

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”  
Інститут прикладної математики та фундаментальних наук  
Кафедра прикладної математики

Курсова робота

На тему: " Розробка та аналіз надвеликої бази даних з використанням  
технологій Microsoft SQL Server"  
З курсу “Надвеликі бази даних”

Виконав:

студентка групи ПМ-43

Горбаль Анастасія

Прийняв:

доцент кафедри прикладної математики

Любінський Б. Б.

Львів 2025

## ВСТУП

**Актуальність теми:** В умовах сучасної цифрової економіки інформаційні ресурси відіграють визначальну роль у діяльності підприємств різних галузей. Для компаній, що працюють у сфері проектної діяльності, особливо важливими є ефективне управління проектами, контроль виконання робіт, облік робочого часу персоналу та аналіз фінансових показників. Із розширенням масштабів діяльності організацій суттєво зростають обсяги даних, які необхідно накопичувати, обробляти та аналізувати з метою підтримки управлінських рішень.

Традиційні підходи до зберігання та обробки інформації не завжди забезпечують достатню продуктивність і гнучкість при роботі з великими масивами даних, що може призводити до затримок у формуванні аналітичної звітності та зниження ефективності управління. У зв'язку з цим зростає потреба у використанні систем керування надвеликими базами даних (VLDB) та інструментів бізнес-аналітики (Business Intelligence), які дозволяють централізувати дані та виконувати їх багатовимірний аналіз.

Платформа Microsoft SQL Server надає комплексний набір засобів для побудови таких рішень, зокрема SQL Server Integration Services (SSIS) для реалізації ETL-процесів, SQL Server Analysis Services (SSAS) для створення OLAP-кубів та SQL Server Reporting Services (SSRS) для формування аналітичної звітності. Використання даного стеку технологій дозволяє перетворити операційні дані на структуровану аналітичну інформацію, придатну для прийняття обґрунтованих управлінських рішень.

У даній курсовій роботі розглядається розробка інформаційно-аналітичної системи для предметної області «Виконання проектів». Актуальність обраної теми зумовлена необхідністю автоматизації процесів обліку виконаних робіт,

розрахунку заробітної плати працівників та аналізу фінансової ефективності проектів на основі значних обсягів історичних даних.

**Метою курсової роботи** є розробка повнофункціональної системи керування надвеликою базою даних для предметної області «Виконання проектів» із реалізацією ETL-процесів, побудовою багатовимірного OLAP-куба та створенням аналітичних звітів з використанням технологій Microsoft SQL Server.

Щоб досягти поставленої мети у роботі необхідно вирішити такі завдання:

1. Провести аналіз предметної області «Виконання проектів», визначити основні бізнес-процеси та вимоги до інформаційної системи.
2. Спроектувати концептуальну, логічну та фізичну моделі реляційної бази даних з урахуванням вимог цілісності та нормалізації.
3. Згенерувати великий обсяг тестових даних (понад 500 000 записів), що імітують діяльність компанії протягом кількох років.
4. Розробити сховище даних та реалізувати ETL-процеси для перенесення і трансформації даних з операційної бази до аналітичної.
5. Побудувати OLAP-куб із необхідними вимірами та мірами для багатовимірного аналізу.
6. Створити набір аналітичних звітів різного типу засобами SSRS для візуалізації ключових показників діяльності.

**Об'єктом дослідження** є процеси управління проектами та аналізу даних про виконання робіт, фінансові витрати та завантаженість персоналу в проектно-орієнтованій організації.

**Предметом дослідження** є методи та засоби проєктування надвеликих баз даних, побудови сховищ даних, реалізації ETL-процесів і систем бізнес-аналітики на базі платформи Microsoft SQL Server. Методи дослідження. У роботі використано методи системного аналізу для дослідження предметної області, методи теорії баз даних для проєктування структури БД (ER-моделювання, нормалізація), а також інструментальні засоби середовища Microsoft Visual Studio та SQL Server Data Tools (SSDT) для реалізації практичної частини роботи.

**Практичне значення** отриманих результатів полягає у розробці функціональної інформаційної системи, що забезпечує автоматизацію типових операцій з обробки даних про проекти, надає керівництву підприємства своєчасну аналітичну звітність та сприяє прийняттю обґрунтованих управлінських рішень щодо ефективного розподілу ресурсів і фінансового планування.

## РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1. Характеристика предметної області "Виконання проектів"

В умовах сучасної ринкової економіки проектна діяльність є ключовою складовою функціонування багатьох підприємств, зокрема у сферах інформаційних технологій, консалтингу, будівництва та маркетингу. Рівень ефективності управління проектами безпосередньо визначає фінансові результати діяльності компаній та впливає на її ділову репутацію. Предметна область «Виконання проектів» охоплює сукупність взаємопов'язаних процесів, спрямованих на досягнення запланованих цілей у межах встановлених термінів та бюджетних обмежень.

Основними сутностями даної предметної області є проекти, замовники, виконавці (персонал), а також фінансові операції, пов'язані з виставленням

рахунків і нарахуванням заробітної плати. Однією з головних проблем є необхідність інтеграції розрізнених інформаційних ресурсів: облік робочого часу може здійснюватися в окремих системах, дані про клієнтів зберігатися в інших, а фінансовий облік - у незалежних підсистемах. Така фрагментованість ускладнює формування цілісного уявлення про ефективність реалізації проектів та діяльності підприємства в цілому.

Створення централізованої системи керування надвеликою базою даних дозволяє усунути вказані недоліки та забезпечує вирішення таких проблем:

- втрата або неповнота інформації щодо фактично використаних ресурсів у межах проектів;
- помилки при розрахунку заробітної плати на основі погодинних ставок працівників;
- несвоєчасне формування та виставлення рахунків замовникам;
- відсутність аналітичних інструментів для оцінки рентабельності окремих проектів та їх категорій.

## **1.2. Аналіз бізнес-процесів**

Для коректного проектування бази даних необхідно детально проаналізувати основні бізнес-процеси компанії. Виділимо ключові процеси, які підлягають автоматизації:

- 1. Ініціалізація проекту.** Менеджер реєструє нового замовника (або вибирає існуючого) та створює картку проекту. Обов'язково вказуються терміни виконання (Start Date, End Date), бюджет та категорія проекту (наприклад, "Веб-розробка", "Консалтинг", "Міграція у хмару").

2. **Облік виконання робіт (Work Logging).** Виконавці (Employees) щоденно вносять дані про виконану роботу. Фіксується дата, проект, кількість витрачених годин та опис задачі. Цей процес є критичним, оскільки на його основі формується собівартість проекту та зарплата співробітника.
3. **Фінансові розрахунки із замовниками (Invoicing).** На основі етапів виконання проекту або витраченого часу формуються рахунки (Invoices). Система повинна відстежувати статус оплати (IsPaid) та дату платежу.
4. **Нарахування заробітної плати.** В кінці звітного періоду (місяця) відбувається розрахунок виплат співробітникам. Система аналізує всі записи про роботу (WorkLogs), зіставляє їх з погодинною ставкою посади (Position Rate) та розраховує підсумкову суму до виплати, включаючи можливі бонуси.

### **1.3. Огляд існуючих рішень (аналогів)**

Розробка власної системи на базі MS SQL Server дозволяє створити кастомізоване рішення, яке ідеально відповідає специфічним вимогам підприємства, забезпечує високу продуктивність при роботі з великими обсягами даних (Big Data) та дозволяє будувати складні аналітичні звіти без додаткових ліцензійних витрат.

### **1.4. Визначення вимог до системи**

#### **Функціональні вимоги:**

1. Система повинна забезпечувати збереження інформації про не менш ніж 5 000 проектів та 500 000 записів про роботу.
2. Можливість додавання, редагування та видалення даних про клієнтів, співробітників та проекти.

3. Автоматичний розрахунок суми рахунків та заробітної плати.
4. Підтримка історичних даних за 5 років для аналізу трендів.

### **Нефункціональні вимоги:**

- Цілісність даних:** Забезпечення посилальної цілісності між таблицями (неможливість видалити проект, який має активні задачі).
- Продуктивність:** Час виконання складних аналітичних запитів не повинен перевищувати прийнятних меж (до кількох секунд) завдяки індексації.
- Масштабованість:** Можливість збільшення обсягу даних без зміни архітектури.

### **1.5. Характеристика користувачів системи та їх ролей**

Система проєктується як багатокористувацьке середовище з розмежуванням прав доступу. Для забезпечення інформаційної безпеки та коректності бізнес-процесів необхідно виділити основні групи користувачів (акторів) та визначити їхні функціональні можливості.

В рамках предметної області "Виконання проектів" виділено наступні ролі:

- Менеджер проектів (Project Manager).** Це ключова фігура в системі, яка відповідає за планування та контроль виконання робіт.
  - Функції:** Створення нових проектів, призначення виконавців, встановлення бюджетних обмежень, моніторинг дедлайнів, спілкування із замовниками.
  - Потреби в інформації:** Менеджеру необхідні зведені звіти про стан проектів (Dashboard), діаграми Ганта для відстеження часових рамок та попередження про перевитрати бюджету.
- Виконавець (Employee / Developer).** Технічний спеціаліст або консультант, який безпосередньо виконує задачі.

- **Функції:** Перегляд призначених завдань, внесення даних про витрачений час (Log Work), оновлення статусу виконання задач (Active -> Completed).
  - **Потреби в інформації:** Чіткий список своїх задач, історія власних трудовитрат.
3. **Бухгалтер / Фінансовий менеджер.** Відповідає за фінансову сторону діяльності компанії.
- **Функції:** Формування рахунків (Invoices) для клієнтів на основі звітів про роботу, контроль оплати рахунків, нарахування заробітної плати працівникам.
  - **Потреби в інформації:** Деталізовані фінансові звіти, акти виконаних робіт, відомості про заборгованість клієнтів.
4. **Адміністратор системи.** Технічний спеціаліст, що забезпечує працездатність бази даних.
- **Функції:** Керування обліковими записами користувачів, налаштування довідників (категорії проектів, посади), резервне копіювання даних (Backup), моніторинг продуктивності сервера.

Таке розмежування дозволяє уникнути конфліктів даних та забезпечити конфіденційність фінансової інформації (наприклад, розробники не повинні бачити зарплати колег).

## **1.6. Формалізація бізнес-правил та обмежень**

Для забезпечення цілісності даних на етапі проектування бази даних необхідно чітко сформулювати бізнес-правила, які будуть реалізовані через механізми обмежень (Constraints), тригерів та збережених процедур.

### **Основні бізнес-правила:**

#### **1. Часові обмеження:**

- Дата завершення проекту не може бути ранішою за дату його початку ( $\text{EndDate} \geq \text{StartDate}$ ).
- Дата звіту про роботу ( $\text{WorkDate}$ ) не може бути в майбутньому.
- Дата звіту про роботу повинна знаходитися в межах часового інтервалу активності проекту.

## 2. **Фінансові обмеження:**

- Погодинна ставка працівника ( $\text{HourlyRate}$ ) повинна бути більше нуля.
- Сума рахунку ( $\text{InvoiceAmount}$ ) не може бути від'ємною.
- Загальна сума виплаченої зарплати не може перевищувати нарахований фонд оплати праці.

## 3. **Структурні обмеження:**

- Кожен проект повинен мати обов'язкову прив'язку до Замовника (**Customer**).
- Видалення запису про Замовника неможливе, якщо у системі існують пов'язані з ним проекти або неоплачені рахунки (заборона каскадного видалення для критичних даних).
- Один співробітник може працювати над кількома проектами одночасно, але сумарна кількість залогованих годин за добу не повинна перевищувати фізично можливий ліміт (наприклад, 24 години).

## 4. **Правила станів (Status Flow):**

- Проект не може бути переведений у статус "Завершено", якщо по нему є незакриті критичні задачі.
- Рахунок не може бути змінений (відредагований) після того, як він отримав статус "Оплачено".

## 1.7. Обґрунтування вибору інструментальних засобів

Для реалізації інформаційної системи необхідно обрати систему управління базами даних (СУБД), яка б задовольняла вимогам надійності, масштабованості та мала розвинені засоби бізнес-аналітики. Проведемо порівняльний аналіз найбільш популярних реляційних СУБД: Microsoft SQL Server, MySQL та PostgreSQL.

*Таблиця 1.1. Порівняльний аналіз СУБД*

Критерій порівняння	Microsoft SQL Server	MySQL	PostgreSQL
Тип ліцензії	Комерційна (є безкоштовна Developer/Express версія)	Open Source (GPL)	Open Source (BSD)
Підтримка OLAP	Вбудована потужна служба SSAS (Analysis Services)	Відсутня (потребує сторонніх інструментів)	Відсутня (потребує сторонніх інструментів)
ETL інструменти	Вбудований SSIS (Integration Services)	Відсутній	Сторонні рішення (Pentaho, Talend)
Звітність	Вбудований SSRS (Reporting Services)	Відсутній	Сторонні рішення (JasperReports)
Transact-SQL	Потужна процедурна мова Т-SQL	Обмежена підтримка процедур	Потужна мова PL/pgSQL

Критерій порівняння	Microsoft SQL Server	MySQL	PostgreSQL
Зручність розробки	Висока (SQL Server Management Studio, Visual Studio)	Середня (MySQL Workbench)	Середня (pgAdmin)

**Висновки щодо вибору:** Для виконання курсової роботи обрано **Microsoft SQL Server 2019**, оскільки це єдина СУБД із розглянутих, яка надає повний стек технологій "з коробки" (All-in-one):

1. **Database Engine** - для зберігання даних (OLTP).
2. **SSIS** - для реалізації ETL-процесів та міграції даних.
3. **SSAS** - для побудови багатовимірного куба та OLAP-аналізу.
4. **SSRS** - для створення регламентованої звітності.

