

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут прикладної математики та фундаментальних наук

Кафедра прикладної математики

Курсова робота

з курсу «Надвеликі бази даних»

на тему «Розробка та аналіз надвеликої бази даних
з використанням технологій Microsoft SQL Server»

Виконала

студентка групи ПМ-42

Гусак А.І.

(підпис)

Перевірив:

Любінський Б.Б.

(дата)

(підпис викладача)

Анотація

Метою курсової роботи є розробка повнофункціональної системи керування надвеликою базою даних для обраної предметної області з реалізацією ETL-процесів, побудовою багатовимірного куба та створенням аналітичних звітів з використанням технологій Microsoft SQL Server (SSIS, SSAS, SSRS). У ході роботи проведено аналіз предметної області та спроектовано реляційну базу даних, нормалізовану до 3НФ. Для перевірки працездатності системи на великих навантаженнях згенеровано тестовий набір даних обсягом понад 500 000 записів за допомогою інструменту Redgate SQL Data Generator. Реалізовано підсистему ETL на базі SQL Server Integration Services (SSIS), що забезпечує автоматизоване збирання, очищення та завантаження даних у спроектоване сховище даних (Data Warehouse) архітектури «Зірка». Для задач бізнес-аналітики побудовано багатовимірний OLAP-куб засобами SQL Server Analysis Services (SSAS), що дозволяє аналізувати дані у розрізі часу, спікерів та конференцій. Розроблено систему інтерактивної звітності на платформі SQL Server Reporting Services (SSRS), яка включає параметризовані таблиці, матриці та аналітичні дашборди.

Annotation

The purpose of the course work is to develop a fully functional super-large database management system for a selected subject area with the implementation of ETL processes, construction of a multidimensional cube and creation of analytical reports using Microsoft SQL Server technologies (SSIS, SSAS, SSRS). During the work, an analysis of the subject area was conducted and a relational database normalized to 3NF was designed. To check the system's performance under heavy loads, a test data set of over 500,000 records was generated using the Redgate SQL Data Generator tool. An ETL subsystem based on SQL Server Integration Services (SSIS) was implemented, which provides automated collection, cleaning and loading of data into the designed data warehouse (Data Warehouse) of the "Star" architecture. For business analytics tasks, a multidimensional OLAP cube was built using SQL

Server Analysis Services (SSAS), which allows analyzing data by time, speakers and conferences. An interactive reporting system has been developed on the SQL Server Reporting Services (SSRS) platform, which includes parameterized tables, matrices, and analytical dashboards.

Зміст

1. Вступ.....	5
2. Розділ 1. Аналіз предметної області.....	9
2. Розділ 2. Проектування бази даних	14
2. Розділ 3. Реалізація ETL-процесів	18
2. Розділ 4. Побудова OLAP-куба	23
2. Розділ 5. Аналітичні звіти (SQL Server Reporting Services).....	31
3. Висновки	38
4. Список використаних джерел	41
5. Додаток А.....	42

Вступ

Актуальність теми. Сучасний етап розвитку інформаційного суспільства характеризується стрімким зростанням обсягів даних у всіх сферах людської діяльності. Наукова та освітня сфери не є виключенням. Організація масштабних заходів, таких як міжнародні наукові конференції, симпозиуми та форуми, супроводжується накопиченням та обробкою значних масивів різномірної інформації: даних про учасників, спікерів, тематичні секції, розклад виступів, використання матеріально-технічних ресурсів та аудиторного фонду.

В умовах зростання кількості учасників та ускладнення логістики заходів, традиційні методи обліку (використання електронних таблиць або розрізнених файлових систем) стають неефективними. Вони не забезпечують цілісності даних, не дозволяють оперативно отримувати аналітичну звітність та призводять до помилок при плануванні ресурсів. Крім того, з часом накопичується історична інформація, яка має значну цінність для виявлення тенденцій, аналізу популярності наукових напрямків та оцінки ефективності роботи організаторів.

Вирішенням цієї проблеми є перехід від простих облікових систем до використання технологій надвеликих баз даних (NVBD) та систем бізнес-аналітики (Business Intelligence). Розробка спеціалізованого сховища даних (Data Warehouse) дозволяє консолідувати інформацію, забезпечити її високу якість та надати інструменти для багатовимірного аналізу. Використання сучасного стеку технологій Microsoft SQL Server (SSIS, SSAS, SSRS) надає можливість побудувати масштабовану систему, здатну обробляти сотні тисяч записів, автоматизувати процеси завантаження даних та формувати аналітичні звіти в реальному часі.

Таким чином, розробка інформаційно-аналітичної системи для предметної області «Конференції» є актуальним завданням, що дозволяє підвищити якість управління науковими заходами та оптимізувати використання ресурсів.

Мета і завдання дослідження. Метою курсової роботи є проектування та розробка повнофункціональної системи керування надвеликою базою даних для предметної області «Конференції» з реалізацією повного циклу обробки даних: від проектування транзакційної бази даних до побудови багатовимірного OLAP-куба та створення аналітичної звітності.

Для досягнення поставленої мети необхідно вирішити наступні **завдання**:

1. Провести системний аналіз предметної області, визначити основні бізнес-процеси, сутності та інформаційні потоки, що виникають під час організації конференцій.
2. Розробити концептуальну, логічну та фізичну моделі реляційної бази даних (OLTP), забезпечивши нормалізацію до третьої нормальної форми (3НФ) для підтримки цілісності даних.
3. Здійснити генерацію великого масиву тестових даних (понад 500 000 записів), що імітують реальну діяльність організаторів конференцій за період у 10 років, для перевірки продуктивності системи.
4. Спроекувати структуру сховища даних (Data Warehouse) за схемою «Зірка» та реалізувати ETL-процеси (Extract, Transform, Load) для автоматизованого перенесення та очищення даних з операційної бази у сховище за допомогою SQL Server Integration Services (SSIS).
5. Побудувати багатовимірний OLAP-куб засобами SQL Server Analysis Services (SSAS), визначити виміри, ієрархії та ключові показники ефективності (KPI) для аналізу діяльності.
6. Розробити набір інтерактивних аналітичних звітів з використанням SQL Server Reporting Services (SSRS), що забезпечують візуалізацію даних та підтримку прийняття управлінських рішень.

Об'єктом дослідження є процеси інформаційної підтримки та управління даними в організаціях, що займаються проведенням масових наукових заходів.

Предметом дослідження є методи та засоби проектування реляційних баз даних, сховищ даних, процесів інтеграції даних (ETL) та систем бізнес-аналітики на платформі Microsoft SQL Server.

Методи дослідження. У роботі використано методи системного аналізу для декомпозиції предметної області; методи теорії реляційних баз даних для нормалізації та проектування схеми даних; методологію багатовимірного моделювання Ральфа Кімбалла для побудови сховища даних; методи процедурного програмування (T-SQL) для реалізації бізнес-логіки на стороні сервера.

Наукова новизна та практичне значення. Практичне значення одержаних результатів полягає у створенні працездатної інформаційної системи, яка дозволяє автоматизувати рутинні операції з обліку спікерів та виступів, а також надає потужний інструментарій для аналітики. Розроблена система дозволяє:

- Аналізувати завантаженість аудиторій та обладнання для оптимізації витрат.
- Відстежувати популярність тематичних секцій та динаміку активності спікерів за роками.
- Формувати складні аналітичні звіти за лічені секунди завдяки використанню технологій попередньої агрегації даних (OLAP).

Отримані результати можуть бути використані як основа для впровадження реальної системи управління конференціями в університетах або івент-агентствах.

Інструментальні засоби. Для реалізації курсової роботи використано програмний комплекс Microsoft SQL Server, що включає:

- SQL Server Database Engine — для зберігання та обробки даних.
- SQL Server Integration Services (SSIS) — для розробки ETL-пакетів.
- SQL Server Analysis Services (SSAS) — для побудови OLAP-кубів.
- SQL Server Reporting Services (SSRS) — для створення звітів.
- Середовища розробки: SQL Server Management Studio (SSMS) та Visual Studio (SSDT).
- Redgate SQL Data Generator — для генерації синтетичних даних.

Розділ 1. Аналіз предметної області

1.1. Загальна характеристика об'єкта дослідження

Предметна область "Конференції" являє собою складну організаційну структуру, діяльність якої спрямована на обмін знаннями та нетворкінг. Основним об'єктом є "Конференція" — подія, яка має чітко визначені часові рамки, місце проведення, тематику та склад учасників. Специфіка варіанту №3 передбачає необхідність підтримки історичних даних за 10 років, що включає мінімум 1 000 конференцій та 100 000 виступів. Такий масштаб вимагає врахування не лише поточних операційних потреб, але й довгострокових трендів.

Предметною областю даної курсової роботи є інформаційні процеси, що супроводжують організацію та проведення масштабних наукових конференцій. У сучасному науковому середовищі конференції є основним інструментом комунікації, апробації результатів досліджень та нетворкінгу. Зі зростанням популярності таких заходів збільшується і складність управління ними.

Типова міжнародна конференція сьогодні — це не просто набір доповідей, а складна ієрархічна структура подій. Вона включає роботу десятків секцій (track/session), сотні спікерів з різних країн, складну логістику приміщень та необхідність точного розподілу технічного обладнання.

Проблема масштабу. При переході від локальних семінарів до подій рівня "Надвеликої бази даних" (NVBD), організатори стикаються з проблемою обробки великих масивів інформації. Наприклад, якщо конференція проводиться щорічно протягом 10 років, накопичується база даних, що містить інформацію про тисячі учасників та сотні тисяч виступів. Ручна обробка таких обсягів (за допомогою Excel або паперових носіїв) призводить до:

- Втрати даних про історію виступів спікерів.

- Неможливості аналізу динаміки популярності тем.
- Помилки у розкладі (накладання секцій, подвійне бронювання аудиторій).
- Неефективного використання ресурсів (простоювання дорогого обладнання).

Тому метою роботи є створення автоматизованої системи, яка базується на реляційній базі даних та технологіях Business Intelligence, що дозволить вирішити ці проблеми.

1.2. Визначення та опис основних сутностей системи

На етапі системного аналізу, базуючись на технічному завданні (Варіант 3), було виділено ключові інформаційні об'єкти (сутності), якими оперує система.

