

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»  
Варіант 1  
«Студент (Деканат)»

**Виконала:**  
студентка групи ПМ-41  
Рогос В. М.  
**Перевірив:**  
Любінський Б. Б.

Львів 2026

У курсовій роботі розглянуто процес проєктування, реалізації та аналізу надвеликої бази даних для інформаційної системи деканату закладу вищої освіти. Предметною областю дослідження є облік студентів, організація навчального процесу, аналіз успішності, користування бібліотекою та проживання студентів у гуртожитках.

У межах роботи виконано аналіз предметної області, побудовано концептуальну, логічну та фізичну моделі бази даних. На основі реляційної бази даних розроблено сховище даних із використанням багатовимірної моделі типу «зірка». Для аналітичної обробки інформації створено OLAP-куб у середовищі Microsoft SQL Server Analysis Services, що забезпечує багатовимірний аналіз даних за факультетами, групами, студентами, предметами та часовими періодами.

З метою візуалізації та аналізу даних розроблено набір аналітичних звітів у SQL Server Reporting Services, зокрема табличні, матричні, графічні звіти та дашборд з використанням параметрів, умовного форматування та інтерактивних елементів. Реалізовані звіти дозволяють формувати рейтинги студентів, аналізувати успішність, виявляти заборгованість по бібліотечних кни�ах та оцінювати процес проживання студентів у гуртожитках.

Результати курсової роботи підтверджують доцільність використання технологій Microsoft SQL Server для побудови масштабованих інформаційно-аналітичних систем та ефективного аналізу великих обсягів даних.

# ЗМІСТ

Курсова робота.....	1
ВСТУП .....	5
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	7
1.1 Загальна характеристика предметної області.....	7
1.2 Актуальність автоматизації обраної предметної області .....	7
1.3 Опис основних бізнес-процесів .....	8
1.3.1 Процес обліку студентів .....	8
1.3.2 Процес організації навчального процесу .....	8
1.3.3 Процес управління бібліотечним фондом.....	9
1.3.4 Процес заселення студентів у гуртожитки .....	9
1.4 Основні сутності предметної області .....	9
1.5 Інформаційні потоки та взаємозв'язки даних.....	10
1.6 Вимоги до інформаційної системи.....	10
1.7 Висновки до розділу .....	10
РОЗДІЛ 2. ПРОЕКТУВАННЯ БАЗИ ДАНИХ.....	11
2.1 Концептуальне проєктування бази даних .....	11
2.1.1 Мета концептуального проєктування.....	11
2.1.2 Вибір нотації ЕР-діаграми.....	12
2.1.3 Основні сутності предметної області .....	12
2.1.4 Зв'язки між сутностями та їх кардинальність.....	13
2.1.5 Узагальнення концептуальної моделі .....	14
2.2 Логічне проєктування бази даних .....	14
2.2.1 Перехід від концептуальної до логічної моделі .....	14
2.2.2 Нормалізація бази даних .....	15
2.2.3 Реалізація зв'язків типу «багато-до-багатьох» .....	15
2.2.4 Первінні та зовнішні ключі .....	16
2.3 Фізичне проєктування бази даних .....	16
2.3.1 Вибір типів даних .....	17
2.3.2 Проєктування індексів .....	17
2.3.3 Підготовка бази даних до генерації великих обсягів даних .....	18
2.4 Генерація та наповнення бази даних тестовими даними.....	18
2.4.1 Мета та завдання генерації даних.....	18
2.4.2 Вимоги до обсягів та реалістичності даних .....	19
2.4.3 Обґрунтування вибору інструменту генерації даних.....	19
2.4.4 Налаштування правил генерації даних.....	20
2.4.5 Забезпечення цілісності та перевірка згенерованих даних .....	21

2.5 Висновки до розділу .....	22
<b>РОЗДІЛ 3. РЕАЛІЗАЦІЯ ETL-ПРОЦЕСІВ .....</b>	<b>23</b>
3.1 Загальна характеристика сховища даних .....	23
3.2 Архітектура побудови сховища даних .....	23
3.3 Вибір моделі сховища даних .....	24
3.4 Таблиці фактів сховища даних.....	24
<b>3.4.1 Фактова таблиця успішності студентів.....</b>	<b>24</b>
<b>3.4.2 Фактова таблиця користування бібліотекою .....</b>	<b>25</b>
<b>3.4.3 Фактова таблиця поселення студентів.....</b>	<b>25</b>
3.5 Таблиці вимірів .....	26
3.6 Реалізація ETL-процесів .....	26
3.7 Контроль якості та надійності ETL.....	28
<b>РОЗДІЛ 4. ПОБУДОВА OLAP-КУБА ТА АНАЛІТИЧНІ ЗВІТИ.....</b>	<b>29</b>
4.1 Призначення OLAP-аналізу в інформаційній системі деканату .....	29
<b>4.2 Архітектура сховища даних .....</b>	<b>29</b>
<b>4.2.1 Таблиці фактів .....</b>	<b>30</b>
<b>4.2.2 Взаємозв'язок таблиць фактів .....</b>	<b>31</b>
4.3 Створення проекту SSAS .....	31
4.4 Проектування вимірів (Dimensions).....	32
<b>4.4.1 Часовий вимір (DimTime) .....</b>	<b>32</b>
<b>4.4.2 Вимір студентів (DimStudent).....</b>	<b>32</b>
<b>4.5 Міри та групи мір .....</b>	<b>33</b>
4.7 Призначення аналітичних звітів у системі .....	34
4.8 Створення проекту SQL Server Reporting Services .....	34
<b>4.8.1 Створення проекту звітності.....</b>	<b>34</b>
<b>4.8.2 Налаштування підключення до OLAP-куба .....</b>	<b>35</b>
4.9 Загальна характеристика розроблених аналітичних звітів .....	35
4.10 Табличний звіт «Список студентів по групах» .....	36
4.11 Матричний звіт (Matrix Report).....	37
4.12 Звіти з діаграмами (Chart Reports) .....	38
4.13 Інтерактивний дашборд.....	39
4.14 Параметризовані звіти .....	40
<b>ВИСНОВКИ .....</b>	<b>42</b>