Використання єдиної екосистеми Microsoft значно спрощує процес розробки, розгортання та адміністрування системи, а середовище Visual Studio надає зручні інструменти для роботи з усіма компонентами проекту.

## 1.8. Концептуальне моделювання даних

На основі аналізу предметної області було виділено основні сутності та зв'язки між ними. Основні сутності системи:

- **Employees (Виконавці):** Зберігає дані про персонал. Атрибути: ПІБ, email, телефон, дата найму.

- **Positions (Посади):** Довідник посад із зазначенням базової погодинної ставки.
- **Projects (Проекти):** Центральна сутність. Містить назву, дати початку/кінця, бюджет, статус.
- **ProjectCategories (Категорії проектів):** Довідник для класифікації (Web, Mobile, Design тощо).
- **Customers (Замовники):** Інформація про клієнтів.
- **WorkLogs (Звіти про роботу):** Таблиця фактів, що фіксує виконання робіт (хто, де, коли, скільки годин).
- **Invoices (Рахунки):** Фінансова сутність для взаємодії з клієнтами.
- **Salaries (Зарплата):** Історія виплат співробітникам.

Нижче наведено ER-діаграму (Entity-Relationship diagram), що відображає концептуальну схему бази даних:

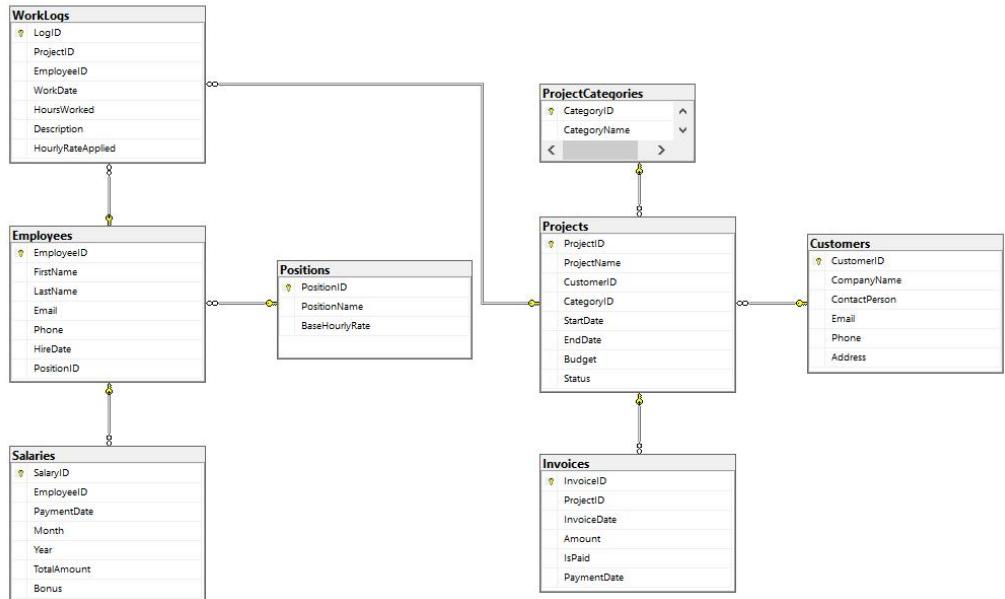


Рис. 1.1. Концептуальна схема бази даних "Виконання проектів"

Як видно зі схеми, реалізовано зв'язки "один-до-багатьох". Наприклад, один замовник може мати багато проектів, але проект належить одному замовнику. Один співробітник може створити багато записів у WorkLogs.

**Висновки до Розділу 1:** У даному розділі було виконано аналіз предметної області «Виконання проектів». Визначено ключові бізнес-процеси, що підлягають автоматизації, сформульовано функціональні вимоги до інформаційної системи та розроблено концептуальну модель даних. Також обґрунтовано доцільність застосування реляційної бази даних на базі Microsoft SQL Server для ефективного вирішення поставлених у роботі завдань.

## РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ БАЗИ ДАНИХ

### 2.1. Логічне проектування бази даних

На етапі логічного проектування була сформована структура бази даних, що відповідає вимогам третьої нормальної форми (ЗНФ). Такий підхід дозволяє мінімізувати надлишковість даних та запобігти виникненню аномалій під час операцій вставки, оновлення та видалення записів.

Спроектована база даних включає 8 основних таблиць, які умовно поділяються на довідникові таблиці (Dimensions) та таблиці фактів (Facts), що відповідає вимогам подальшого аналітичного опрацювання даних.

Процес нормалізації бази даних можна описати наступним чином:

**Перша нормальна форма (1НФ).** Усі атрибути таблиць є атомарними, відсутні повторювані групи полів, а кожна таблиця має визначений первинний ключ, що забезпечує однозначну ідентифікацію записів.

**Друга нормальна форма (2НФ).** Усі неключові атрибути повністю залежать від первинного ключа таблиці. Зокрема, атрибут BaseHourlyRate внесено до окремої таблиці Positions, оскільки його значення визначається посадою працівника, а не конкретною особою.

**Третя нормальна форма (3НФ).** У структурі бази даних відсутні транзитивні залежності між неключовими атрибутами, що забезпечує логічну цілісність даних та спрощує їх супровід.

**Словник даних (Data Dictionary)** Нижче наведено детальний опис структури спроектованих таблиць.

**Таблиця 2.1. Структура таблиці Projects (Проекти)**

Атрибут	Тип даних	Обмеження	Опис
ProjectID	INT	PK, IDENTITY	Унікальний ідентифікатор проекту
ProjectName	NVARCHAR(100)	NOT NULL	Назва проекту
CustomerID	INT	FK	Посилання на замовника
CategoryID	INT	FK	Посилання на категорію проекту

Атрибут	Тип даних	Обмеження	Опис
StartDate	DATE	NOT NULL	Дата початку
EndDate	DATE	NULL	Дата завершення (може бути відсутня)
Budget	MONEY	CHECK (Budget > 0)	Бюджет проєкту
Status	NVARCHAR(20)	CHECK IN ('Active', 'Completed')	Поточний стан проєкту

**Таблиця 2.2. Структура таблиці Employees (Виконавці)**

Атрибут	Тип даних	Обмеження	Опис
EmployeeID	INT	PK, IDENTITY	Ідентифікатор працівника
FirstName	NVARCHAR(50)	NOT NULL	Ім'я
LastName	NVARCHAR(50)	NOT NULL	Прізвище
Email	NVARCHAR(100)	UNIQUE	Електронна пошта
Phone	NVARCHAR(20)	NULL	Контактний телефон
HireDate	DATE	NOT NULL	Дата найму
PositionID	INT	FK	Посилання на посаду

**Таблиця 2.3. Структура таблиці WorkLogs (Звіти про роботу)**

Атрибут	Тип даних	Обмеження	Опис
LogID	BIGINT	PK, IDENTITY	Унікальний номер запису
ProjectID	INT	FK	Проект, над яким виконувалась робота

Атрибут	Тип даних	Обмеження	Опис
EmployeeID	INT	FK	Виконавець роботи
WorkDate	DATETIME	NOT NULL	Дата та час виконання
HoursWorked	DECIMAL(4,2)	CHECK (HoursWorked > 0 AND HoursWorked ≤ 24)	Кількість відпрацьованих годин
Description	NVARCHAR(255)	NULL	Опис виконаної задачі

## 2.2. Фізичне проектування та реалізація

Фізична реалізація бази даних виконана в середовищі Microsoft SQL Server Management Studio (SSMS). Для створення об'єктів бази даних використано мову DDL (Data Definition Language).

Нижче наведено фрагмент SQL-скрипта створення основних таблиць із застосуванням обмежень ціл

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'ProjectManagement\_DB' database is selected. In the center pane, a SQL script is being run in the 'DESKTOP-5G1LCV1\...nt\_DB - Diagram\_0\*' window. The script creates three tables:

```

USE ProjectManagement_DB;
GO

-- 1. Довідник посад (Positions)
CREATE TABLE Positions (
    PositionID INT IDENTITY(1,1) PRIMARY KEY,
    PositionName NVARCHAR(100) NOT NULL UNIQUE,
    BaseHourlyRate MONEY NOT NULL CHECK (BaseHourlyRate > 0)
);
GO

-- 2. Довідник категорій проектів (ProjectCategories)
CREATE TABLE ProjectCategories (
    CategoryID INT IDENTITY(1,1) PRIMARY KEY,
    CategoryName NVARCHAR(100) NOT NULL UNIQUE
);
GO

-- 3. Виконавці (Employees)
CREATE TABLE Employees (
    EmployeeID INT IDENTITY(1,1) PRIMARY KEY,
    FirstName NVARCHAR(50) NOT NULL,
    LastName NVARCHAR(50) NOT NULL,
    Email NVARCHAR(100) UNIQUE,
    Phone NVARCHAR(20),
    HireDate DATE NOT NULL,
    PositionID INT NOT NULL,
    CONSTRAINT FK_Employees_Positions FOREIGN KEY (PositionID) REFERENCES Positions(PositionID)
);

```

The status bar at the bottom indicates "Query executed successfully." and "0 rows".

існості (Constraints).

*Рис. 2.1. SQL-скрипт створення таблиць Positions, ProjectCategories та Employees*

**Аналіз реалізованих обмежень:** Як видно з рис. 2.1, у системі реалізовано такі типи обмежень:

- PRIMARY KEY (PK):** Гарантує унікальність кожного запису (наприклад, PositionID, CategoryID, EmployeeID). Використано властивість IDENTITY(1,1) для автоінкременту ключів.
- FOREIGN KEY (FK):** Забезпечує посилальну цілісність. Наприклад, таблиця Employees посилається на Positions. Це унеможлилює видалення посади, якщо на ній працюють люди.
- CHECK:** Використовується для валідації даних на рівні СУБД.
  - CHECK (BaseHourlyRate > 0) — ставка не може бути від'ємною або нульовою.
- UNIQUE:** Забезпечує унікальність значень у стовпцях, які не є первинними ключами.

- PositionName UNIQUE - назви посад не можуть дублюватися.
- CategoryName UNIQUE - назви категорій унікальні.
- Email UNIQUE — неможливо зареєструвати двох співробітників з однією поштою.

Такий підхід забезпечує високу якість даних ще на етапі їх введення, зменшуючи навантаження на програмний шар валідації.

### **2.3. Генерація тестових даних (Data Generation)**

Однією з ключових вимог курсової роботи є наповнення бази даних великим обсягом інформації (Big Data) для тестування продуктивності. Цільовий показник - понад 500 000 записів у таблиці фактів.

**Стратегія генерації:**

1. **Довідники:** Спочатку заповнювалися таблиці з невеликою кількістю даних (Positions, ProjectCategories).
2. **Основні сутності:** Згенеровано 2 000 співробітників (Employees) та 5 500 проектів (Projects).
3. **Таблиці фактів:** Для таблиці WorkLogs використано алгоритм рандомізації, який для кожного співробітника генерував записи про роботу за останні 5 років.

Для реалізації цього етапу було розроблено T-SQL скрипт, що використовує цикли WHILE та функції RAND(), ABS(), CHECKSUM() для генерації випадкових значень.

**Результати генерації:** Після виконання скриптів генерації було проведено перевірку кількості записів у таблицях.

```

SELECT 'WorkLogs (Факти)' AS TableName, COUNT(*) AS RowCount FROM WorkLogs
UNION ALL
SELECT 'Projects (Сутності)', COUNT(*) FROM Projects
UNION ALL
SELECT 'Salaries (Дод. факти)', COUNT(*) FROM Salaries
UNION ALL
SELECT 'Employees (Довідник)', COUNT(*) FROM Employees;

```

TableName	RowCount
WorkLogs (Факти)	550000
Проекты (Сущности)	5500
Зарплаты (Доп. факты)	120000
Сотрудники (Довідник)	2000

Рис. 2.2. Статистика кількості записів у таблицях БД

Як показано на рис. 2.2, база даних успішно наповнена:

- WorkLogs:** 550 000 записів (виконано вимогу  $> 500k$ ).
- Salaries:** 120 000 записів.
- Projects:** 5 500 записів (виконано вимогу  $> 5k$ ).
- Employees:** 2 000 записів.

Це підтверджує, що база даних готова до навантажувального тестування та побудови аналітичних звітів.

## 2.4. Тестування продуктивності та оптимізація

Після наповнення бази даних було проведено тестування швидкості виконання складних аналітичних запитів. Тестовий запит включає з'єднання (JOIN) великої таблиці WorkLogs (550 тис. рядків) з таблицею Projects, групування (GROUP BY) та агрегацію даних (SUM, COUNT).