Для забезпечення коректної структури даних було визначено таку ієрархію сутностей:

1. **Conferences (Конференції).** Це коренева сутність. Кожна конференція є унікальною подією, що має назву, дату початку та завершення, а також локацію (місто). Очікуваний обсяг даних: ~1 000 записів за 10 років.
2. **Sections (Секції).** Логічний підрозділ конференції. Оскільки конференція надто велика, щоб проходити в одному потоці, вона ділиться на тематичні секції (наприклад, "Штучний інтелект", "Бази даних", "Кібербезпека"). Секція завжди прив'язана до конкретної Конференції.
3. **Rooms (Зали/Аудиторії).** Фізичні місця проведення секцій. Кожна кімната має назву та, що критично важливо, місткість (Capacity). Це дозволяє контролювати переповнення залів.
4. **Speakers (Спікери/Виступаючі).** Фізичні особи, які проводять доповіді. Для аналітичних цілей важливо зберігати не тільки ім'я, але й:

- Країну походження (для географічного аналізу).
 - Email та телефон (для комунікації).
 - **Academic Degree (Науковий ступінь).** Це окремий довідник, необхідний для виконання вимоги про статистику виступів за ступеннями (PhD, Doctor of Science, Master).
5. **Equipment (Обладнання).** Ресурсна сутність. Включає типи обладнання: проектори, мікрофони, ноутбуки, дошки.
6. **Speeches (Виступи).** Це центральна транзакційна сутність системи (таблиця фактів). Виступ — це подія, коли конкретний *Спікер* виступає в конкретній *Секції* у визначений час.
- Очікуваний обсяг: >100 000 записів.
 - Атрибути: Тема доповіді, Час початку, Тривалість, Рейтинг.

Окремо слід виділити зв'язок "багато-до-багатьох" між Виступами та Обладнанням (**SpeechEquipment**). Один виступ може потребувати кілька одиниць техніки (наприклад, мікрофон + проектор), і один тип техніки використовується на багатьох виступах.

1.3. Аналіз бізнес-процесів

Процес 1: Планування структури конференції. Організатор створює запис про нову Конференцію. Далі, в рамках цієї конференції, створюються Секції. Система повинна перевіряти, щоб дати Секцій потрапляли в діапазон дат Конференції. *Вхід:* Дані про подію. *Вихід:* Сформована ієрархія "Конференція — Секції".

Процес 2: Управління спікерами та ступенями. При реєстрації спікера оператор обирає його науковий ступінь з довідника AcademicDegrees. Це забезпечує стандартизацію даних (унікнення дублів типу "PhD" та "Ph.D."). *Вхід:* Анкетні дані. *Вихід:* Унікальний запис у таблиці Speakers.

Процес 3: Формування розкладу (Scheduling). Найскладніший процес. Менеджер додає Виступ (Speech) у Секцію. При цьому система фіксує:

- Хто виступає (Link to Speaker).
- В якій секції (Link to Section).
- В якій кімнаті (Link to Room).
- Який час (Start Time + Duration).

Процес 4: Розподіл ресурсів (Equipment Allocation). Спікер заявляє потребу в техніці. Система створює записи в таблиці SpeechEquipment, зв'язуючи конкретний виступ з інвентарем. Це дозволяє формувати звіти "Потреба в обладнанні".

1.4. Функціональні вимоги до системи

Виходячи з аналізу предметної області та технічного завдання, сформульовано наступні вимоги:

1. Вимоги до зберігання даних:

- Система повинна зберігати історію за 10 років (2014–2024 рр.).
- База даних повинна містити не менше 1000 конференцій та 150 000 виступів для забезпечення репрезентативності аналітики.

2. Вимоги до цілісності:

- Видалення конференції повинно призводити до каскадного видалення або архівування всіх її секцій та виступів (Referential Integrity).
- Час завершення виступу не може бути меншим за час початку.

3. Вимоги до звітності:

- Система повинна надавати можливість формувати звіт по завантаженості обладнання.
- Необхідна аналітика по популярності тем (на основі рейтингів виступів).

- Статистика активності спікерів у розрізі наукових ступенів.

1.5. Функціональні вимоги до системи

Однією з ключових проблем при розробці навчальних систем класу NVBD є відсутність реальних даних великого обсягу. Ручне введення 150 000 записів неможливе. Написання власних скриптів (loop inserts) часто призводить до створення одноманітних, нереалістичних даних, які погано підходять для аналітики.

Для вирішення цієї проблеми в курсовій роботі обрано інструмент **Redgate SQL Data Generator**.

Переваги використання Redgate:

1. **Збереження референційної цілісності:** Інструмент автоматично зчитує зовнішні ключі (Foreign Keys). Він спочатку генерує Конференції, потім Секції (прив'язуючи їх до існуючих ID конференцій), і тільки потім Виступи. Це гарантує відсутність "сирітських" записів.
2. **Реалістичність розподілу:** Можливість налаштування генераторів імен, email-адрес, реальних міст та країн.
3. **Часові ряди:** Інструмент дозволяє задати чіткий період генерації (2014–2024), що є критичним для побудови Time Dimension в OLAP-кубі.
4. **Швидкодія:** Генерація 500 000+ рядків займає лічені хвилини, що дозволяє швидко проводити ітерації розробки.

Розділ 2. Проектування бази даних

2.1. Концептуальне проектування:

На етапі концептуального проектування було розроблено модель "Сутність-Зв'язок" (Entity-Relationship Model), що відображає логічну структуру предметної області "Конференції". Головною метою цього етапу є забезпечення цілісності даних та уникнення надлишковості.

Враховуючи специфіку завдання (Варіант 3), було виділено наступні сутності та типи зв'язків між ними:

1. **Conferences (Конференції) — Sections (Секції).**

- *Тип зв'язку:* Один-до-багатьох (1:M).
- *Логіка:* Одна конференція складається з багатьох секцій, але кожна секція належить лише одній конференції.

2. **Sections (Секції) — Speeches (Виступи).**

- *Тип зв'язку:* Один-до-багатьох (1:M).
- *Логіка:* У секції відбувається багато виступів.

3. **Speakers (Спікери) — Speeches (Виступи).**

- *Тип зв'язку:* Один-до-багатьох (1:M).
- *Логіка:* Один спікер може мати декілька виступів на різних конференціях протягом років.

4. **AcademicDegrees (Наукові ступені) — Speakers (Спікери).**

- *Тип зв'язку:* Один-до-багатьох (1:M).
- *Логіка:* Довідкова сутність для стандартизації назв ступенів (PhD, Професор).

5. **Speeches (Виступи) — Equipment (Обладнання).**

- *Тип зв'язку:* Багато-до-багатьох (M:N).
- *Реалізація:* Оскільки один виступ може потребувати різного обладнання, а один тип обладнання використовується на багатьох

виступах, цей зв'язок реалізовано через асоціативну таблицю SpeechEquipment.

ER-ДІАГРАМА 3 SSMS

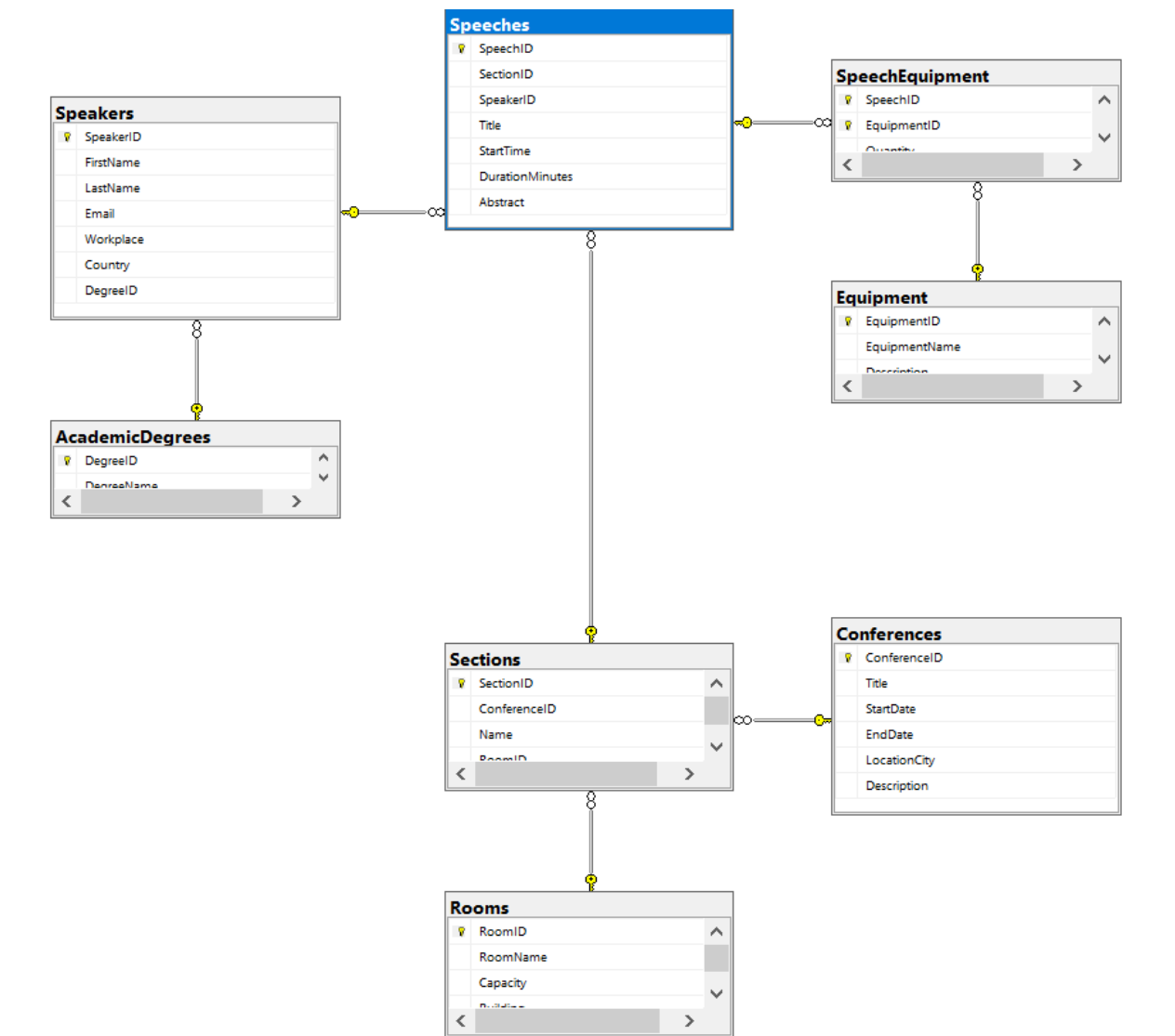


Рис. 2.1 ER-ДІАГРАМА

2.2. Логічне проектування та нормалізація

Розроблена схема бази даних відповідає вимогам Третьої нормальної форми (3НФ):

- 1НФ**: Всі атрибути є атомарними (наприклад, ім'я та прізвище розділені), відсутні групи, що повторюються.
- 2НФ**: Всі неключові атрибути залежать від повного первинного ключа.

3. **ЗНФ:** Відсутні транзитивні залежності між неключовими атрибутами (наприклад, назва наукового ступеня винесена в окремий довідник AcademicDegrees, а в таблиці Speakers зберігається лише ID).