## **ВСТУП**

Розвиток інформаційних технологій суттєво змінив підходи до зберігання та обробки даних у різних сферах діяльності, зокрема в системі вищої освіти. Навчальні заклади щороку накопичують великі масиви інформації, пов'язані з освітнім процесом, контингентом студентів, результатами навчання, роботою викладачів, функціонуванням бібліотек і гуртожитків. Зі збільшенням обсягів таких даних виникає потреба не лише у їх збереженні, а й у швидкому аналізі та узагальненні для підтримки управлінських рішень.

Традиційні реляційні бази даних добре підходять для обробки транзакцій, однак при роботі з багаторічною історією та великими обсягами інформації їх можливостей стає недостатньо. У таких умовах доцільним є використання підходів, орієнтованих на аналітичу обробку даних, зокрема сховищ даних та OLAP-технологій. Вони дозволяють здійснювати багатовимірний аналіз, агрегувати показники за різними вимірами та отримувати аналітичні звіти у зручній для користувача формі.

Деканат закладу вищої освіти є прикладом підрозділу, діяльність якого тісно пов'язана з аналізом великої кількості різномірних даних. До його функцій належить ведення обліку студентів, контроль навчального процесу, аналіз успішності, формування рейтингів, а також опрацювання інформації про користування бібліотекою та проживання студентів у гуртожитках. Наявність єдиної інформаційно-аналітичної системи значно підвищує ефективність виконання цих завдань.

У межах даної курсової роботи розглядається предметна область «Студент (Деканат)», яка характеризується складною структурою даних та великою кількістю взаємопов'язаних об'єктів. Для її реалізації використовується платформа Microsoft SQL Server, що надає повний набір інструментів для роботи з надвеликими базами даних: проєктування реляційної бази, реалізацію ETL-процесів, побудову OLAP-куба та створення аналітичних звітів.

Метою курсової роботи є розробка та аналіз надвеликих баз даних для підтримки діяльності деканату закладу вищої освіти з використанням сучасних технологій Microsoft SQL Server.

Для досягнення поставленої мети у роботі необхідно виконати такі основні завдання:

- дослідити предметну область та визначити ключові бізнес-процеси;
- спроєктувати реляційну базу даних з урахуванням вимог цілісності та масштабованості;
- згенерувати великі обсяги тестових даних, що імітують реальну діяльність деканату;
- реалізувати ETL-процеси для формування сховища даних;
- побудувати OLAP-куб для багатовимірного аналізу інформації;
- створити аналітичні звіти та дашборди для візуалізації результатів.

Об'ектом дослідження є інформаційні процеси, пов'язані з діяльністю деканату закладу вищої освіти. Предметом дослідження є методи та засоби проєктування, обробки та аналітичного аналізу надвеликих обсягів даних.

Практична цінність курсової роботи полягає у можливості використання отриманих результатів для аналізу навчальної діяльності студентів, контролю бібліотечної заборгованості та дослідження процесів проживання у гуртожитках. Запропоновані рішення можуть бути адаптовані для впровадження в реальних інформаційних системах закладів вищої освіти.

## **РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

### **1.1 Загальна характеристика предметної області**

Сучасні заклади вищої освіти функціонують як складні організаційні системи, що щоденно опрацьовують значні обсяги різноманітної інформації. До такої інформації належать персональні дані студентів, результати навчальної діяльності, дані про викладачів, навчальні дисципліни, бібліотечні ресурси, а також інформація про проживання студентів у гуртожитках. Ефективне управління цими даними є неможливим без використання сучасних інформаційних систем та баз даних.

Предметною областю даної курсової роботи є діяльність деканату закладу вищої освіти, яка охоплює процеси обліку студентів, організації навчального процесу, контролю успішності, управління бібліотечним фондом та заселення студентів у гуртожитки. Вибір саме цієї предметної області зумовлений її практичною значущістю, високим рівнем складності та наявністю великої кількості взаємопов'язаних сутностей.