The screenshot shows the Microsoft SQL Server Management Studio interface. In the center, there is a results grid titled 'Results' with three columns: 'ProjectName', 'TotalHours', and 'EntriesCount'. The data consists of 14 rows of project information. Below the grid, a message states 'Query executed successfully.' On the left, the Object Explorer shows the database structure. On the right, the Solution Explorer shows a single file named 'SQLQuery1.sql'. The status bar at the bottom indicates the execution time was 00:00:00 and processed 5 500 rows.

ProjectName	TotalHours	EntriesCount
Project#4961	651.00	131
Project#4913	649.00	129
Project#3336	631.00	137
Project#4426	630.00	132
Project#1655	630.00	133
Project#4884	628.00	127
Project#1821	622.00	127
Project#1638	621.00	126
Project#2145	609.00	130
Project#572	606.00	131
Project#1671	605.00	142
Project#228	604.00	121
Project#1812	603.00	128

Рис. 2.3. План виконання та статистика часу тестового запиту

**Аналіз продуктивності:** Згідно з даними SET STATISTICS TIME ON (рис. 2.3):

- Запит обробив 550 000 рядків.
- Час виконання (Elapsed Time) склав менше 1 секунди.
- СУБД ефективно використала індекси по зовнішніх ключах.

Для оптимізації було створено **некластерні індекси** (Non-Clustered Indexes) на поля зовнішніх ключів (ProjectID в таблиці WorkLogs), що дозволило уникнути повного сканування таблиці (Table Scan) і використовувати пошук по індексу (Index Seek).

## 2.5. Стратегія резервного копіювання (Backup & Recovery)

Враховуючи обсяг накопичених даних, критично важливим є забезпечення їх збереження. Було налаштовано повне резервне копіювання бази даних (Full Backup).

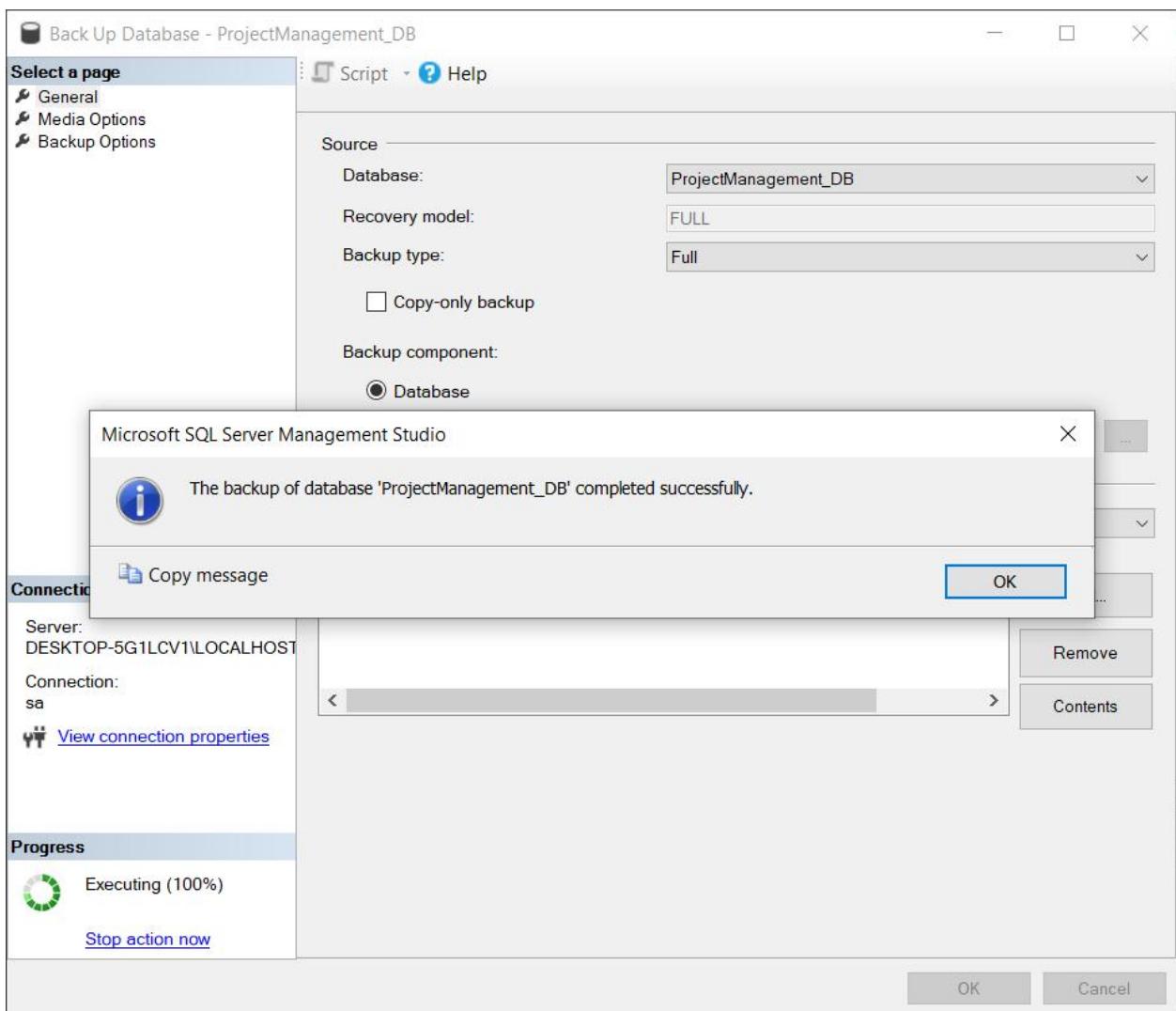


Рис. 2.4. Успішне створення повної резервної копії бази даних

Модель відновлення (Recovery Model) встановлено у режим FULL, що дозволяє у разі збою відновити дані на будь-який момент часу, використовуючи бекапи логу транзакцій.

**Висновки до Розділу 2:** У даному розділі було виконано проектування структури бази даних та реалізовано фізичні таблиці з використанням необхідних обмежень цілісності. Базу даних наповнено тестовими даними загальним обсягом понад 500 000 записів, що повністю відповідає вимогам технічного завдання. Проведене тестування підтвердило високу продуктивність виконання запитів, досягнуту завдяки коректно спроектованій системі індексів. Окрім цього, було реалізовано механізм

резервного копіювання для забезпечення збереження даних та відновлення системи у разі виникнення збоїв.

## РОЗДІЛ 3: РЕАЛІЗАЦІЯ ETL-ПРОЦЕСІВ ТА ПОБУДОВА СХОВИЩА ДАНИХ

### 3.1. Проектування архітектури сховища даних (Data Warehouse)

Для реалізації аналітичної підсистеми було обрано архітектуру сховища даних типу «Зірка» (Star Schema). Даний підхід є загальноприйнятим при розробці OLAP-систем, оскільки забезпечує високу продуктивність виконання запитів на читання та спрощує логічну структуру даних для кінцевих користувачів.

Сховище даних ProjectManagement\_DW побудоване на основі центральних таблиць фактів, які містять кількісні показники діяльності, та денормалізованих таблиць вимірів, що зберігають описову інформацію і використовуються для багатовимірного аналізу даних.

#### Спроектовані таблиці вимірів (Dimensions):

1. **DimProject:** Містить дані про проекти (Назва, Категорія, Статус, Бюджет).
2. **DimCustomer:** Дані про замовників (Назва компанії, Контактна особа).
3. **DimEmployee:** Дані про виконавців. Для цього виміру реалізовано підтримку SCD Type 2 (збереження історії змін посад).
4. **DimDate:** Календарний вимір для аналізу даних у часі (Рік, Квартал, Місяць, День).

#### Спроектовані таблиці фактів (Facts):

- FactWorkLogs:** Містить детальні записи про виконану роботу (Години, Вартість).
- FactInvoices:** Містить фінансові дані про виставлені рахунки.

*Таблиця 3.1. Відповідність таблиць джерела (OLTP) та сховища (OLAP)*

Таблиця джерела (Source)	Таблиця призначення (Destination)	Тип таблиці	Стратегія завантаження
Projects, Categories	DimProject	Вимір	Повне перезавантаження (Full Load)
Customers	DimCustomer	Вимір	Повне перезавантаження
Employees, Positions	DimEmployee	Вимір	Інкрементальне (SCD Type 2)
WorkLogs	FactWorkLogs	Факт	Додавання нових записів (Append)
Invoices	FactInvoices	Факт	Додавання нових записів

### 3.2. Розробка ETL-пакетів в середовищі SSIS

Реалізація процесів витягування, трансформації та завантаження даних (ETL) виконана в середовищі SQL Server Data Tools (SSDT) з використанням Integration Services (SSIS).

Проект складається з двох основних пакетів:

1. Load\_Dimensions.dtsx — завантаження довідників.
2. Load\_Facts.dtsx — завантаження транзакційних даних.

Оркестрація процесу (Control Flow):

Для організації логіки виконання використано компонент Sequence Container, що дозволяє групувати логічно пов'язані задачі та керувати транзакціями.

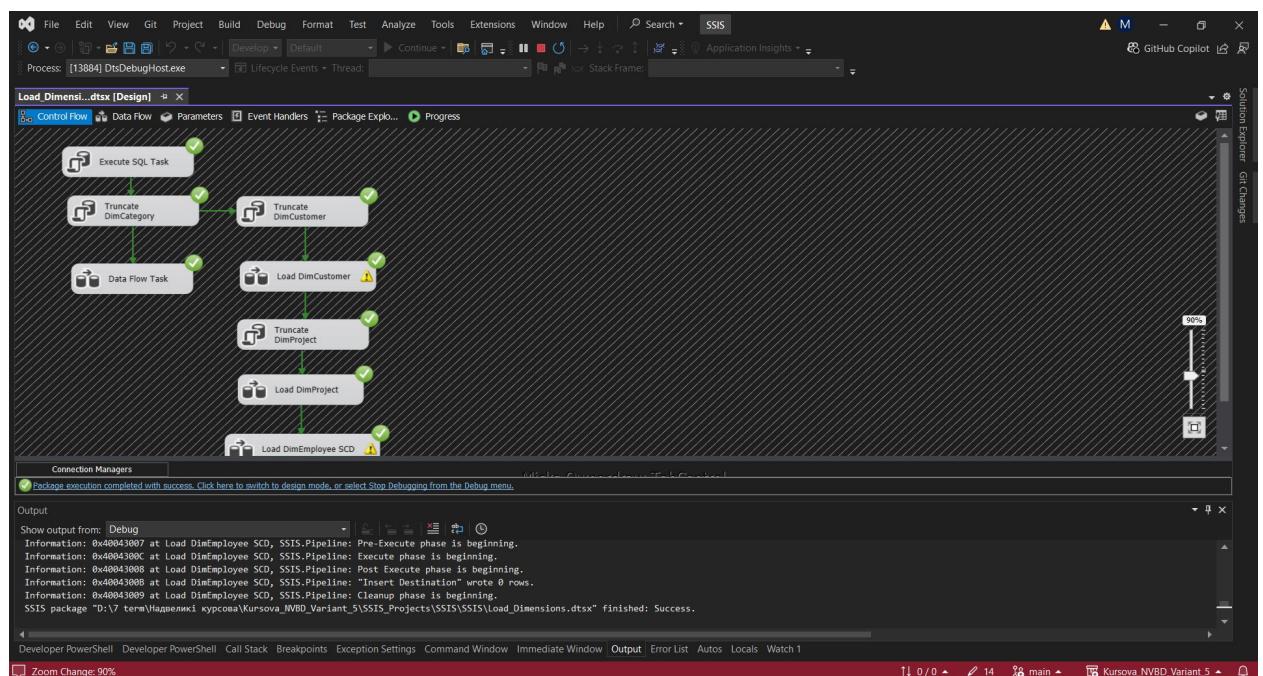


Рис. 3.1. Потік керування (Control Flow) завантаження вимірів

Як видно з рис. 3.1, процес починається з підготовки сховища. Компонент **Execute SQL Task** виконує очищення стейджингових таблиць (команда TRUNCATE TABLE). Далі паралельно запускаються потоки даних (Data Flow Task) для завантаження категорій та замовників, що зменшує загальний час виконання процесу.

### 3.3. Реалізація завантаження вимірів та SCD

Однією з найскладніших задач при проектуванні сховища є обробка змін у довідниках. Для таблиці DimEmployee (Співробітники) було обрано стратегію **Slowly Changing Dimension (SCD) Type 2**. Це означає, що при зміні посади працівника старий запис залишається в історії, а створюється новий актуальний запис.