Опис структури основних таблиць:

- **Table: Conferences**
 - ConferenceID (INT, PK, Identity) — унікальний ідентифікатор.
 - Title (NVARCHAR) — назва заходу.
 - StartDate / EndDate (DATE) — дати проведення.
 - LocationCity (NVARCHAR) — місто.
- **Table: Speeches (Таблиця фактів OLTP)**
 - SpeechID (INT, PK, Identity).
 - SectionID (INT, FK) — посилання на секцію.
 - SpeakerID (INT, FK) — посилання на спікера.
 - StartTime (DATETIME) — точний час початку.
 - DurationMinutes (INT) — тривалість.
 - Abstract (NVARCHAR MAX) — короткий опис доповіді.
- **Table: SpeechEquipment (Таблиця зв'язку)**
 - SpeechID (INT, FK).
 - EquipmentID (INT, FK).
 - Quantity (INT) — кількість одиниць.
 - Primary Key: Складений ключ (SpeechID, EquipmentID).

2.3. Стратегія генерації великих обсягів даних

Однією з ключових вимог до курсової роботи є наповнення бази даних обсягом не менше 500 000 записів для перевірки продуктивності. Ручне створення такого масиву даних неможливе, тому було обрано автоматизований підхід з використанням інструменту **Redgate SQL Data Generator**.

Процес генерації був налаштований таким чином, щоб забезпечити реалістичність даних та їх узгодженість.

 **redgate**

SQL Data Generator - New File

Target server: DESKTOP-IFD83CQ\LOCALHOST**Target database:** ConferenceDB

Date generation started at: 28 грудня 2025 р. 19:06:50ended at: 28 грудня 2025 р. 19:13:15

[dbo].[Rooms]
Rows inserted: 50
Generation started at 28 грудня 2025 р. 19:06:50, taken: less than a second

[dbo].[Equipment]
Rows inserted: 20
Generation started at 28 грудня 2025 р. 19:06:50, taken: less than a second

[dbo].[Conferences]
Rows inserted: 2,000
Generation started at 28 грудня 2025 р. 19:06:50, taken: less than a second

[dbo].[Sections]
Rows inserted: 1,000
Generation started at 28 грудня 2025 р. 19:06:51, taken: less than a second

[dbo].[AcademicDegrees]
Rows inserted: 6
Generation started at 28 грудня 2025 р. 19:06:51, taken: less than a second

[dbo].[Speakers]
Rows inserted: 100,000
Generation started at 28 грудня 2025 р. 19:06:51, taken: 00:00:02 (hh:mm:ss)

[dbo].[Speeches]
Rows inserted: 500,000
Generation started at 28 грудня 2025 р. 19:06:53, taken: 00:05:05 (hh:mm:ss)

[dbo].[SpeechEquipment]
Rows inserted: 600,000
Generation started at 28 грудня 2025 р. 19:11:58, taken: 00:01:16 (hh:mm:ss)

Рис 2.2. Генерація даних в Redgate

Розділ 3. Реалізація ETL-процесів

3.1. Створення та конфігурація SSIS-проекту

У середовищі Visual Studio (SSDT) створено новий проект типу «Integration Services Project» з назвою ConferenceETL.

Налаштування з'єднань (Connection Managers): Для роботи пакету налаштовано два менеджери з'єднань OLE DB, які забезпечують зв'язок між джерелом та призначенням:

1. **localhost.ConferenceDB:** Підключення до операційної бази даних (Source), звідки витягуються "сирі" дані.
2. **localhost.ConferenceDW:** Підключення до сховища даних (Destination), куди завантажуються очищені та трансформовані дані.

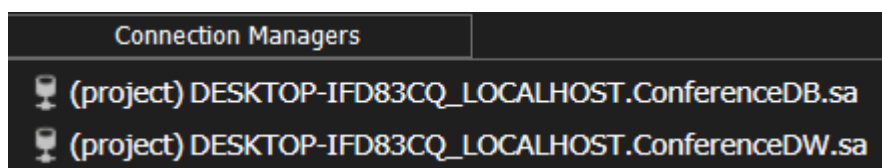


Рис 3.1. Менеджери з'єднань з базами даних

3.2. Реалізація потоку керування (Control Flow)

Для структурування логіки виконання пакету використано компонент **Sequence Container**, названий «ETL Process». Це дозволяє групувати пов'язані завдання та керувати транзакціями.

Потік керування складається з наступних кроків:

1. **Execute SQL Task ("Truncate DW Tables"):** Виконує SQL-команду TRUNCATE TABLE та DELETE FROM для очищення таблиць перед завантаженням. Це гарантує відсутність дублікатів при повторному запуску пакету.

2. **Data Flow Tasks:** Серія блоків, що відповідають за фізичне переміщення даних. Встановлено сувору послідовність виконання (Precedence Constraints), що відображено зеленими стрілками:
- Load DimRooms -> Load DimConferences -> Load DimSpeakers -> Load DimEquipment -> Load FactSpeeches -> Load FactEquipmentUsage.
3. **Script Task ("Post-Processing"):** Використовує мову C# для реалізації складної бізнес-логіки, яку важко виконати стандартними компонентами (наприклад, перевірка флагов успішного завершення).

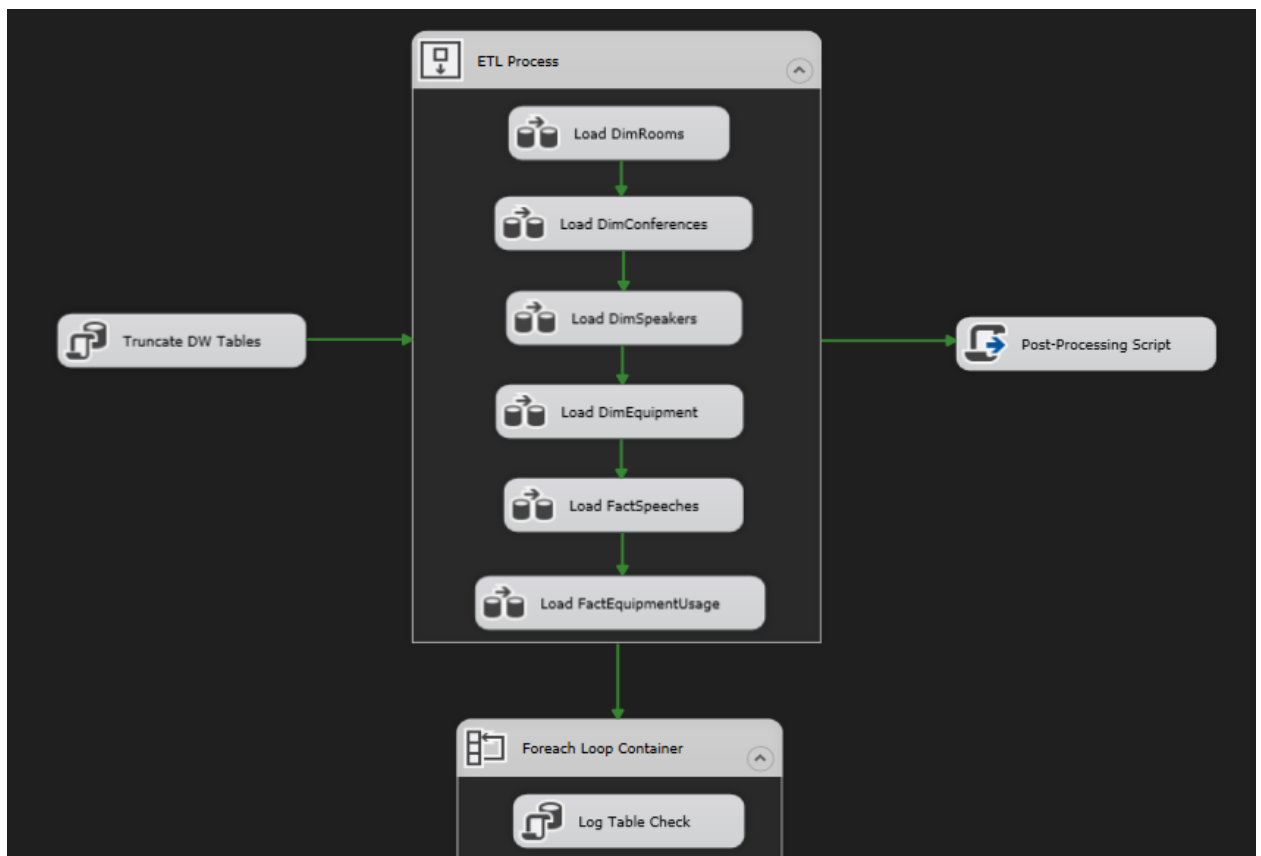


Рис 3.2. Загальна схема потоку керування (Control Flow).

3.3. Реалізація потоку даних (Data Flow) та трансформацій

У даному підрозділі описано деталі реалізації окремих Data Flow Tasks.

3.3.1. Завантаження простих вимірів (Rooms, Conferences, Equipment) Для цих таблиць використано пряме відображення (Direct Mapping) стовпців джерела на стовпці призначення.

- У потоці Load DimEquipment додатково використано компонент **Derived Column** для встановлення значення за замовчуванням для поля Type.

3.3.2. Завантаження виміру Спікерів (Load DimSpeakers) Цей потік містить обов'язкові трансформації даних:

1. **Derived Column:** Створення поля FullName (FirstName + " " + LastName).
2. **Data Conversion:** Приведення текстових полів до формату Unicode (DT_WSTR).
3. **Conditional Split:** Фільтрація записів.
 - Умова: !ISNULL(Email)
 - Дія: Спікери без Email вважаються некоректними даними і не завантажуються.

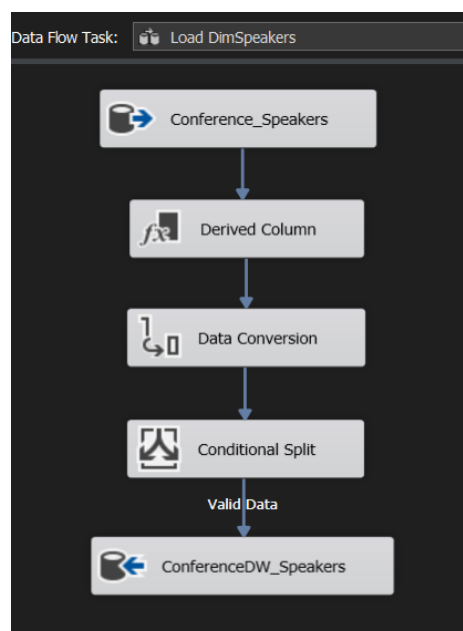


Рис 3.3. Трансформації при завантаженні спікерів

3.4.3. Завантаження таблиці фактів виступів (Load FactSpeeches) Це найскладніший потік, що зв'язує всі дані.

1. **Lookups (Пошук ключів):** Використано три послідовні компоненти Lookup для заміни ID з операційної бази на ключі сховища:
 - Пошук SpeakerKey у DimSpeakers.
 - Пошук ConferenceKey у DimConferences.
 - Пошук RoomKey у DimRooms.
2. **Derived Column:** Генерація DateKey з дати початку виступу.
3. **Conditional Split:** Перевірка цілісності. Завантажуються лише ті факти, де успішно знайдено всі три ключі (Speaker, Conference, Room).

