

-- create tables
-- create date dimension table
USE GreenhouseSystem_DW
GO
IF EXISTS (SELECT name FROM sys.objects
           WHERE name = 'DateDim' AND type_desc='USER_TABLE')
DROP TABLE DateDim
GO
CREATE TABLE DateDim (
id_date INT NOT NULL PRIMARY KEY ,
[year] NVARCHAR(4) NOT NULL,
[month] NVARCHAR(2) NOT NULL,
[day] NVARCHAR(2) NOT NULL
);

-- create greenhouse dimension table
IF EXISTS (SELECT name FROM sys.objects
           WHERE name = 'GreenhouseDim' AND type_desc='USER_TABLE')
DROP TABLE GreenhouseDim
GO
CREATE TABLE GreenhouseDim (
id_greenhouse INT NOT NULL PRIMARY KEY,
[name_greenhouse] NVARCHAR(100) NOT NULL,
[address] NVARCHAR(200) NOT NULL,
[square] DECIMAL(10,5) NOT NULL
);

-- create sort dimension table
IF EXISTS (SELECT name FROM sys.objects
           WHERE name = 'SortDim' AND type_desc='USER_TABLE')
DROP TABLE SortDim
GO
CREATE TABLE SortDim (
id_culture INT NOT NULL,
id_sort INT NOT NULL,
[name_sort] NVARCHAR(100) NOT NULL,
[name_culture] NVARCHAR(100) NOT NULL,
);

ALTER TABLE SortDim
ADD CONSTRAINT XPKSortDim PRIMARY KEY CLUSTERED (id_culture ASC, id_sort ASC)
GO

-- create optimal condition dimension table
IF EXISTS (SELECT name FROM sys.objects
           WHERE name = 'OptConditionDim' AND type_desc='USER_TABLE')
DROP TABLE OptConditionDim
GO
CREATE TABLE OptConditionDim (
id_condition INT NOT NULL PRIMARY KEY,
id_culture INT NOT NULL,
id_sort INT NOT NULL,
min_temperature DECIMAL(10,5) NOT NULL,
max_temperature DECIMAL(10,5) NOT NULL,
min_humidity DECIMAL(10,5) NOT NULL,
max_humidity DECIMAL(10,5) NOT NULL,
);

ALTER TABLE OptConditionDim
ADD CONSTRAINT FK_OptConditionDim FOREIGN KEY (id_culture, id_sort) REFERENCES
SortDim(id_culture, id_sort)
ON DELETE CASCADE
ON UPDATE CASCADE
go

```

```

-- create harvest fact table
IF EXISTS (SELECT name FROM sys.objects
           WHERE name = 'HarvestFact' AND type_desc='USER_TABLE')
DROP TABLE HarvestFact
GO
CREATE TABLE HarvestFact (
id_date INT NOT NULL,
id_greenhouse INT NOT NULL,
id_culture INT NOT NULL,
id_sort INT NOT NULL,
avg_temperature DECIMAL(10,5) NOT NULL,
avg_humidity DECIMAL(10,5) NOT NULL,
min_temperature DECIMAL(10,5) NOT NULL,
max_temperature DECIMAL(10,5) NOT NULL,
min_humidity DECIMAL(10,5) NOT NULL,
max_humidity DECIMAL(10,5) NOT NULL,
harvest DECIMAL(10,5) NOT NULL,
-- FOREIGN KEY (id_date) REFERENCES DateDim(id_date) ON DELETE CASCADE,
--FOREIGN KEY (id_greenhouse) REFERENCES GreenhouseDim(id_greenhouse) ON DELETE
CASCADE,
--FOREIGN KEY (id_culture, id_sort) REFERENCES SortDim(id_culture, id_sort) ON DELETE
CASCADE
);

ALTER TABLE HarvestFact
    ADD CONSTRAINT XPKHarvestFact PRIMARY KEY CLUSTERED (id_date ASC,
id_greenhouse ASC, id_culture ASC, id_sort ASC)
GO

ALTER TABLE HarvestFact
    ADD CONSTRAINT Constr_SortDim FOREIGN KEY (id_culture, id_sort) references
SortDim(id_culture, id_sort)
    ON DELETE CASCADE
    ON UPDATE CASCADE
GO

ALTER TABLE HarvestFact
    ADD CONSTRAINT Constr_Date_HarvestFact FOREIGN KEY (id_date) REFERENCES
DateDim(id_date)
    ON DELETE CASCADE
    ON UPDATE CASCADE
GO

ALTER TABLE HarvestFact
    ADD CONSTRAINT Constr_Greenh_HarvestFact FOREIGN KEY (id_greenhouse) REFERENCES
GreenhouseDim(id_greenhouse)
    ON DELETE CASCADE
    ON UPDATE CASCADE
GO

```