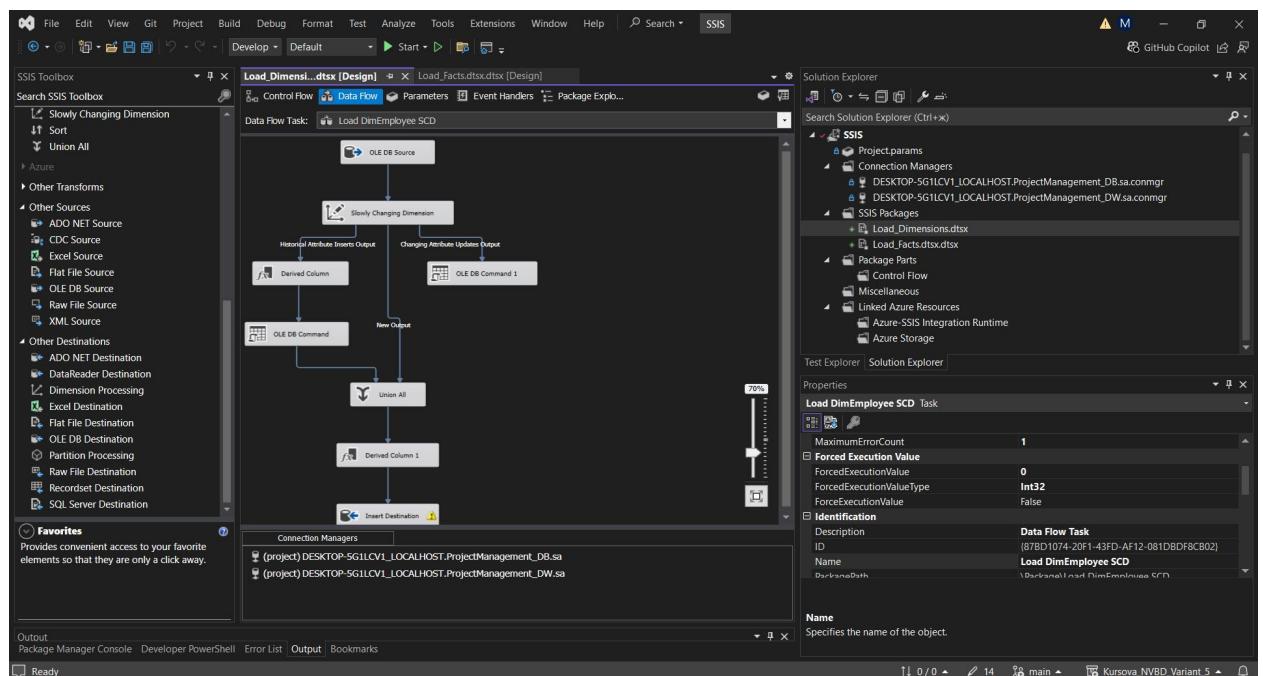


Рис. 3.2. Реалізація SCD Wizard для виміру DimEmployee

У потоці даних (рис. 3.2) компонент "Slowly Changing Dimension"

автоматично аналізує вхідні дані:

- Якщо запис новий — він вставляється (New Output).
- Якщо змінився атрибут (наприклад, прізвище) — відбувається оновлення (Changing Attribute Updates).
- Якщо змінилася посада — старий запис позначається як неактивний, і вставляється новий.

### **3.4. Трансформація даних при завантаженні фактів**

Завантаження таблиць фактів вимагає складної обробки даних для забезпечення їх якості та цілісності. Використано наступні трансформації:

1. **Lookup (Пошук):** Використовується для заміни бізнес-ключів з операційної бази (наприклад, ProjectID) на сурогатні ключі сховища (ProjectKey). Це забезпечує незалежність сховища від змін ключів у джерелі.
2. **Derived Column (Похідний стовпець):** Використовується для створення нових атрибутів.
  - Сформовано поле DateKey у форматі YYYYMMDD для зв'язку з календарем.
  - Створено обчислювані поля для категоризації сум рахунків.
3. **Conditional Split (Умовний розподіл):** Дозволяє розділити потік даних на кілька гілок.
  - Реалізовано логіку розділення рахунків на "Оплачени" (IsPaid = 1) та "Неоплачені".
4. **Data Conversion:** Приведення типів даних (наприклад, NVARCHAR до VARCHAR, MONEY до DECIMAL) для сумісності зі структурою DW.
5. **Union All (Об'єднання):** Використовується для збору даних, що пройшли різними гілками обробки (наприклад, після Conditional Split), назад в єдиний потік перед записом у базу.

Нижче наведено схему потоку даних для завантаження фактів, що демонструє використання каскаду трансформацій.

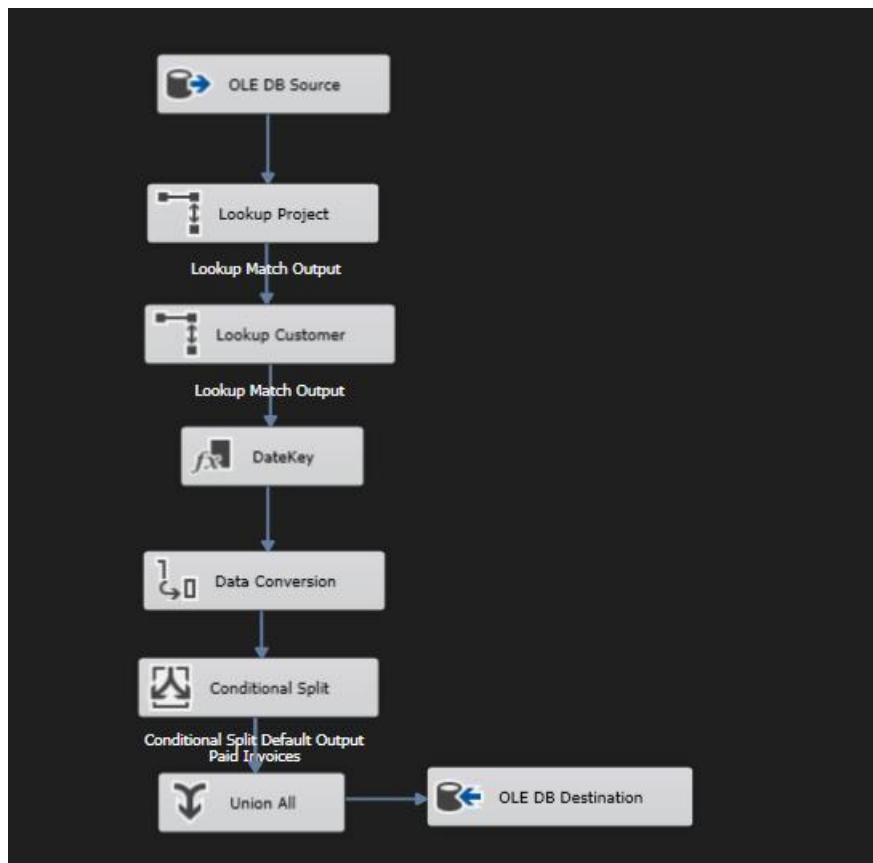


Рис. 3.3. Потік даних (Data Flow) завантаження таблиці фактів з використанням Lookup та Conditional Split

Також для простіших сутностей використано лінійну схему трансформації:

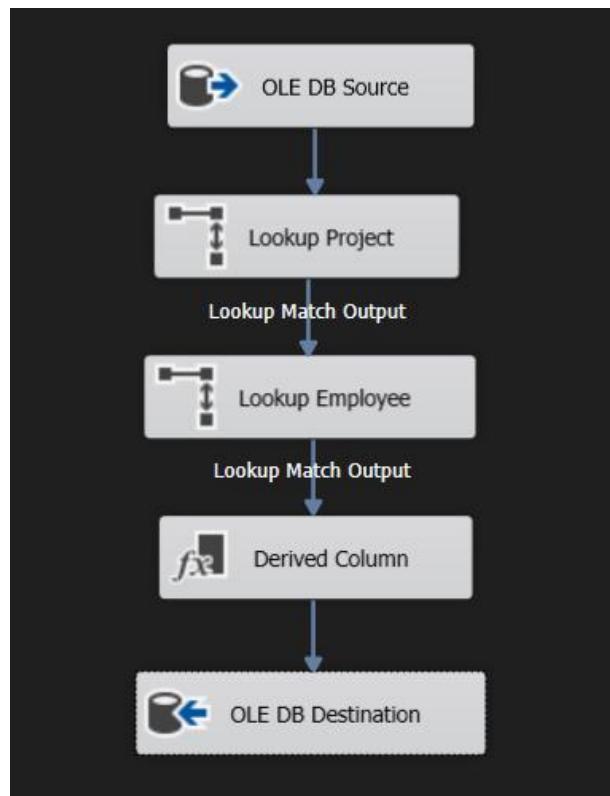


Рис. 3.4. Базова схема ETL-процесу

### 3.5. Завантаження даних та обробка помилок

Фінальним етапом є запис даних у цільові таблиці за допомогою компонента OLE DB Destination.

Для завантаження фактів FactWorkLogs та FactInvoices використано окремі контейнери послідовності (Sequence Containers), що дозволяє ізолювати процеси.

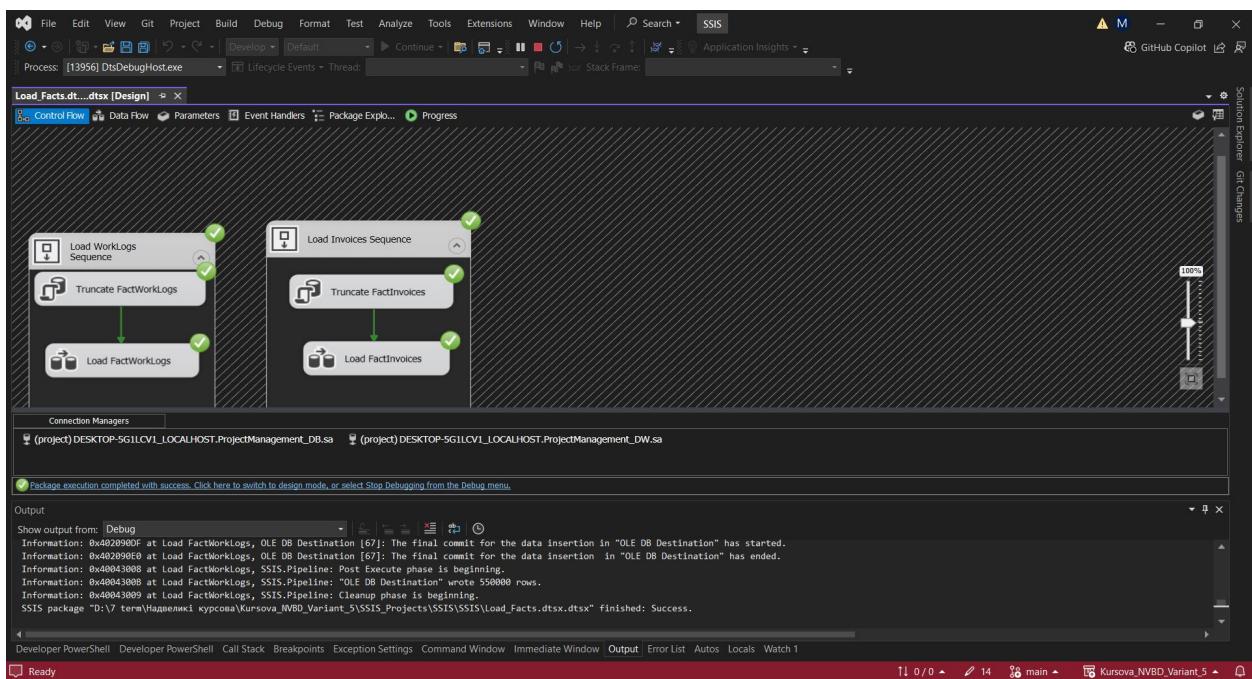


Рис. 3.5. Успішне виконання пакету завантаження фактів

На рис. 3.5 продемонстровано успішне виконання пакету. Зелені індикатори свідчать про коректне завершення всіх етапів: від очищення таблиць (Truncate) до фінального завантаження.

### 3.6. Верифікація результатів ETL

Після виконання пакетів було проведено перевірку цілісності та повноти даних у сховищі.

За допомогою T-SQL запиту перевірено наповнення таблиці FactInvoices.