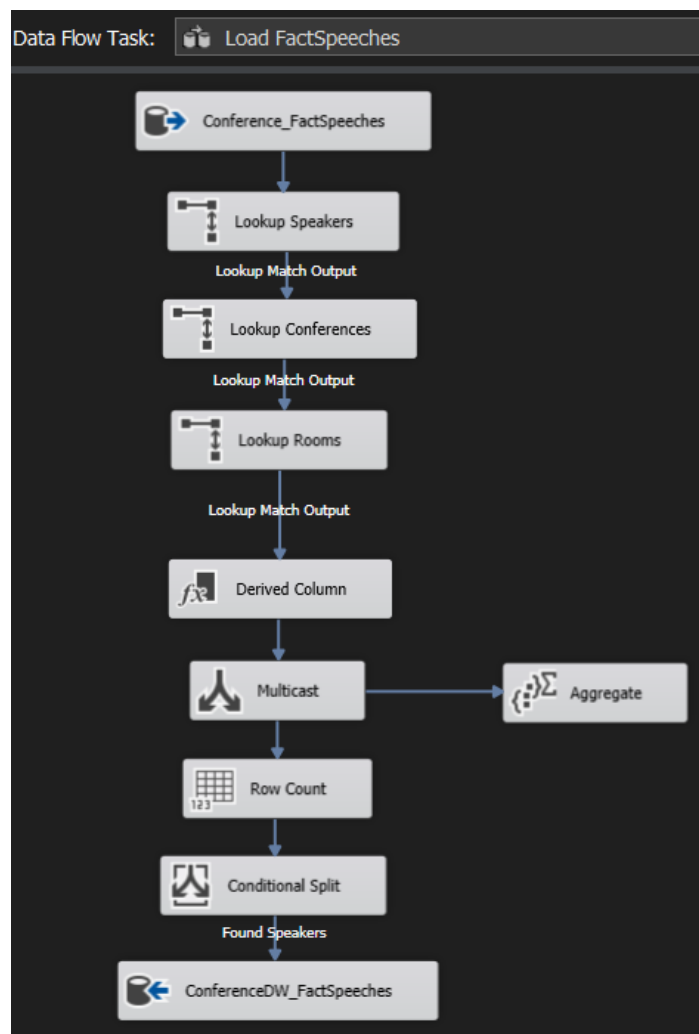


Рис 3.4. Потік даних FactSpeeches з використанням Lookup

3.4.4. Завантаження фактів використання обладнання (Load FactEquipmentUsage) Для реалізації другої таблиці фактів використано підхід генерації даних на рівні джерела.

- **Source Query:** Використано SQL-запит із CROSS JOIN між Конференціями та Обладнанням, що імітує розподіл ресурсів.
- **Lookups:** Виконано пошук ключів обладнання та конференцій.
- **Destination:** Дані завантажено у FactEquipmentUsage.

Розділ 4. Побудова OLAP-куба

4.1. Створення проекту багатовимірної аналізи (SSAS)

Для реалізації аналітичної складової системи обрано технологію **Microsoft SQL Server Analysis Services (SSAS)** у режимі Multidimensional. Цей підхід дозволяє попередньо агрегувати великі обсяги даних, що забезпечує миттєвий відгук системи при формуванні складних звітів.

У середовищі Visual Studio створено проект типу Analysis Services Multidimensional and Data Mining Project з назвою **ConferenceCube**.

Налаштування джерела даних (Data Source): Налаштовано підключення до створеного раніше сховища даних ConferenceDW.

- *Provider:* Native OLE DB \ SQL Server Native Client 11.0.
- *Impersonation Info:* Використано режим Inherit для забезпечення безперебійного доступу до даних у середовищі розробки.

4.2. Розробка представлення джерела даних (DSV)

Для абстрагування фізичної структури бази даних від логічної моделі куба створено **Data Source View (DSV)**. До нього включено всі необхідні таблиці: дві таблиці фактів (FactSpeeches, FactEquipmentUsage) та п'ять таблиць вимірів.

Іменовані обчислення (Named Calculations): Для покращення читабельності звітів у таблиці DimDate створено віртуальний стовпець QuarterName.

- *SQL Expression:* 'Q' + CAST([Quarter] AS VARCHAR(1)) + ' ' + CAST([Year] AS VARCHAR(4))
- *Результат:* Замість простого номера кварталу користувач бачить зрозумілий підпис, наприклад, "Q1 2024".

4.3.Проектування вимірів (Dimensions) та ієрархій

На основі таблиць вимірів розроблено об'єкти Dimensions. Особливу увагу приділено налаштуванню атрибутів та ієрархій для зручної навігації (Drill-down).

1. Вимір Часу (DimDate): Це ключовий вимір для аналітики.

- **Ієрархія (Calendar Hierarchy):** Year -> QuarterName -> MonthName -> Day.
- **Сортування:** Для атрибута MonthName налаштовано сортування за ключем (KeyColumns), щоб місяці відображалися у календарному порядку (Січень, Лютий...), а не за алфавітом.
- **Зв'язки атрибутів (Attribute Relationships):** Налаштовано жорсткі зв'язки між рівнями ієрархії для оптимізації продуктивності запитів.

2. Інші виміри:

- **DimSpeakers:** Включає атрибути FullName, Email, AcademicDegree, Country.
- **DimConferences:** Дозволяє аналізувати дані в розрізі міст (LocationCity) та назв подій.
- **DimEquipment:** Створено ієрархію Type -> EquipmentName для аналізу категорій обладнання.

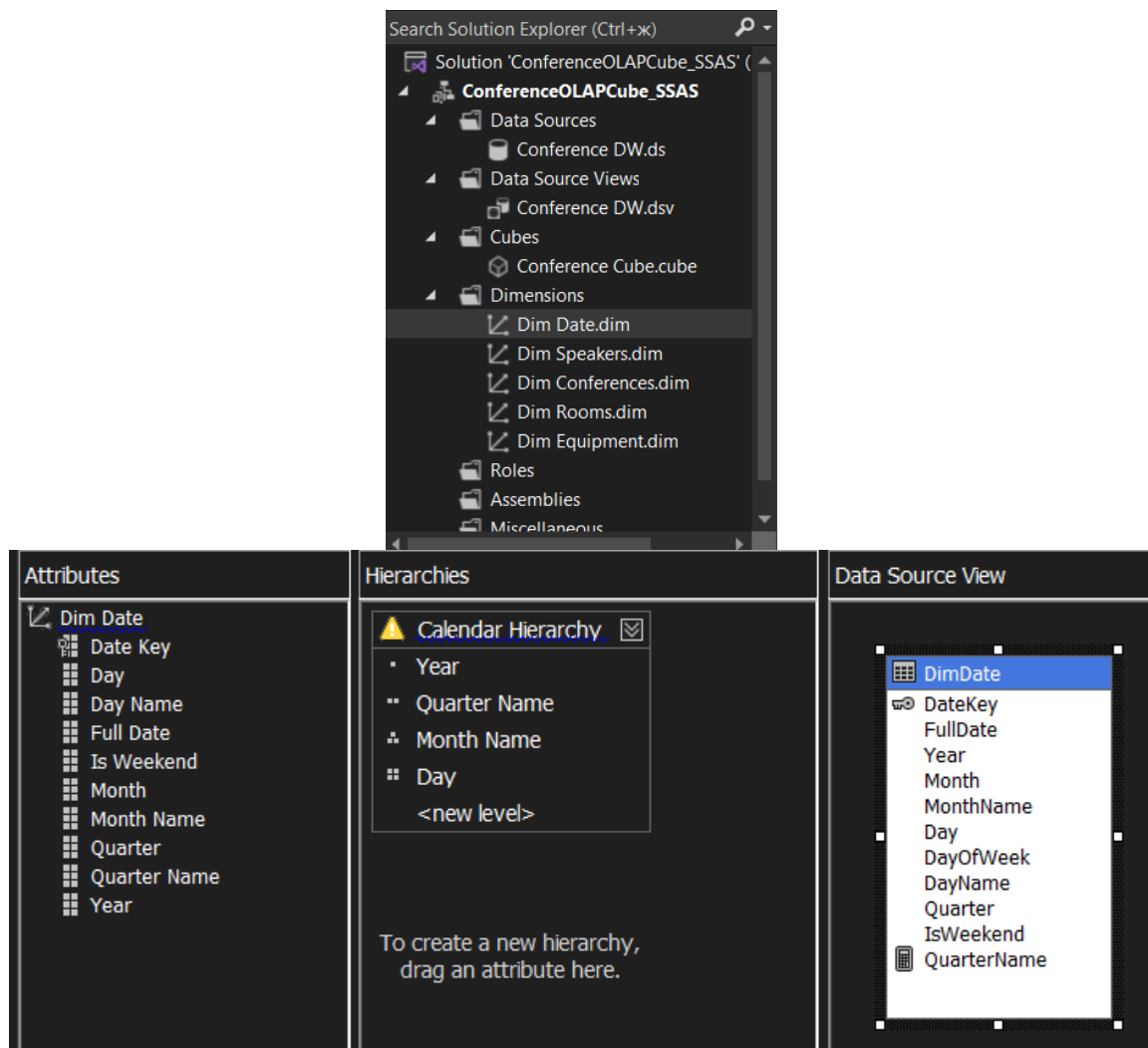


Рис 4.1. Структура виміру часу та налаштована ієрархія.

4.4. Розробка структури Куба та Мір

Створено куб Conference Cube, який підтримує архітектуру з кількома групами мір (Measure Groups). Це дозволяє аналізувати різні бізнес-процеси в єдиному просторі.

Реалізовані типи агрегації (відповідно до вимог ТЗ):

1. Sum (Сума):

- Duration Minutes — загальна тривалість виступів.
- Quantity Used — загальна кількість використаного обладнання.

2. Count (Кількість рядків):

- Speeches Count — кількість проведених доповідей.

- Equipment Usage Count — кількість фактів видачі обладнання.

3. Maximum / Minimum (Екстремуми):

- Max Duration — найдовший виступ.
- Min Duration — найкоротший виступ.

4. Distinct Count (Унікальний підрахунок):

- Unique Speakers Count — кількість унікальних спікерів. Ця міра дозволяє оцінити реальне охоплення аудиторії без дублювання осіб, що виступали кілька разів.

Такий набір мір забезпечує всебічний статистичний аналіз діяльності конференцій.