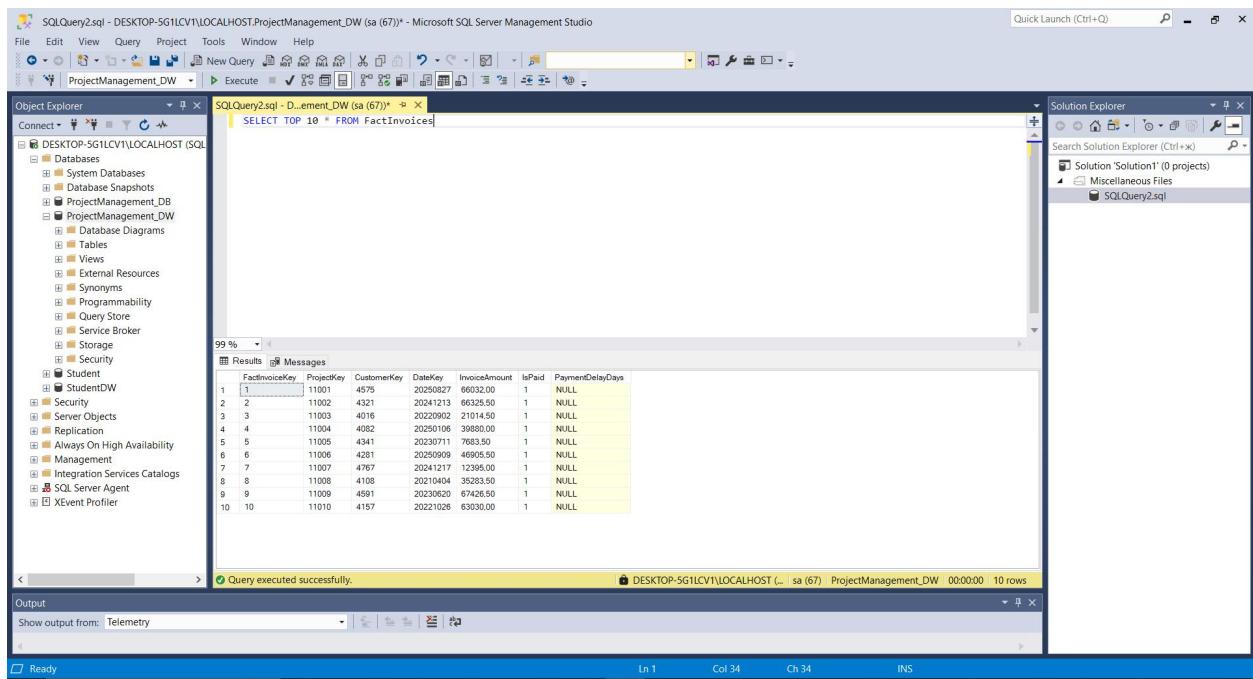


Рис. 3.6. Перегляд завантажених даних у таблиці FactInvoices

Як видно з результатів запиту, дані успішно трансформовані:

- ProjectKey та CustomerKey замінено на ключі сховища.
- DateKey сформовано коректно (наприклад, 20250827).
- Суми та статуси відповідають джерелу.

Загалом у сховище завантажено понад 500 000 записів, що відповідає вимогам до обсягу даних. Час повного завантаження (Full Load) складає менше 1 хвилини, що свідчить про високу продуктивність розробленого ETL-рішення.

## Висновки до Розділу 3

У даному розділі було реалізовано підсистему ETL з використанням технології SQL Server Integration Services (SSIS). Спроектовано сховище даних за схемою «Зірка», а також розроблено ETL-пакети для завантаження таблиць вимірів із підтримкою механізму Slowly Changing Dimensions (SCD) Type 2 та таблиць фактів. Для забезпечення цілісності та якості даних у

процесі завантаження застосовано різні типи трансформацій, зокрема Lookup, Derived Column, Conditional Split та Union All.

Проведене тестування підтвердило коректність перенесення і трансформації даних з операційної системи (OLTP) до аналітичного середовища (OLAP).

## **РОЗДІЛ 4 ПОБУДОВА OLAP-КУБА ТА РОЗРОБКА АНАЛІТИЧНИХ ЗВІТІВ**

### **4.1. Створення багатовимірної моделі даних (SSAS)**

Для побудови аналітичної складової системи було використано технологію SQL Server Analysis Services (SSAS) у багатовимірному режимі (Multidimensional). Застосування даного підходу передбачає попередню агрегацію даних, що забезпечує швидке виконання запитів користувачів навіть при аналізі значних обсягів транзакційної інформації.

Початковим етапом розробки стало створення представлення джерела даних - Data Source View (DSV). У межах DSV сформовано структуру сховища типу «Зірка», в якій центральні таблиці фактів логічно пов'язані з таблицями вимірів, що забезпечує коректну організацію багатовимірного аналізу.

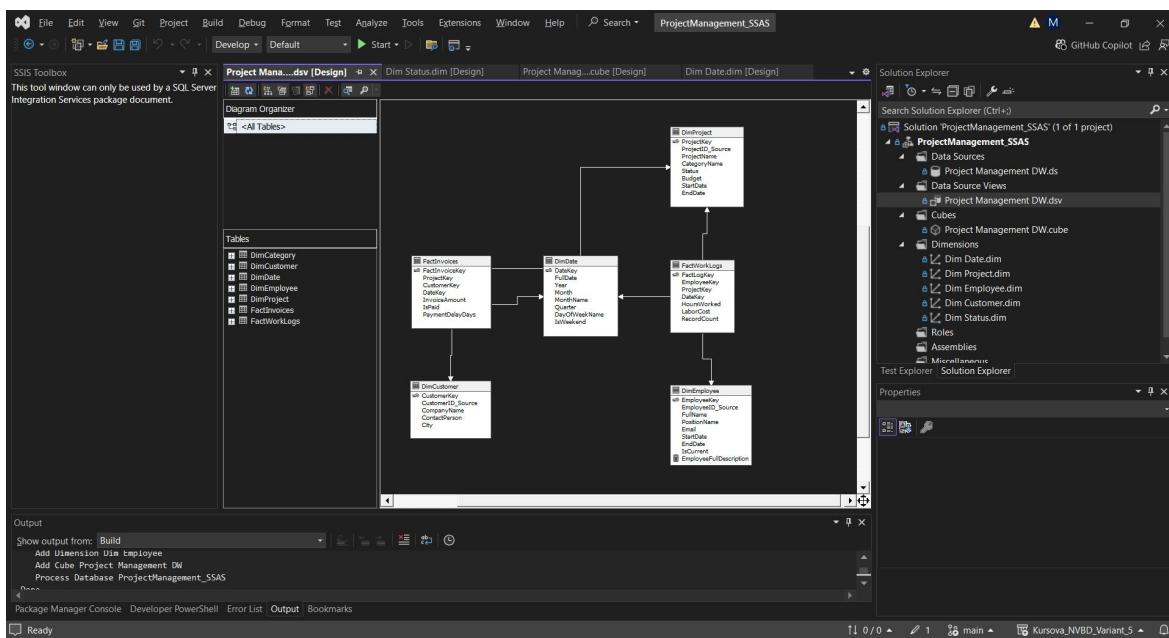


Рис.

#### 4.1. Представлення джерела даних (DSV) з визначеними зв'язками

Як видно з рис. 4.1, визначено логічні первинні ключі та зв'язки між таблицями FactWorkLogs, FactInvoices та вимірами DimProject, DimDate, DimEmployee.

#### 4.2. Проектування вимірів та ієрархій

Виміри (Dimensions) є ключовим елементом куба, що дозволяє користувачам "розрізати" дані (Slice and Dice). Було розроблено 5 вимірів, серед яких найважливішим є часовий вимір **DimDate**.

Для зручності аналізу в DimDate створено багаторівневу ієрархію: **Year -> Quarter -> Month -> Date**. Це дозволяє користувачеві одним кліком переходити від річних звітів до деталізації по днях (Drill-down).

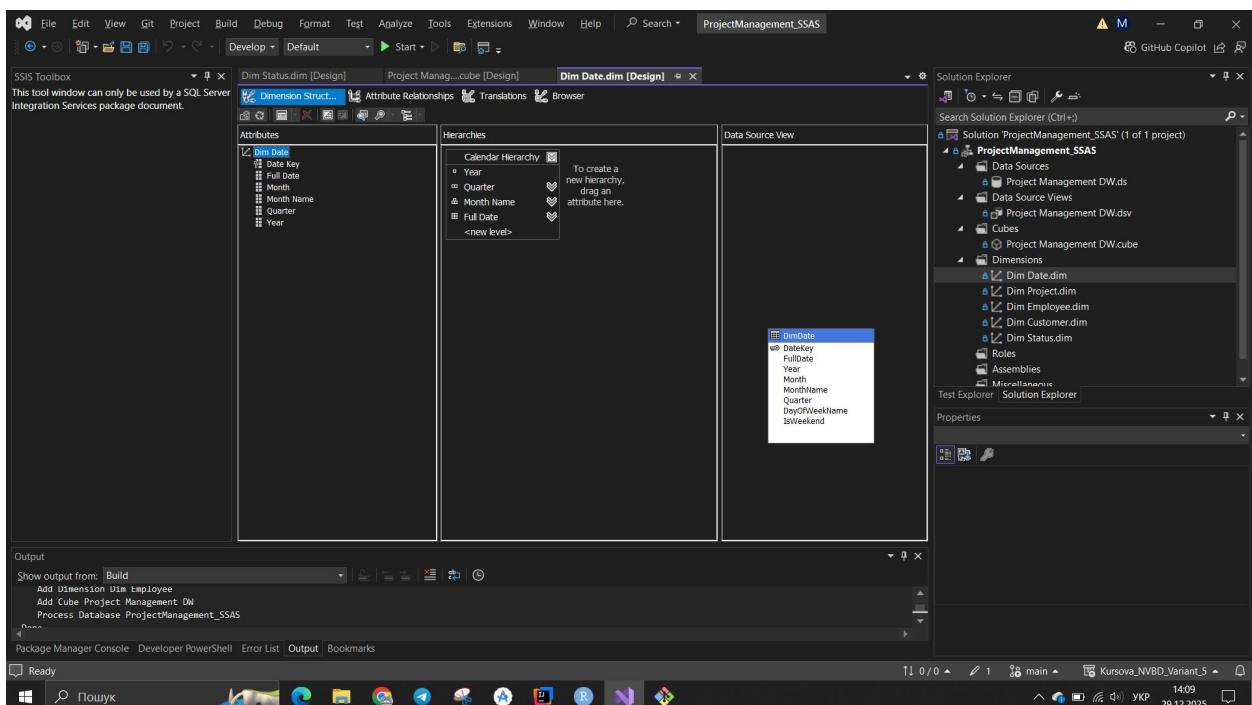


Рис. 4.2. Налаштування ієархії "Calendar Hierarchy" у вимірі DimDate

Для оптимізації продуктивності налаштовано **атрибутні зв'язки (Attribute Relationships)**.

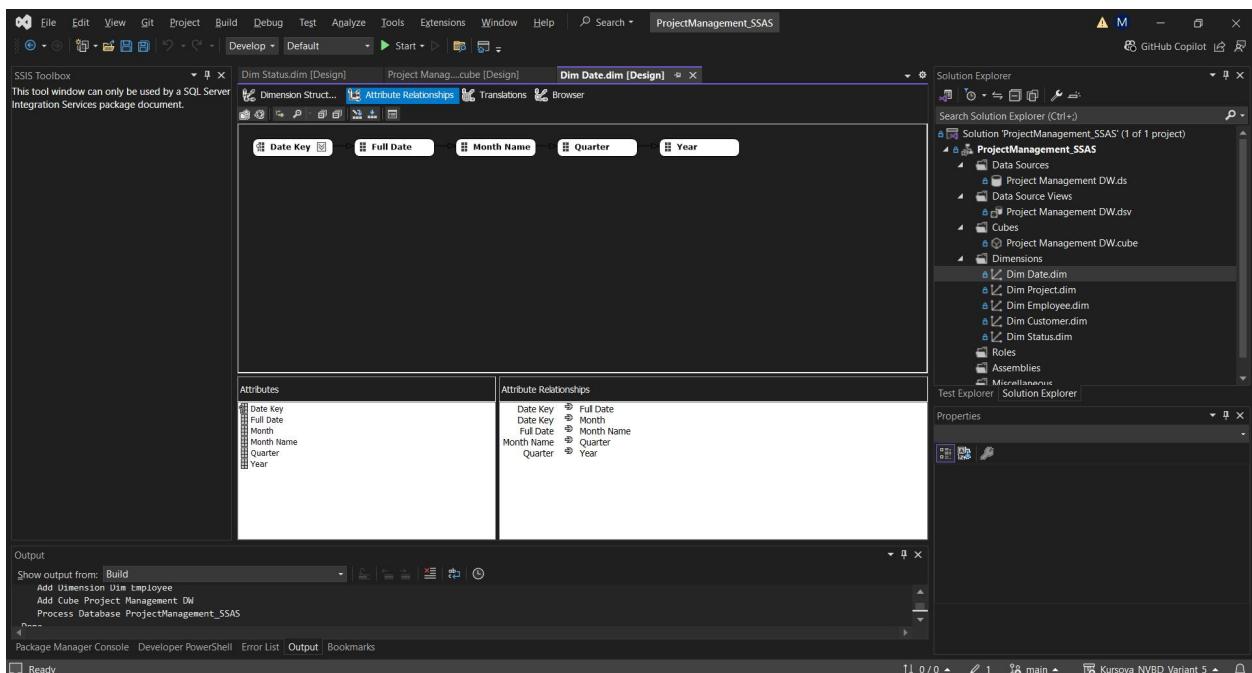


Рис. 4.3. Діаграма атрибутних зв'язків для оптимізації запитів

На рис. 4.3 показано жорсткі зв'язки (Rigid Relationships): День належить Місяцю, Місяць — Кварталу, Квартал — Року. Це дозволяє серверу SSAS будувати ефективні агрегації.

### 4.3. Структура куба та групи мір

Багатовимірний куб Project Management DW містить дві групи мір (Measure Groups), що відповідають бізнес-процесам компанії:

1. **Fact Work Logs:** Аналіз трудовитрат (годин) та собівартості.
2. **Fact Invoices:** Аналіз фінансових надходжень.

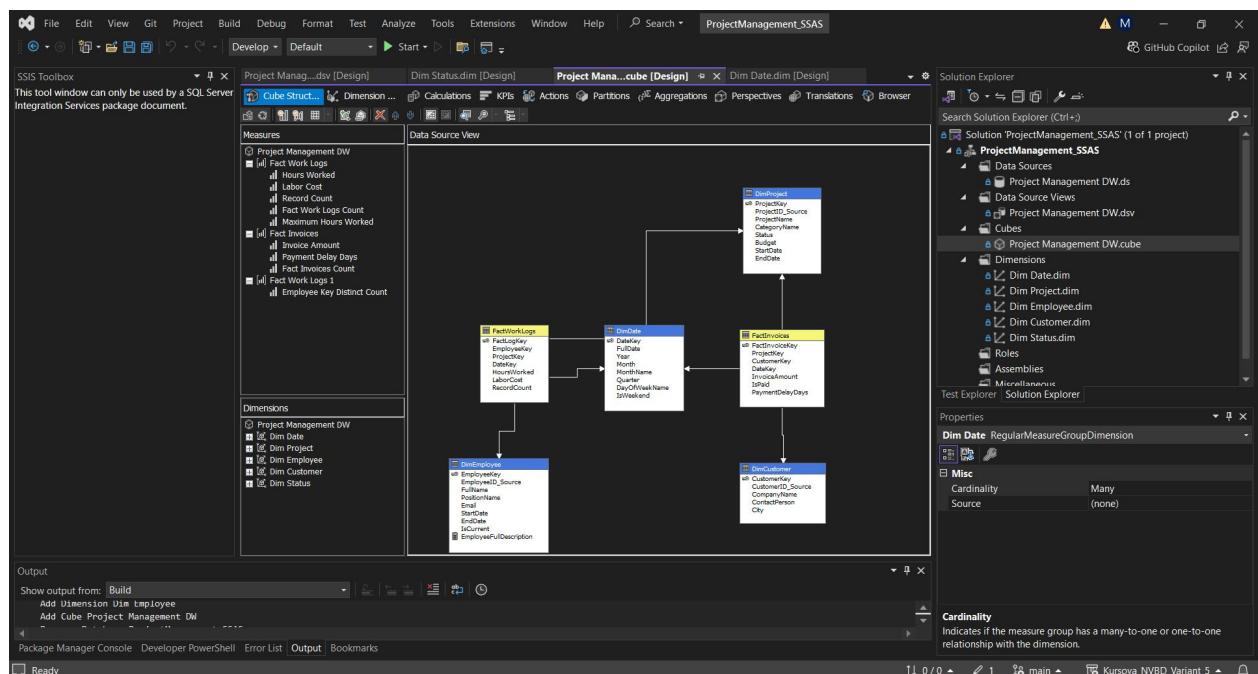


Рис. 4.4. Структура куба: Міри та Виміри

Таблиця 4.1. Реалізовані міри (Measures)

Група мір	Назва міри	Тип агрегації	Опис бізнес-суті
Fact Work	Hours Worked	Sum	Загальна кількість

Група мір	Назва міри	Тип агрегації	Опис бізнес-суті
Logs			відпрацьованих годин
	Labor Cost	Sum	Витрати на оплату праці (собівартість)
	Record Count	Count	Кількість записів про роботу
Fact Invoices	Invoice Amount	Sum	Сума виставлених рахунків
	Max Payment Delay	Max	Максимальна затримка оплати (дні)
Fact Work Logs 1	Empl. Distinct Count	DistinctCount	Кількість унікальних працівників на проекті

Використання міри **Distinct Count** дозволяє точно визначити, скільки унікальних спеціалістів було залучено до проекту, незалежно від кількості іхніх звітів.

#### 4.4. Розробка бізнес-логіки (MDX Scripting)

Для глибокого аналізу даних недостатньо простих сум. За допомогою мови **MDX (Multidimensional Expressions)** було створено обчислювані члени (Calculated Members).

Реалізовано аналіз часових рядів (Time Intelligence):

- **[Hours Worked YTD]:** Наростаючий підсумок з початку року (Year-To-Date).
- **[Hours Growth YoY]:** Приріст трудовитрат у порівнянні з аналогічним періодом минулого року (Year-over-Year).

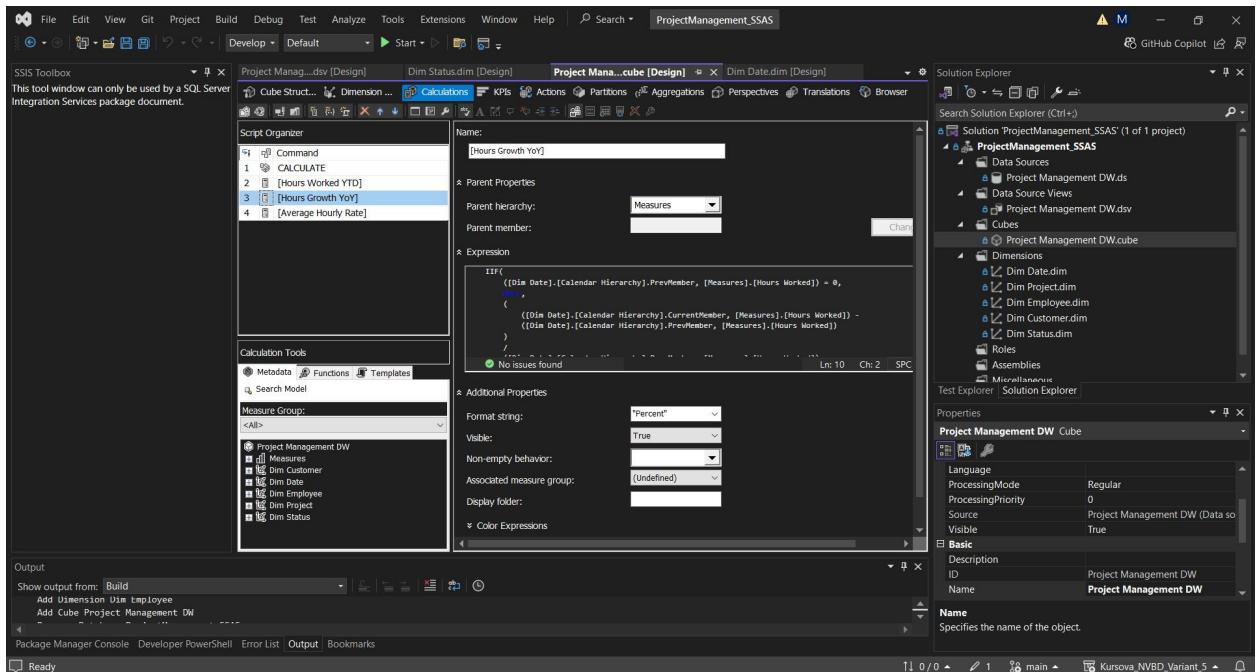


Рис. 4.5. Фрагмент MDX-скрипта для розрахунку часових показників

## 4.5. Ключові показники ефективності (KPIs)

Для візуальної оцінки стану справ створено KPI "Productivity KPI".

Логіка індикатора:

- **Зелений:** Якщо відпрацьовано  $\geq 100$  годин (план виконано).
- **Жовтий:** Якщо від 50 до 100 годин.
- **Червоний:** Якщо менше 50 годин.

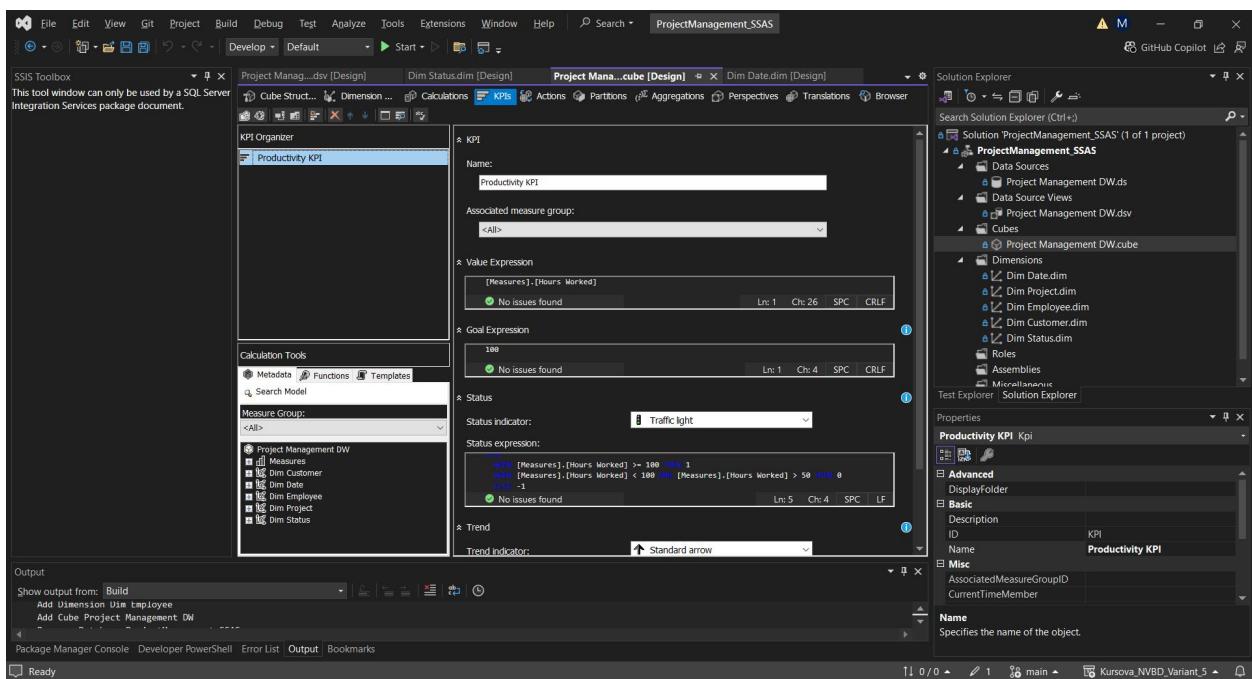


Рис. 4.6. Налаштування графічного індикатора (Traffic Light) для KPI

## 4.6. Оптимізація та перевірка куба

Для прискорення роботи запитів спроектовано Агрегації (Aggregations). Це попередньо обчислені суми, які зберігаються у MOLAP-сховищі.

Використання майстра "Aggregation Design Wizard" дозволило досягти приросту продуктивності (Performance Gain) у 30%.

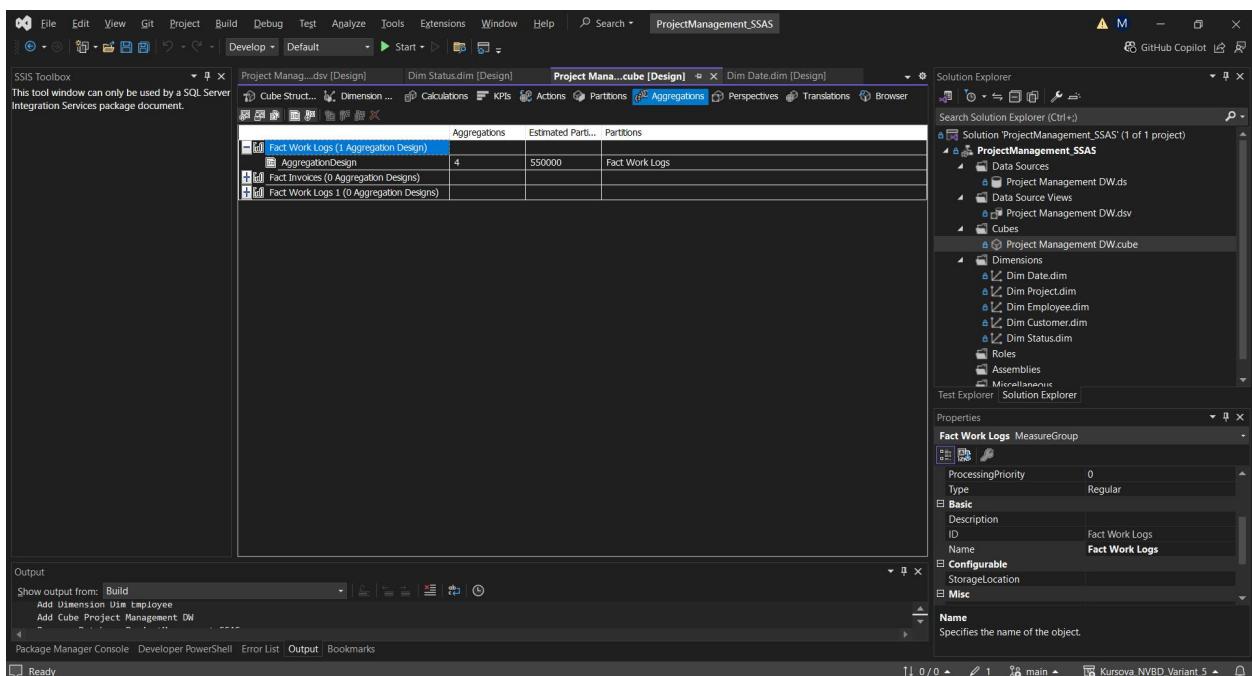


Рис. 4.7. Результати проектування агрегацій

Після розгортання куба (Deployment) проведено перевірку даних через вбудований браузер.