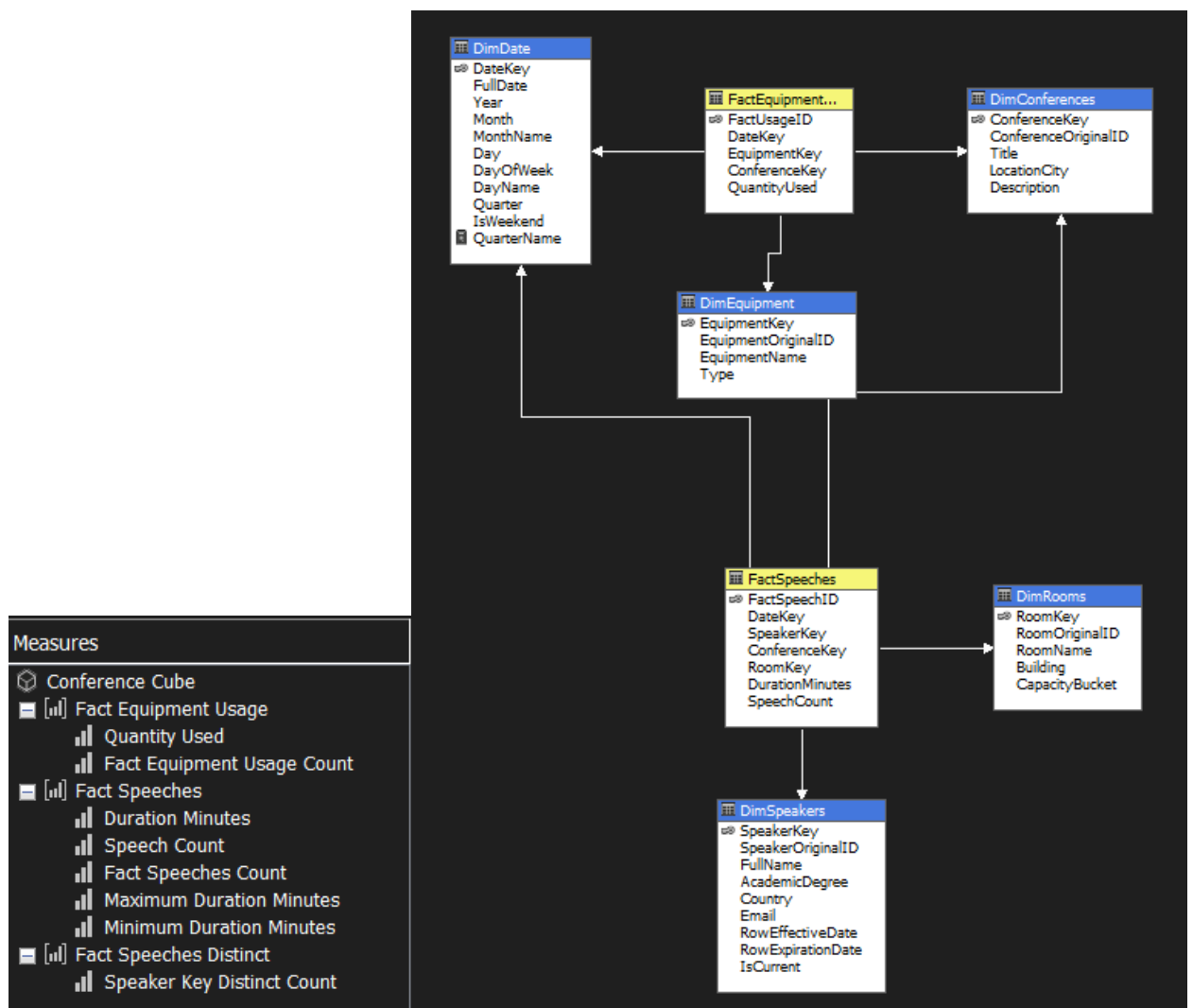


Рис 4.2. Структура куба та групи мір

4.5. Реалізація бізнес-логіки (MDX Scripting)

Для забезпечення глибокого аналізу даних стандартних агрегацій (сума, кількість) недостатньо. Тому на вкладці **Calculations** було розроблено набір обчислюваних членів (Calculated Members) з використанням мови запитів MDX.

Реалізовано наступні показники:

1. **Average Duration (Середня тривалість):** Розраховується як відношення загального часу до кількості виступів. Використано функцію IIF для обробки ділення на нуль. *Формула:* [Measures].[Duration Minutes] / [Measures].[Speeches Count]
2. **Avg Equipment Per Speech (Навантаження на обладнання):** Бізнес-метрика, що показує середню кількість одиниць техніки на один виступ.
3. **Duration YTD (Year To Date):** Показник часового інтелекту (Time Intelligence), що розраховує наростаючий підсумок з початку року. *Функція:* PeriodsToDate([DimDate].[Calendar Hierarchy].[Year], ...)
4. **Growth Percentage (Темп зростання):** Показник динаміки (Year-over-Year / Month-over-Month). Розраховує відсоток зміни тривалості виступів порівняно з попереднім періодом.
5. **Speaker Rank (Рейтинг спікера):** Визначає позицію спікера в загальному рейтингу на основі кількості його виступів. Використано функцію RANK та сортування BDESC.

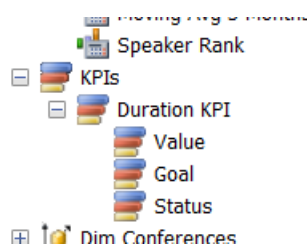


Рис 4.3. Реалізація бізнес-логіки мовою MDX.

Ключові показники ефективності (KPI)

Для візуальної оцінки стану справ розроблено KPI «**Duration KPI**».

- **Goal (Ціль):** 45 хвилин (оптимальна тривалість).
- **Status (Статус):** Реалізовано логіку «Світлофора»:
 - *Зелений:* 40-50 хвилин.
 - *Червоний:* <30 або >60 хвилин.
 - *Жовтий:* інші значення.

	A	B	C
1	Duration KPI	Duration KPI Мета	Duration KPI Стан
2	74,98	45	○
3			

Рис 4.4. Візуалізація KPI у браузері куба і Exel

Оптимізація та налаштування доступу

Для підвищення продуктивності та зручності користування виконано наступні дії:

1. **Агрегації (Aggregations):** За допомогою майстра *Usage Based Optimization* спроектовано схему агрегацій для таблиць фактів із приростом продуктивності 30%. Це дозволяє системі миттєво віддавати попередньо пораховані суми.
2. **Перспективи (Perspectives):** Створено спрощену перспективу Manager View, в якій приховано технічні поля (ключі, ID) та залишено лише бізнес-метрики, необхідні для прийняття управлінських рішень.

Fact Equipment Usage (1 Aggregation Design)			
AggregationDesign	2	40000	Fact Equipment Usage
Fact Speeches (1 Aggregation Design)			
AggregationDesign	5	494866	Fact Speeches
Fact Speeches Distinct (0 Aggregation Designs)			
Unsigned Aggregation Design	-	-	Fact Speeches

Рис 4.5. Схема агрегацій для таблиць фактів


Cube Objects	Object Type	Perspective Name
 Conference Cube	Name	Manager View
	DefaultMeas...	
[-] Measure Groups		
- Fact Equipment Usage	MeasureGro...	<input checked="" type="checkbox"/>
Quantity Used	Measure	<input checked="" type="checkbox"/>
Fact Equipment Usage C...	Measure	<input checked="" type="checkbox"/>
- Fact Speeches	MeasureGro...	<input checked="" type="checkbox"/>
Duration Minutes	Measure	<input checked="" type="checkbox"/>
Speech Count	Measure	<input checked="" type="checkbox"/>
Fact Speeches Count	Measure	<input type="checkbox"/>
Maximum Duration Minutes	Measure	<input checked="" type="checkbox"/>
Minimum Duration Minutes	Measure	<input checked="" type="checkbox"/>
- Fact Speeches Distinct	MeasureGro...	<input checked="" type="checkbox"/>
Speaker Key Distinct Count	Measure	<input checked="" type="checkbox"/>
[-] Dimensions		
+ Dim Equipment	CubeDimen...	<input checked="" type="checkbox"/>
+ Dim Date	CubeDimen...	<input checked="" type="checkbox"/>
+ Dim Conferences	CubeDimen...	<input checked="" type="checkbox"/>
+ Dim Speakers	CubeDimen...	<input checked="" type="checkbox"/>
+ Dim Rooms	CubeDimen...	<input checked="" type="checkbox"/>
[-] KPIs		
Duration KPI	Kpi	<input checked="" type="checkbox"/>
[-] Calculations		
Speaker Rank	CalculatedM...	<input type="checkbox"/>
Moving Avg 3 Months	CalculatedM...	<input type="checkbox"/>
Average Duration	CalculatedM...	<input checked="" type="checkbox"/>

Рис 4.6. Перспективи

4.6. Розгортання та перевірка OLAP-куба

Після завершення проектування виконано процедуру розгортання (Deployment) проекту на локальний сервер SSAS. Процес включав:

1. Валідацію структури.
2. Створення фізичних файлів бази даних.
3. Процесинг (Processing) — завантаження даних зі сховища ConferenceDW у структури куба.

Статус розгортання: **Deployment Completed Successfully.**

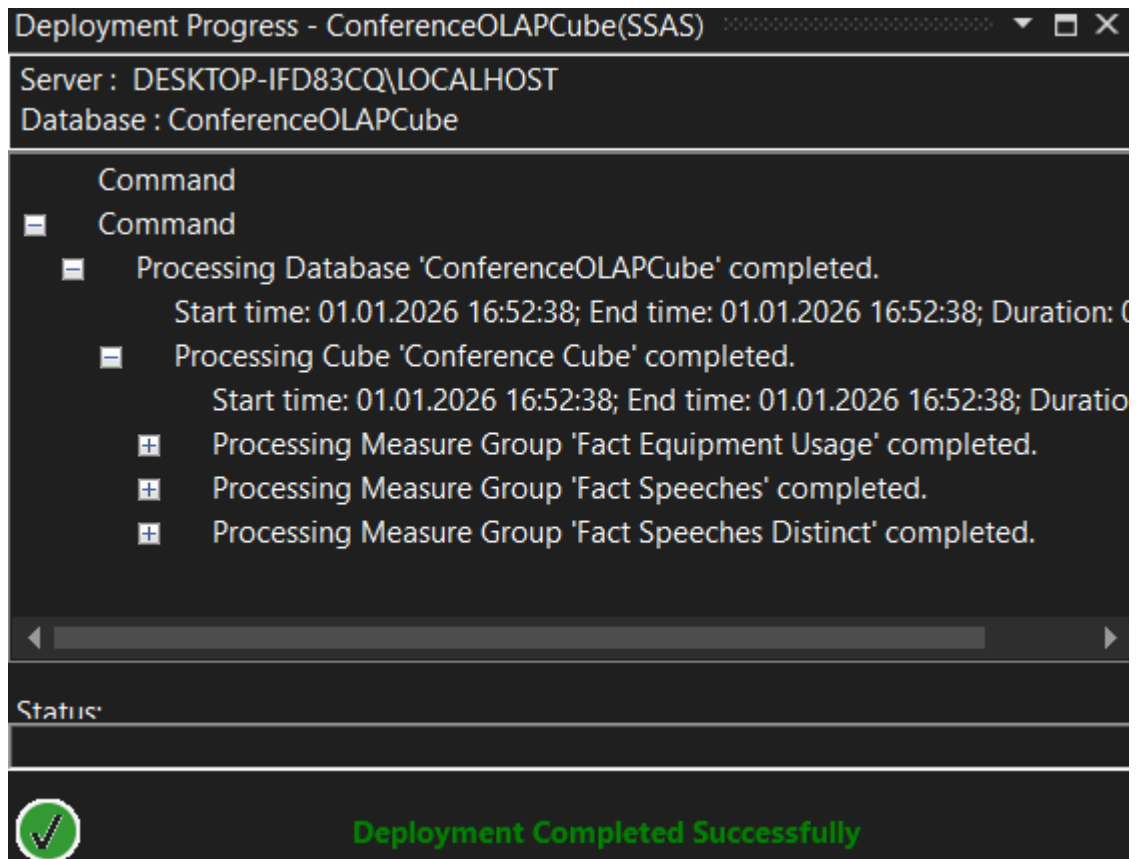


Рис 4.7. Успішне розгортання проекту.