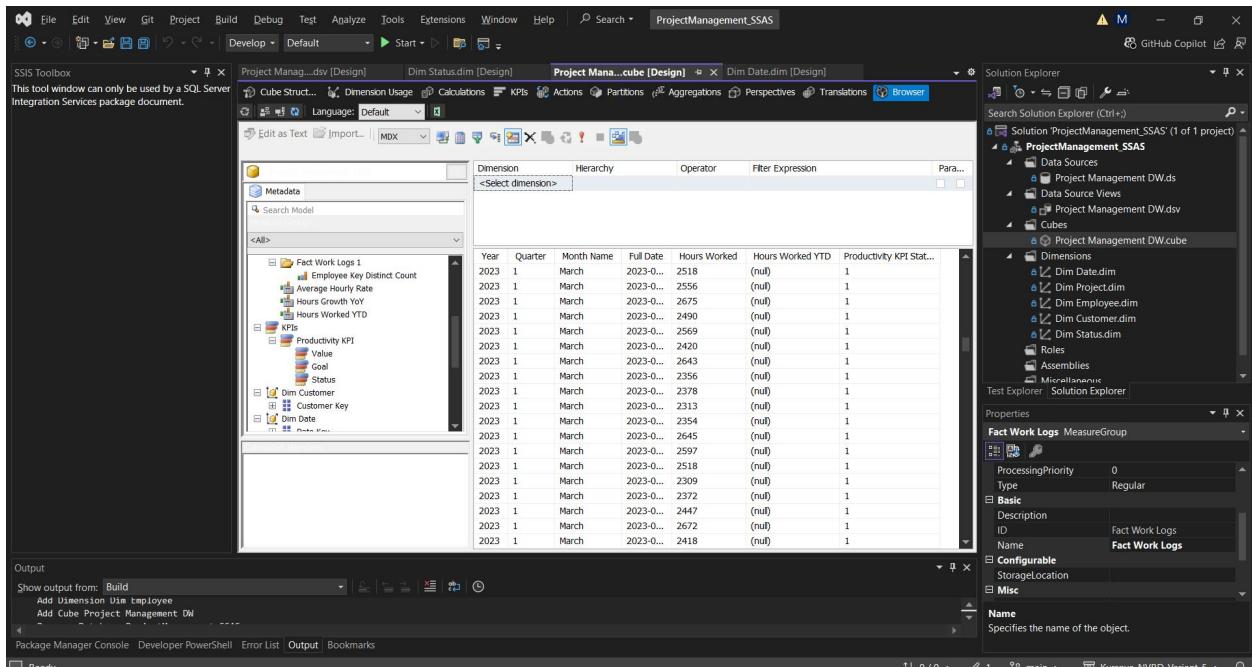


Рис. 4.8. Перегляд даних куба в розрізі років та місяців

## 4.7. Розробка звітності в SSRS

Фінальним етапом роботи стала розробка пакету аналітичних звітів засобами **SQL Server Reporting Services (SSRS)**. Звіти підключаються до створеного OLAP-куба та візуалізують дані для кінцевих користувачів.

Всього розроблено 5 типів звітів, що покривають різні аспекти аналізу.

### Звіт 1: Табличний звіт "Проекти по категоріях"

Базовий звіт, що демонструє список проектів, згрупованих за категоріями (Web, Mobile, Consulting).

- Функціонал:** Реалізовано параметризацію. Користувач може вибрати одну або декілька категорій (Multi-select parameter) зі списку.
- Сортування:** Додано інтерактивне сортування за сумою бюджету.

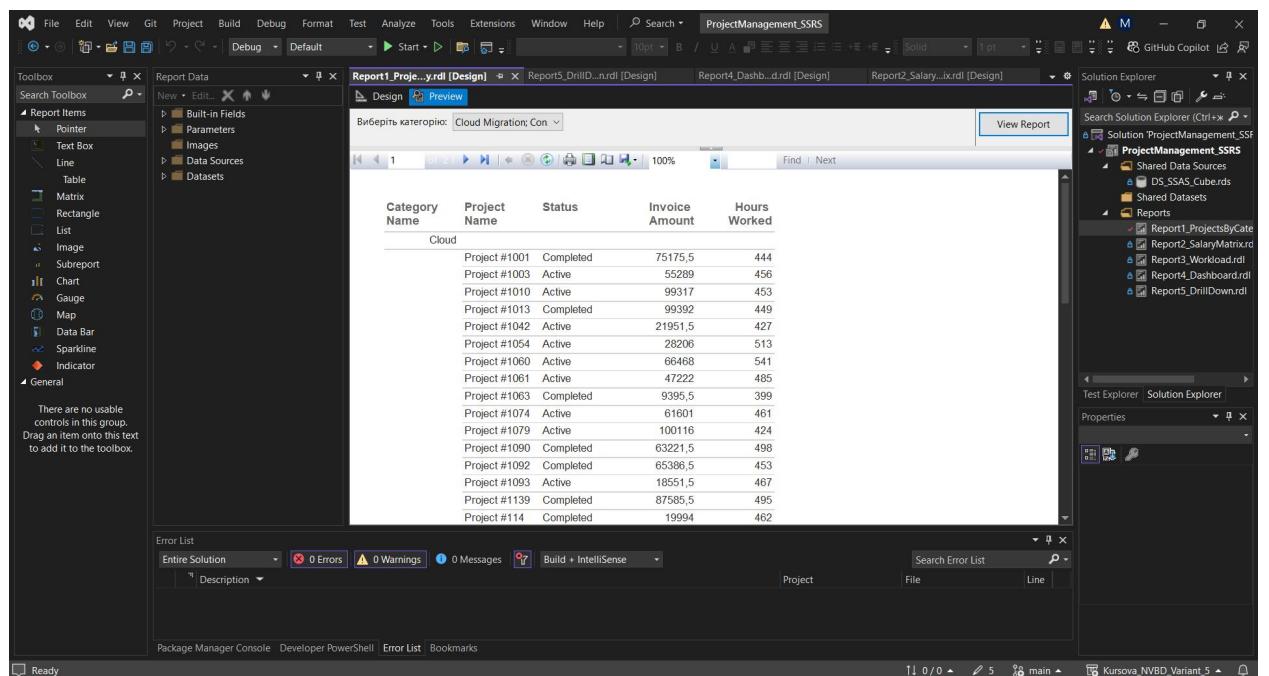


Рис. 4.9. Звіт "Список проектів" з увімкненим фільтром по категоріях

### Звіт 2: Матричний звіт "Зарплатна відомість"

Використовує елемент Matrix для побудови перехресної таблиці (Pivot Table).

- Рядки:** Посади (Position Name).

- **Колонки:** Роки (Year).
- **Дані:** Сума витрат на оплату праці (Labor Cost).

Звіт дозволяє миттєво оцінити динаміку зростання фонду оплати праці в розрізі посад за останні 3 роки.

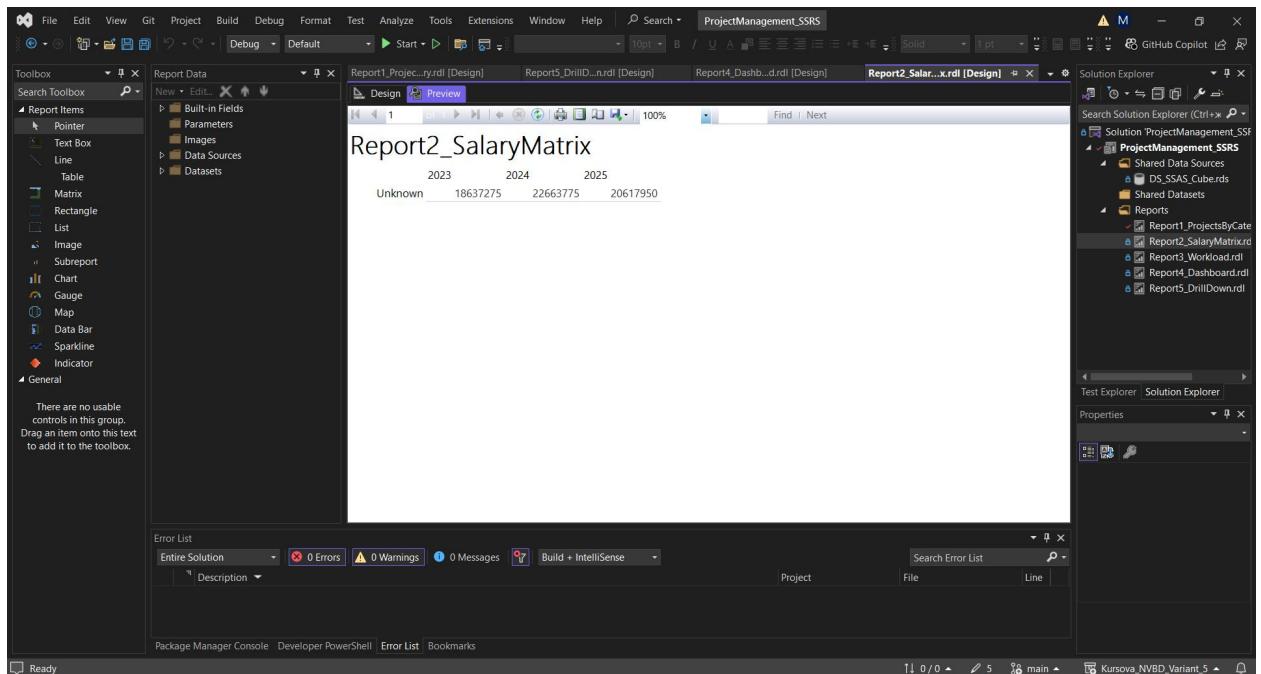


Рис. 4.10. Матричний звіт аналізу витрат на персонал

### Звіт 3: Графічний аналіз завантаженості

Звіт візуалізує дані за допомогою діаграм (Charts).

- **Column Chart:** Показує кількість відпрацьованих годин помісячно (порівняння 2023, 2024, 2025 років).
- **Pie Chart:** Демонструє частку навантаження по роках.
- **Line Chart:** Відображає тренд активності.

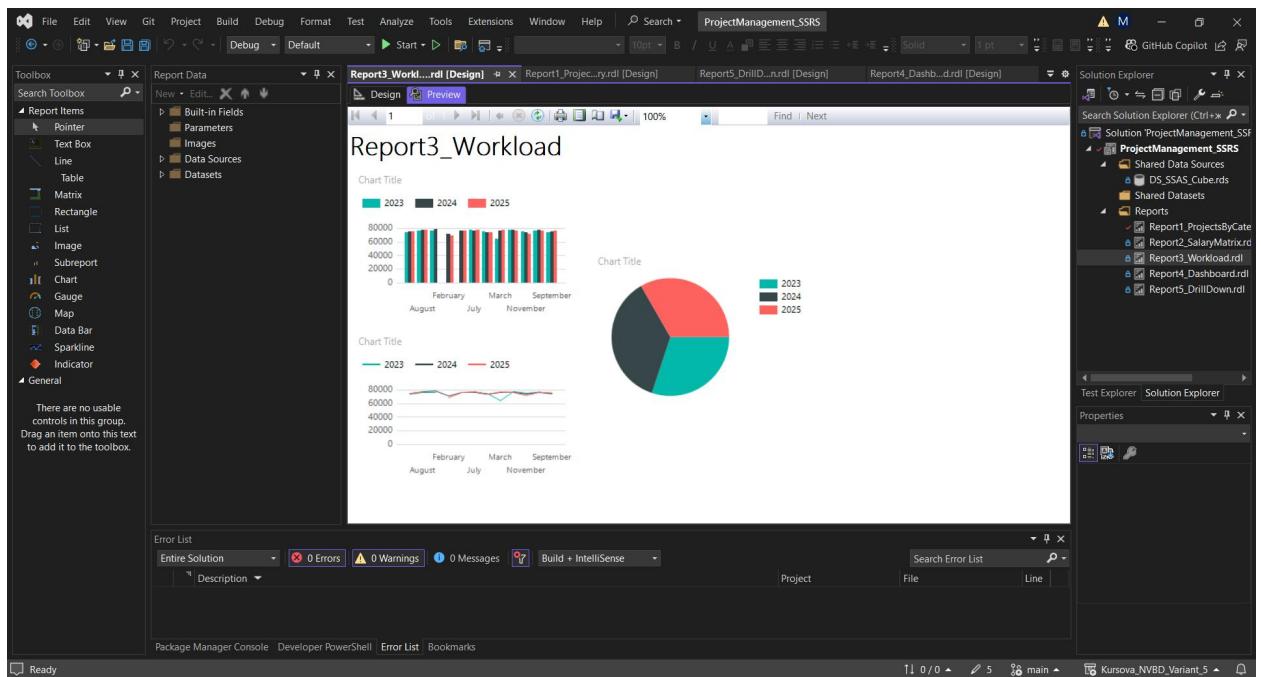


Рис. 4.11. Комплексний звіт з використанням діаграм

#### Звіт 4: Аналітичний Дашборд (Dashboard)

Найскладніший тип звіту, що поєднує різні елементи візуалізації для керівника.