Розділ 5. Аналітичні звіти (SQL Server Reporting Services)

5.1. Налаштування середовища звітності

Розробка звітів виконувалася у середовищі Visual Studio в рамках проекту типу **Report Server Project** з назвою ConferenceReports. Для доступу до даних OLAP-куба налаштовано спільне джерело даних (Shared Data Source) DS_ConferenceCube, яке використовує провайдер **Microsoft SQL Server Analysis Services**. Це забезпечує прямий доступ до вимірів, мір та KPI, створених на попередньому етапі.

5.2. Розробка комплексу звітів

Відповідно до технічного завдання та бізнес-вимог, розроблено 5 типів звітів, що покривають різні аспекти аналізу конференцій.

5.2.1. Табличний звіт «Список учасників з місцем праці» (Tabular Report)

Призначення: Детальний перелік спікерів у розрізі конференцій. *Реалізація:*

- Використано групування даних за назвою конференції (Conference Title).
- У деталях виведено: Ім'я спікера, Науковий ступінь, Кількість виступів.
- Додано **Interactive Sorting** (Інтерактивне сортування), що дозволяє користувачеві сортувати список спікерів за алфавітом безпосередньо у браузері.
- Додано підсумкові рядки (Totals) для підрахунку загальної кількості виступів на кожній конференції.

Report_Participants_Table

Title	Full Name	Academic Degree	Country	Speech Count
Adbanicator International	Albert Weaver	Доктор філософії	Germany	5
	Alissa Moore	Аспірант	Palau	5
	Allyson Madden	Професор	Tuvalu	5
	Angela Henson	Кандидат наук	Pitcairn	5
	...			
	Travis Reese	Професор	Canada	5
	Vicky Atkins	Кандидат наук	Slovakia	5
	Wayne Williams	Професор	Sri Lanka	5
	Yesenia Burgess	Доктор наук	Réunion	5
	Yvette Mayer	Доцент	Jordan	5
				494

Рис 5.1. Табличний звіт учасників із групуванням.

5.2.2. Матричний звіт «Потреба в обладнанні» (Matrix Report)

- **Рядки:** Типи обладнання (Equipment Name).
- **Стовпці:** Міста проведення (Location City).
- **Дані:** Кількість використань (Speech Count / Usage Count).
- **Conditional Formatting (Умовне форматування):** Застосовано вираз $=\text{IF}(\text{Sum}(\dots) < 50, \text{"Red"}, \text{"Green"})$, який підсвічує зеленим кольором позиції з високим навантаженням (понад 50 одиниць). Це дозволяє швидко виявляти "вузькі місця" у матеріальному забезпеченні.

Report_Equipment_Matrix

	Akron	Albuquerque	Anaheim	Anchorage	Arlington	/
Wi-Fi роутер	26	38	36	44	90	
Екран	26	38	36	44	90	
Клікер	39	57	54	66	135	
Маркерна дошка	26	38	36	44	90	
Мікрофон	91	133	126	154	315	
Ноутбук	39	57	54	66	135	
Проектор	13	19	18	22	45	

Рис 5.2. Матриця розподілу обладнання з умовним форматуванням.

5.2.3. Аналітичний звіт з діаграмами «Статистика виступів по науковим ступеням» (Charts)

Призначення: Візуалізація розподілу спікерів за кваліфікацією. Для забезпечення реалістичності даних було проведено ETL-трансформацію (SQL Update), що сформувала нерівномірний розподіл наукових ступенів (більшість — кандидати наук, меншість — професори). У звіті реалізовано чотири типи візуалізації:

1. **Pie Chart (Кругова):** Частка кожного наукового ступеня у загальній кількості.
2. **Column Chart (Стовпчикова):** Кількісне порівняння активності.
3. **Bar Chart (Лінійна) :** Рейтинг активності.
4. **Area Chart:** Площинна діаграма.

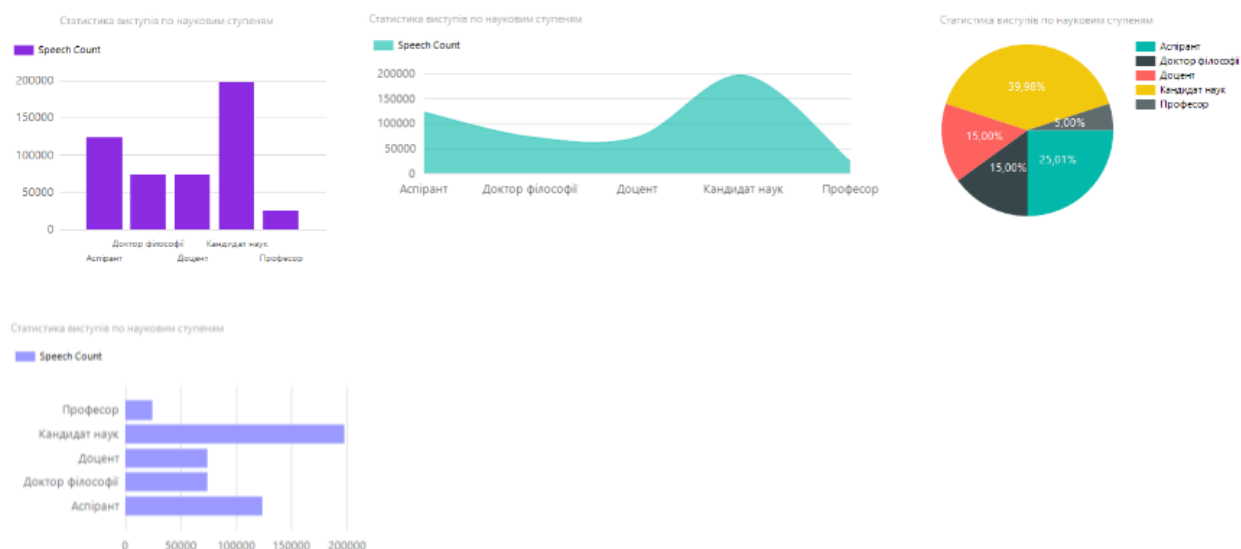


Рис 5.3. Комплексна візуалізація статистики наукових ступенів.

5.2.4. Інтерактивний Дашборд «Аналіз популярності тематик» (Dashboard)

1. **Sparklines (Спарклайни):** Міні-графіки, вбудовані у таблицю міст. Вони показують тренд розподілу тематик за кількістю конференцій.

2. **Gauge (Індикатор/Спідометр):** Візуалізує загальну кількість виступів (Speech Count) відносно планового показника (наприклад, 100 000). Це дозволяє миттєво оцінити масштаб діяльності.

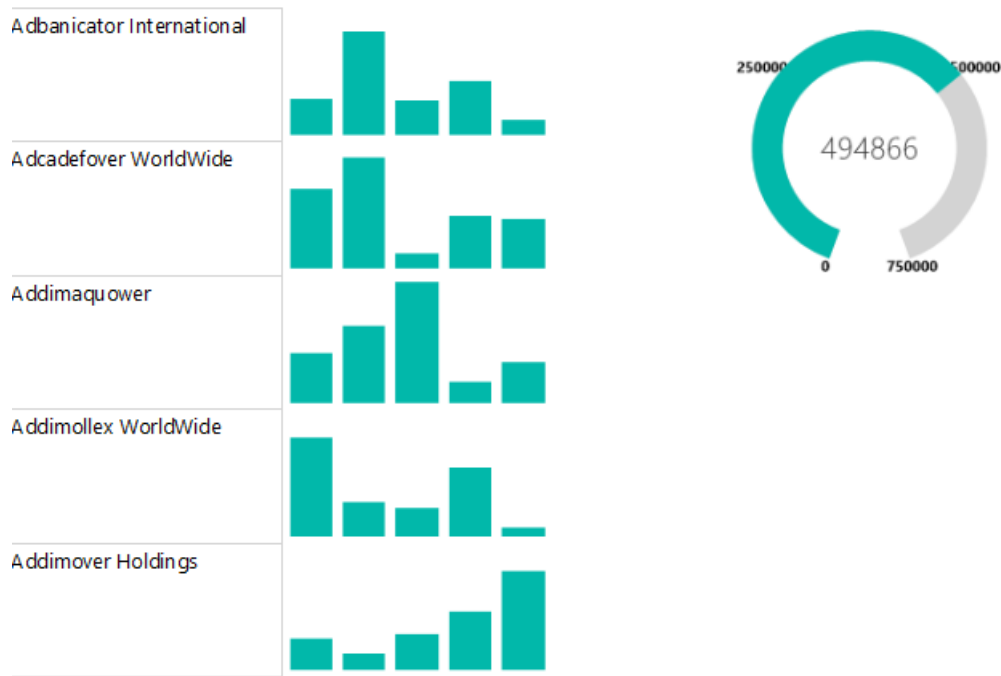


Рис 5.4. Аналітичний дашборд.

5.2.5. Деталізований звіт «Розклад конференції» (Drill-down)

Призначення: Перегляд ієрархічних даних від загального до конкретного.

Реалізація:

- Налаштовано видимість рядків (Visibility: Hide).
- Встановлено перемикач (Toggle Item) на поле назви конференції.
- Користувач бачить лише список подій. При натисканні на знак [+] розгортається детальний список спікерів та виступів цієї конференції.

Title	Full Date	Room Name	<input checked="" type="checkbox"/> Full Name	Speech Count	Duration Minutes
-------	-----------	-----------	---	--------------	------------------

Title	Full Date	Room Name	Full Name	Speech Count	Duration Minutes
Thrufropax Holdings	2024-09-27	Аудиторія393	Jonathon Wallace	1	110
Raptuminor Holdings	2024-06-02	Аудиторія401	Kisha Andrade	1	49
Unjubefax WorldWide	2024-07-20	Зал593	Victoria Levine	1	48
			Russell Aguirre	1	62
Doprobefazz Company	2024-12-30	Зал569	Robert Small	1	72
Ciperax Holdings Group	2024-05-24	Зал185	Sidney Powell	1	88
Supsipor Holdings	2024-02-22	Аудиторія281	Kenya Rangel	1	52
			Gail Santos	1	78
Truhupentor Holdings Inc	2024-03-30	Аудиторія169	Edwin Hensley	1	75
Emerex Group	2024-07-24	Аудиторія330	Trent Bird	1	32
Zeefropazz Holdings	2024-05-28	Зал563	Cheryl Pace	1	68
Resipiman	2024-10-20	Зал248	Seth Jackson	1	85

Рис 5.5. Звіт з функцією Drill-down.

5.3. Реалізація параметризації звітів

1. Dropdown Parameter (Випадаючий список) *Реалізація:* У звіті «Список учасників» (Report_Participants) додано параметр для вибору конкретної конференції або міста. Значення для списку підтягуються автоматично з окремого Dataset. *Перевага:* Запобігає помилкам ручного введення та прискорює пошук.

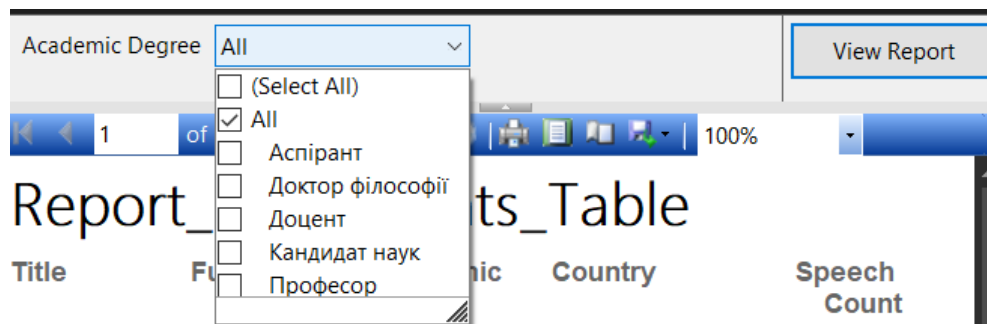


Рис 5.6. Використання параметра типу Dropdown.

2. Multi-select Parameter (Множинний вибір) *Реалізація:* У тому ж звіті реалізовано можливість обирати декілька наукових ступенів одночасно (наприклад, відобразити тільки «Професорів» та «Доцентів»). *Перевага:* Дозволяє формувати кастомні вибірки даних для порівняльного аналізу.

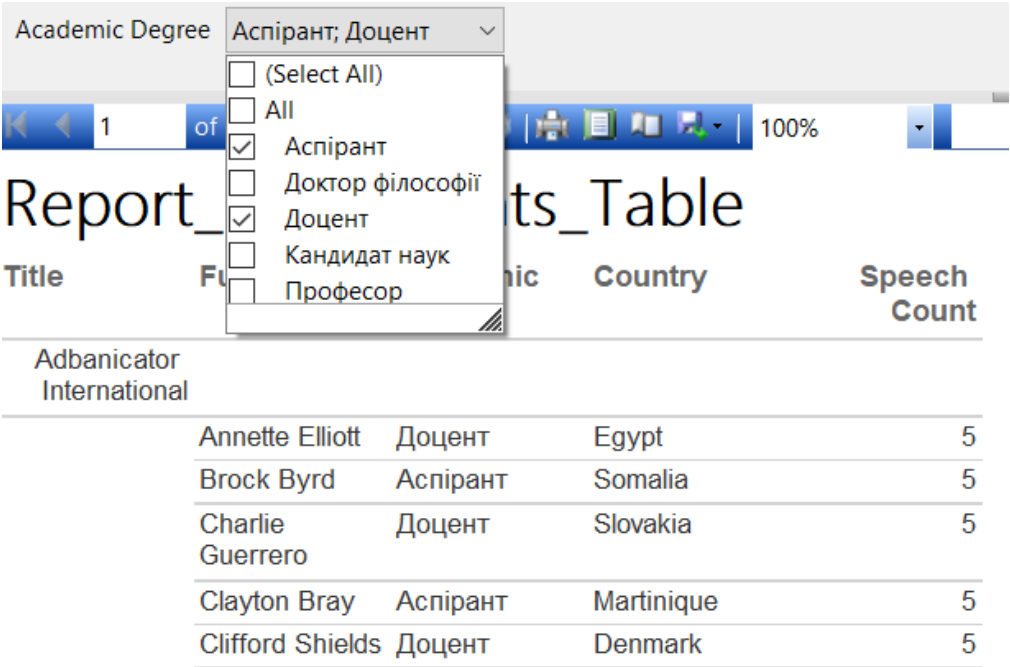


Рис 5.7. Параметр з можливістю множинного вибору (Multi-select).

3. Date Range Parameter (Діапазон дат) *Реалізація:* У звіті «Розклад Конференції» (Report_Schedule) додано два параметри: StartDate та EndDate. *Перевага:* Користувач може відфільтрувати події за певний період (наприклад, за поточний рік або конкретний місяць).

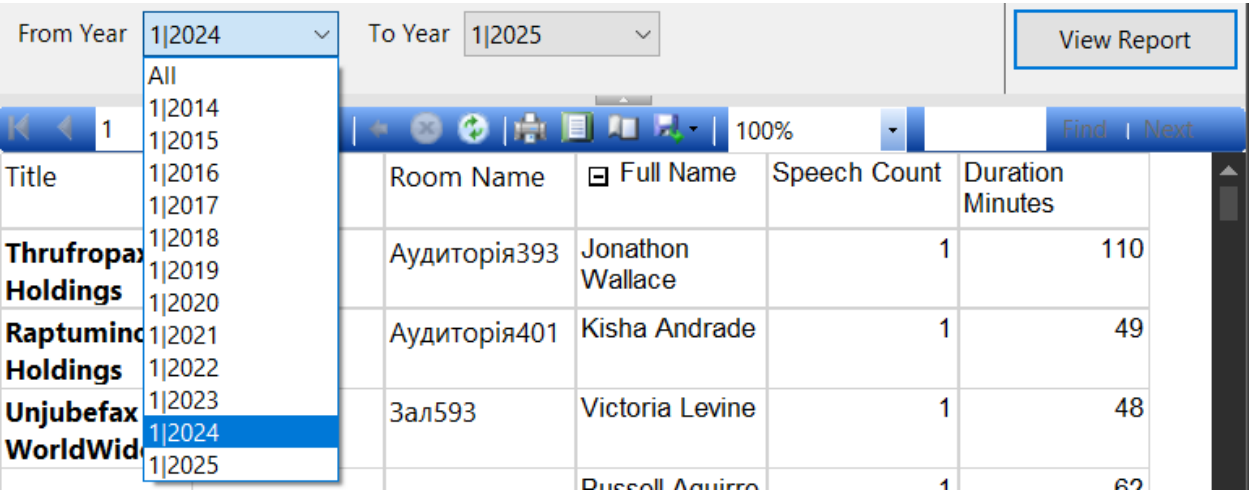


Рис 5.8. Фільтрація даних за часовим діапазоном.

4. Boolean Parameter (Логічний перемикач) *Реалізація:* У звіті з діаграмами (Report_Charts) додано параметр-прапорець (True/False). *Функціонал:* Дозволяє вмикати або вимикати відображення підписів даних (Data Labels) на графіках, що робить візуалізацію менш завантаженою за потреби.

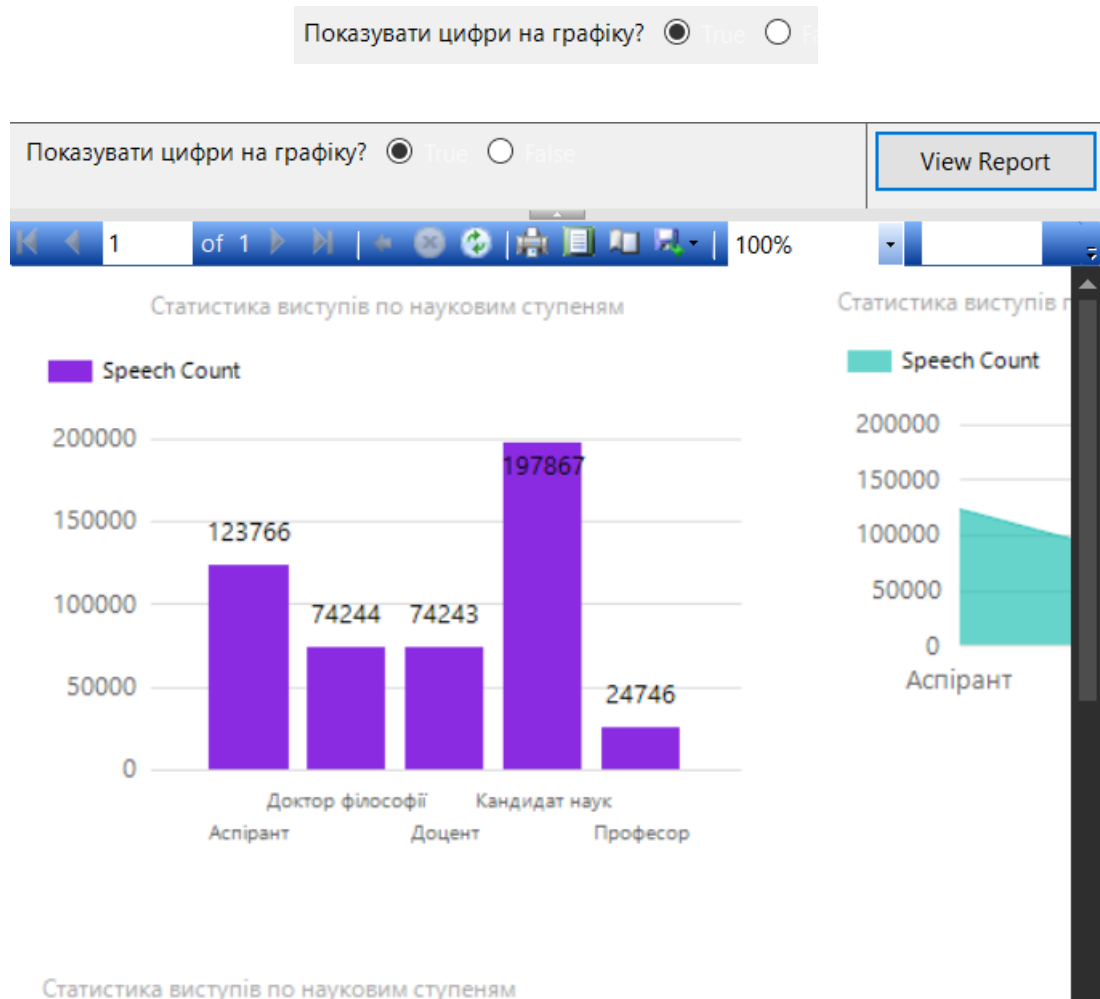


Рис 5.9. Використання логічного параметра (Boolean).

Висновки

У ході виконання курсової роботи було спроектовано та реалізовано комплексну систему бізнес-аналітики для предметної області «Конференції». Основною метою роботи було створення повного циклу обробки даних: від проектування реляційного сховища до візуалізації результатів у вигляді інтерактивних звітів. В результаті було розроблено та імплементовано денормалізоване сховище даних на базі Microsoft SQL Server. Враховуючи необхідність аналізу кількох бізнес-процесів, зокрема проведення виступів та логістики обладнання, було обрано схему зірки (Star Schema). Це дозволило об'єднати дві таблиці фактів через спільні виміри, що забезпечило цілісність моделі та можливість перехресного аналізу діяльності організації.