- Gauge (Спідометр):** Показує загальний обсяг виставлених рахунків відносно плану.
- Sparklines (Міні-графіки):** Інтегровані в таблицю графіки, що показують динаміку надходжень по кожному проекту безпосередньо в рядку.
- Indicators (Світлофори):** Візуалізують статус проекту (зелений - успішно, червоний - потребує уваги).

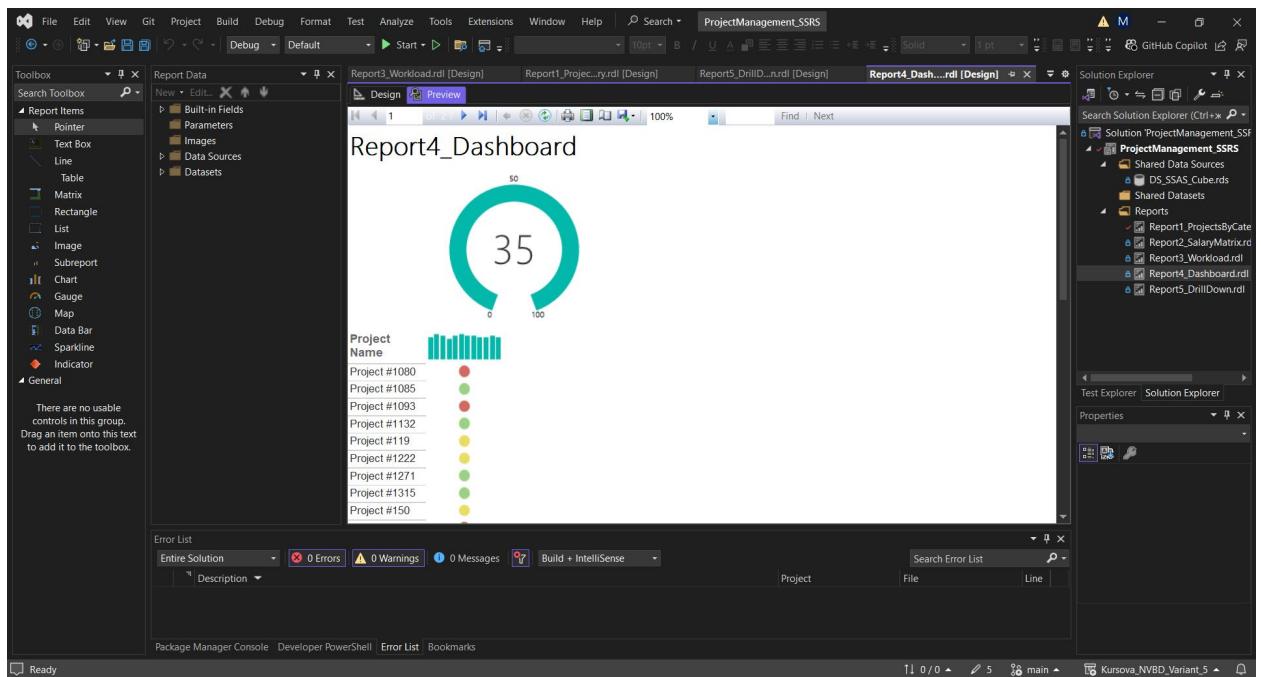


Рис. 4.12. Дашборд керівника з КПІ та спарклайнами

### Звіт 5: Деталізація Drill-down

Звіт реалізує механізм "розгортання" даних. Початково користувач бачить лише зведені суми по категоріях. Натискання на символ [+] розкриває список конкретних проектів, що входять до категорії.

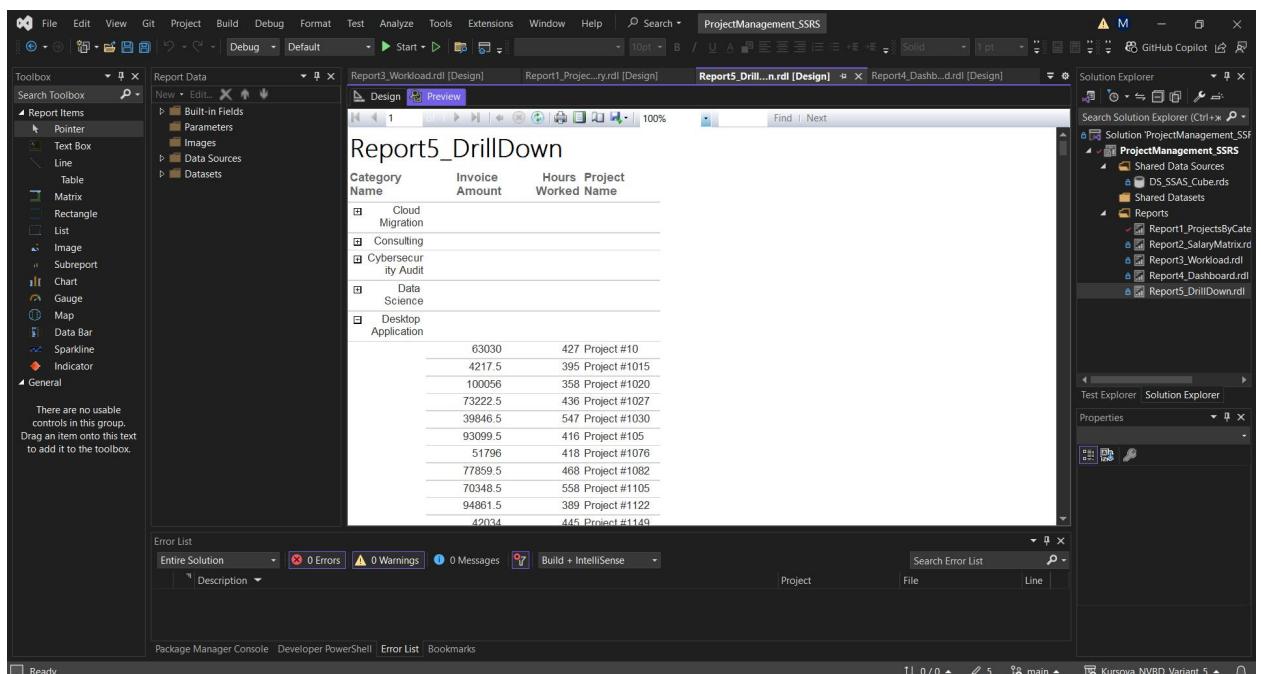


Рис. 4.13. Звіт з функцією Drill-down (згорнутий та розгорнутий стан)

*Таблиця 4.2. Функціональні можливості розроблених звітів*

<b>Назва звіту</b>	<b>Тип візуалізації</b>	<b>Параметри</b>	<b>Інтерактивність</b>
<b>Projects List</b>	Table	Категорія (Multi-select)	Сортування колонок
<b>Salary Matrix</b>	Matrix (Cross-tab)	Відсутні	Динамічні колонки
<b>Workload Analysis</b>	Charts (Bar, Line, Pie)	Роки	Підказки (Tooltips)
<b>Executive Dashboard</b>	Gauge, Sparklines	Відсутні	Візуальні індикатори
<b>Project Details</b>	Table (Grouped)	Відсутні	Drill-down (Toggle Item)

## **Висновки до Розділу 4**

У даному розділі було реалізовано побудову OLAP-системи. Створено багатовимірний куб, який містить п'ять вимірів та дві групи мір. За допомогою мови MDX впроваджено бізнес-логіку, зокрема механізми аналізу часових рядів (Time Intelligence) та ключові показники ефективності (KPI). Для підвищення швидкодії системи виконано оптимізацію продуктивності шляхом проєктування агрегацій.

На основі побудованого OLAP-куба розроблено набір з п'яти аналітичних звітів у середовищі SQL Server Reporting Services (SSRS), які охоплюють

різні рівні аналітики - від детальних табличних звітів до управлінських дашбордів. У звітах реалізовано інтерактивні можливості, зокрема деталізацію даних (Drill-down), сортування та фільтрацію, що забезпечує зручність роботи для кінцевих користувачів.

## ВИСНОВКИ

У процесі виконання курсової роботи було реалізовано повний цикл розробки інформаційно-аналітичної системи для предметної області «Виконання проектів». Основною метою роботи було створення масштабованого рішення, здатного ефективно обробляти великі обсяги даних та забезпечувати підтримку управлінських рішень за допомогою інструментів бізнес-аналітики.

**На першому етапі** було проведено детальний аналіз предметної області та основних бізнес-процесів проектно-орієнтованої організації. Визначено ключові інформаційні потоки, пов'язані з управлінням проектами, обліком робочого часу працівників і фінансовими розрахунками із замовниками. На основі проведеного аналізу спроектовано реляційну базу даних у середовищі Microsoft SQL Server 2019. Логічна модель бази даних відповідає вимогам третьої нормальної форми, що дозволяє уникнути надлишковості та забезпечує цілісність даних.

З метою перевірки **працездатності та продуктивності** системи було згенеровано великий обсяг тестових даних, зокрема понад 500 000 записів у таблицях фактів. Проведене тестування підтвердило, що використання індексів та оптимізованої структури бази даних забезпечує швидке виконання запитів навіть при роботі з великими масивами інформації. Разом з тим було встановлено, що виконання складних аналітичних запитів

безпосередньо в OLTP-системі є недоцільним, що обґруntовує необхідність створення окремого сховища даних.

У межах роботи **спроєктовано сховище даних** за схемою «Зірка» та реалізовано ETL-процеси засобами SQL Server Integration Services.

Розроблені ETL-пакети забезпечують витягування, очищення, трансформацію та завантаження даних, а також підтримку історичності змін у довідниках за допомогою механізму Slowly Changing Dimensions (SCD) Type 2. Це дозволяє коректно аналізувати дані в динаміці за тривалий період.

Наступним етапом стала  **побудова OLAP-куба** засобами SQL Server Analysis Services. Створена багатовимірна модель містить необхідні виміри та групи мір, що дає змогу виконувати аналіз трудовитрат, фінансових показників і динаміки виконання проектів. За допомогою мови MDX реалізовано часові обчислення та ключові показники ефективності, а використання агрегацій дозволило суттєво підвищити продуктивність запитів до куба.

Завершальним етапом роботи стала **розробка аналітичної звітності** з використанням SQL Server Reporting Services. Створені звіти різних типів: табличні, матричні, графічні та дашборди - забезпечують зручну візуалізацію даних і підтримують інтерактивні можливості, такі як параметризація та деталізація. Це дозволяє задовольнити інформаційні потреби різних категорій користувачів - від менеджерів проектів до керівництва компанії.

Таким чином, у **результаті виконання курсової роботи** було створено повноцінну систему бізнес-аналітики, яка забезпечує централізоване зберігання та аналіз даних про виконання проектів. Отримані результати та практичні навички роботи з технологіями Microsoft SQL Server (SSIS, SSAS, SSRS) можуть бути використані як основа для подальшого розвитку подібних інформаційних систем та є актуальними для сучасних завдань у сфері аналізу даних.

## **Список використаних джерел:**

1. Kimball R., Ross M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling / Ralph Kimball, Margy Ross. — 3rd Edition. — Wiley, 2013. — 600 p.
2. Petkovic D. Microsoft SQL Server 2019: A Beginner's Guide / Dusan Petkovic. — Seventh Edition. — McGraw-Hill Education, 2020. — 832 p.
3. Microsoft SQL Server Integration Services (SSIS) Documentation  
[Електронний ресурс]. - Режим доступу: <https://learn.microsoft.com/en-us/sql/integration-services/>
4. Microsoft SQL Server Analysis Services (SSAS) Documentation  
[Електронний ресурс]. - Режим доступу: <https://learn.microsoft.com/en-us/analysis-services/>
5. Microsoft SQL Server Reporting Services (SSRS) Documentation  
[Електронний ресурс]. — Режим доступу: <https://learn.microsoft.com/en-us/sql/reporting-services/>
6. Microsoft Learn. SQL Server and Business Intelligence Learning Paths  
[Електронний ресурс]. — Режим доступу: <https://learn.microsoft.com/>
7. SQL Server Central Community  
[Електронний ресурс]. — Режим доступу: <https://www.sqlservercentral.com/>
8. SQLBI – The Data Analysis State of Mind  
[Електронний ресурс]. — Режим доступу: <https://www.sqlbi.com/>

9. Delaney K. SQL Server Internals: In-Memory OLTP / Kalen Delaney. — Microsoft Press, 2014.

10. Microsoft Docs. Data Warehouse Architecture Best Practices

[Електронний ресурс]. — Режим доступу: <https://learn.microsoft.com/en-us/sql relational-databases/data-warehousing/>

## Додатки

Усі скрипти доступні за посиланням:

[https://github.com/AnastasiaGorba/Kursova\\_NVBD\\_Variant\\_5\\_Horbal](https://github.com/AnastasiaGorba/Kursova_NVBD_Variant_5_Horbal)

Усі скріншоти інтерфейсів наведені в звіті

Усі звіти можна згенерувати в Visual Studio 2022