Для наповнення сховища даними засобами SQL Server Integration Services (SSIS) було створено автоматизований пакет ETL. У рамках цього процесу реалізовано очищення та підготовку даних, керування потоком виконання із дотриманням обмежень посилованої цілісності, а також трансформацію даних, включаючи генерацію сурогатних ключів та обробку бізнес-логіки. Для забезпечення надійності системи впроваджено механізми аудиту та циклічної перевірки цілісності завантажених даних. Наступним етапом стала побудова багатовимірного OLAP-куба засобами SQL Server Analysis Services (SSAS). Було спроектовано ієрархії вимірів, зокрема календарну ієрархію часу, та реалізовано розрахунок ключових показників. За допомогою мови MDX створено обчислювані члени для аналізу середніх величин, часових рядів та відносних показників росту, а також впроваджено KPI для візуального моніторингу ефективності.

На фінальному етапі у середовищі SQL Server Reporting Services (SSRS) розроблено комплект з п'яти звітів, що включають табличні та матричні форми, діаграми та інтерактивні дашборди. У процесі розробки та тестування підсистеми звітності було виявлено ряд технічних особливостей середовища,

які вплинули на фінальну реалізацію окремих функцій. Зокрема, при налаштуванні інтерактивного сортування у табличних звітах було встановлено, що дані вже надходять зі сховища попередньо відсортованими за алфавітом внаслідок налаштувань на етапі ETL та в ієрархіях куба. Через внутрішні обмеження рендерингу звіту повторне накладання інтерактивного сортування в окремих випадках конфліктувало з групуванням, тому для збереження стабільності звіту пріоритет було надано базовому сортуванню на рівні запиту, що повністю задовольняє потреби користувача.

Також певні архітектурні зміни торкнулися реалізації деталізації звітів. Спочатку було спроектовано структуру типу Master-Detail з використанням механізму вкладених звітів (Subreports). Однак, при практичній реалізації виникли технічні обмеження середовища розробки при передачі складних параметрів контексту між звітами. У зв'язку з цим було прийнято рішення реалізувати функціонал деталізації через альтернативні механізми Drill-through та Drill-down. Цей підхід є еквівалентною заміною для навігації, забезпечує кращу швидкодію системи та дозволяє користувачеві ефективно переходити від загальних показників до детальної інформації.

Окремо варто зазначити аспект розгортання розробленої системи. На завершальному етапі було виконано повну конфігурацію проекту для розгортання, включаючи налаштування цільового сервера у властивостях проекту. Однак, фізичне перенесення звітів на сервер не було здійснено через обмеження прав доступу адміністратора до локальної служби SSRS на робочій станції розробки. Проте, проект є повністю сконфігурованим та підготовленим до перенесення на промисловий сервер за наявності відповідних прав доступу.

Незважаючи на зазначені технічні нюанси, розроблена система повністю відповідає вимогам технічного завдання. Курсова робота дозволила закріпити практичні навички роботи зі стеком технологій Microsoft BI та засвоїти методологію проектування корпоративних сховищ даних. Система готова до

подальшого масштабування та інтеграції нових джерел даних, забезпечуючи керівництво актуальною аналітичною інформацією для прийняття управлінських рішень.

Список використаних джерел

- 1) Microsoft Learn Documentation: <https://learn.microsoft.com/en-us/sql/>
- 2) Analysis Services Multidimensional Tutorials:
<https://learn.microsoft.com/en-us/analysis-services/multidimensional-tutorial/>
- 3) Redgate SQL Data Generator Documentation: <https://documentation.red-gate.com/sdg>

Додаток А

Sql-скрипти для створення database and tables:

ConferenceDB:

```
CREATE DATABASE ConferenceDB;  
GO
```

```
USE ConferenceDB;  
GO
```

```
CREATE TABLE AcademicDegrees (  
    DegreeID INT IDENTITY(1,1) PRIMARY KEY,  
    DegreeName NVARCHAR(50) NOT NULL UNIQUE  
);
```

```
CREATE TABLE Equipment (  
    EquipmentID INT IDENTITY(1,1) PRIMARY KEY,  
    EquipmentName NVARCHAR(100) NOT NULL,  
    Description NVARCHAR(255) NULL  
);
```

```
CREATE TABLE Rooms (  
    RoomID INT IDENTITY(1,1) PRIMARY KEY,  
    RoomName NVARCHAR(50) NOT NULL,  
    Capacity INT NOT NULL CHECK (Capacity > 0),  
    Building NVARCHAR(100) NULL  
);
```

```
CREATE TABLE Speakers (  
    SpeakerID INT IDENTITY(1,1) PRIMARY KEY,  
    FirstName NVARCHAR(50) NOT NULL,  
    LastName NVARCHAR(50) NOT NULL,  
    Email NVARCHAR(100) NULL,  
    Workplace NVARCHAR(100) NULL,  
    Country NVARCHAR(50) NULL,  
    DegreeID INT NOT NULL,  
    CONSTRAINT FK_Speakers_Degrees FOREIGN KEY (DegreeID) REFERENCES  
AcademicDegrees(DegreeID)  
);
```

```
CREATE TABLE Conferences (  
    ConferenceID INT IDENTITY(1,1) PRIMARY KEY,  
    Title NVARCHAR(200) NOT NULL,  
    StartDate DATE NOT NULL,  
    EndDate DATE NOT NULL,  
    LocationCity NVARCHAR(100) NOT NULL,  
    Description NVARCHAR(MAX) NULL,  
    CONSTRAINT CK_Conference_Dates CHECK (EndDate >= StartDate)  
);
```

```
CREATE TABLE Sections (  
    SectionID INT IDENTITY(1,1) PRIMARY KEY,  
    ConferenceID INT NOT NULL,  
    Name NVARCHAR(150) NOT NULL,  
    RoomID INT NOT NULL,  
    CONSTRAINT FK_Sections_Conferences FOREIGN KEY (ConferenceID) REFERENCES  
Conferences(ConferenceID),  
    CONSTRAINT FK_Sections_Rooms FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID)  
);
```

```

CREATE TABLE Speeches (
    SpeechID INT IDENTITY(1,1) PRIMARY KEY,
    SectionID INT NOT NULL,
    SpeakerID INT NOT NULL,
    Title NVARCHAR(200) NOT NULL,
    StartTime DATETIME NOT NULL,
    DurationMinutes INT NOT NULL DEFAULT 45,
    Abstract NVARCHAR(MAX) NULL,
    CONSTRAINT FK_Speeches_Sections FOREIGN KEY (SectionID) REFERENCES
Sections(SectionID),
    CONSTRAINT FK_Speeches_Speakers FOREIGN KEY (SpeakerID) REFERENCES
Speakers(SpeakerID)
);

CREATE TABLE SpeechEquipment (
    SpeechID INT NOT NULL,
    EquipmentID INT NOT NULL,
    Quantity INT NOT NULL DEFAULT 1 CHECK (Quantity > 0),
    PRIMARY KEY (SpeechID, EquipmentID),
    CONSTRAINT FK_SpeechEquip_Speeches FOREIGN KEY (SpeechID) REFERENCES
Speeches(SpeechID),
    CONSTRAINT FK_SpeechEquip_Equipment FOREIGN KEY (EquipmentID) REFERENCES
Equipment(EquipmentID)
);
GO

```

ConferenceDW:

```

CREATE DATABASE ConferenceDW;
GO

USE ConferenceDW;
GO

CREATE TABLE DimDate (
    DateKey INT PRIMARY KEY,
    FullDate DATE NOT NULL,
    Year INT,
    Month INT,
    MonthName NVARCHAR(20),
    Day INT,
    DayOfWeek INT,
    DayName NVARCHAR(20),
    Quarter INT,
    IsWeekend BIT
);

CREATE TABLE DimSpeakers (
    SpeakerKey INT IDENTITY(1,1) PRIMARY KEY,
    SpeakerOriginalID INT,
    FullName NVARCHAR(150),
    AcademicDegree NVARCHAR(50),
    Country NVARCHAR(50),
    Email NVARCHAR(100),
    RowEffectiveDate DATETIME DEFAULT GETDATE(),
    RowExpirationDate DATETIME NULL,
    IsCurrent BIT DEFAULT 1
);

```

```

CREATE TABLE DimConferences (
    ConferenceKey INT IDENTITY(1,1) PRIMARY KEY,
    ConferenceOriginalID INT,
    Title NVARCHAR(200),
    LocationCity NVARCHAR(100),
    Description NVARCHAR(MAX)
);

CREATE TABLE DimEquipment (
    EquipmentKey INT IDENTITY(1,1) PRIMARY KEY,
    EquipmentOriginalID INT,
    EquipmentName NVARCHAR(100),
    Type NVARCHAR(50) DEFAULT 'General'
);

CREATE TABLE DimRooms (
    RoomKey INT IDENTITY(1,1) PRIMARY KEY,
    RoomOriginalID INT,
    RoomName NVARCHAR(50),
    Building NVARCHAR(100),
    CapacityBucket NVARCHAR(20)
);

CREATE TABLE FactSpeeches (
    FactSpeechID INT IDENTITY(1,1) PRIMARY KEY,
    DateKey INT NOT NULL,
    SpeakerKey INT NOT NULL,
    ConferenceKey INT NOT NULL,
    RoomKey INT NOT NULL,
    DurationMinutes INT,
    SpeechCount INT DEFAULT 1,

    -- Зовнішні ключі
    CONSTRAINT FK_FactSpeeches_Date FOREIGN KEY (DateKey) REFERENCES DimDate(DateKey),
    CONSTRAINT FK_FactSpeeches_Speaker FOREIGN KEY (SpeakerKey) REFERENCES
DimSpeakers(SpeakerKey),
    CONSTRAINT FK_FactSpeeches_Conference FOREIGN KEY (ConferenceKey) REFERENCES
DimConferences(ConferenceKey),
    CONSTRAINT FK_FactSpeeches_Room FOREIGN KEY (RoomKey) REFERENCES DimRooms(RoomKey)
);

CREATE TABLE FactEquipmentUsage (
    FactUsageID INT IDENTITY(1,1) PRIMARY KEY,
    DateKey INT NOT NULL,
    EquipmentKey INT NOT NULL,
    ConferenceKey INT NOT NULL,
    QuantityUsed INT,

    CONSTRAINT FK_FactEquip_Date FOREIGN KEY (DateKey) REFERENCES DimDate(DateKey),
    CONSTRAINT FK_FactEquip_Equip FOREIGN KEY (EquipmentKey) REFERENCES
DimEquipment(EquipmentKey),
    CONSTRAINT FK_FactEquip_Conf FOREIGN KEY (ConferenceKey) REFERENCES
DimConferences(ConferenceKey)
);

GO

```