У традиційних підходах значна частина облікових операцій виконується у розрізних системах або навіть вручну, що призводить до дублювання даних, помилок та ускладнює аналітичну обробку інформації. Тому актуальним є створення централізованої бази даних, яка забезпечить цілісне зберігання інформації та стане основою для побудови аналітичних звітів і підтримки управлінських рішень.

### **1.2 Актуальність автоматизації обраної предметної області**

Автоматизація процесів деканату є важливою складовою цифрової трансформації освіти. В умовах зростання кількості студентів та збільшення обсягів навчальної інформації виникає потреба у швидкому доступі до достовірних даних, можливості аналізу успішності, контролю академічної заборгованості та ефективного управління ресурсами навчального закладу.

Особливу актуальність має зберігання історичних даних за кілька років, оскільки це дозволяє аналізувати динаміку навчальних результатів, оцінювати ефективність викладання окремих дисциплін та приймати обґрунтовані

управлінські рішення. Крім того, інтеграція даних про бібліотечні операції та заселення у гуртожитки дає змогу формувати комплексні звіти про соціальне забезпечення студентів.

Використання баз даних та аналітичних систем у даній предметній області дозволяє:

- підвищити якість обліку та контролю даних;
- мінімізувати вплив людського фактору;
- забезпечити швидке формування звітності;
- створити передумови для впровадження OLAP-аналізу.

### **1.3 Опис основних бізнес-процесів**

Діяльність деканату включає низку взаємопов'язаних бізнес-процесів, кожен з яких оперує власним набором даних, але при цьому тісно інтегрований з іншими процесами.

#### **1.3.1 Процес обліку студентів**

Процес обліку студентів передбачає зберігання персональних даних студентів, інформації про рік вступу, навчальну групу, факультет та поточний статус. Дані про студентів використовуються практично у всіх інших бізнес-процесах, що робить цю інформацію базовою для всієї системи.

Кожен студент належить до певної навчальної групи та факультету, що дозволяє здійснювати агрегацію даних за структурними підрозділами закладу освіти.

#### **1.3.2 Процес організації навчального процесу**

Навчальний процес охоплює дисципліни, викладачів та результати навчання студентів. Для кожного студента зберігається інформація про оцінки з предметів за семестрами та навчальними роками. Це дозволяє аналізувати успішність як окремих студентів, так і цілих груп або факультетів.

Особливістю даного процесу є необхідність зберігання великої кількості записів з оцінками та підтримка часової історії, що є важливим для подальшого аналізу.

### **1.3.3 Процес управління бібліотечним фондом**

Бібліотечний процес включає облік книг, авторів та операцій видачі літератури студентам. Дляожної операції фіксується дата видачі та дата повернення книги. Це дозволяє визначати заборгованість студентів, аналізувати популярність літератури та формувати відповідні звіти.

Зберігання історії видачі книг є важливим для аналізу використання бібліотечного фонду та оптимізації його поповнення.

### **1.3.4 Процес заселення студентів у гуртожитки**

Процес поселення студентів у гуртожитки передбачає зберігання інформації про гуртожитки, кімнати та періоди проживання студентів. Оскільки студент може проживати в різних кімнатах у різні часові періоди, виникає потреба у зберіганні історичних даних.

Цей процес є прикладом бізнес-процесу з чітко вираженою часовою залежністю, що важливо враховувати при проєктуванні бази даних.

## **1.4 Основні сутності предметної області**

На основі аналізу бізнес-процесів було виділено основні сутності предметної області, які відображають реальні об'єкти та події:

- студенти;
- факультети;
- навчальні групи;
- предмети;
- викладачі;
- оцінки;
- книги та операції їх видачі;
- гуртожитки та кімнати;
- поселення студентів;
- хобі студентів.

Кожна з перелічених сутностей має власний набір атрибутів та взаємодіє з іншими сутностями через логічні зв'язки.

## **1.5 Інформаційні потоки та взаємозв'язки даних**

Інформаційні потоки в системі формуються в результаті взаємодії бізнес-процесів. Наприклад, дані про студентів використовуються одночасно у процесах обліку успішності, бібліотечних операціях та заселенні у гуртожитки. Це зумовлює необхідність централізованого зберігання інформації та уникнення дублювання даних.

Взаємозв'язки між даними мають як ієрархічний характер (факультет – група – студент), так і асоціативний (студент – предмет – оцінка). Така структура ускладнює обробку даних, але водночас робить систему гнучкою для аналітичних запитів.

## **1.6 Вимоги до інформаційної системи**

На основі аналізу предметної області сформульовано такі вимоги до інформаційної системи:

- підтримка великих обсягів даних;
- забезпечення цілісності та узгодженості інформації;
- можливість зберігання історичних даних;
- підтримка складних аналітичних запитів;
- інтеграція з інструментами бізнес-аналітики.

Система повинна бути орієнтована не лише на оперативну роботу, але й на аналітичну обробку інформації.

## **1.7 Висновки до розділу**

У першому розділі було виконано детальний аналіз предметної області «Студент (Деканат)». Розглянуто основні бізнес-процеси, інформаційні потоки та сутності, що беруть участь у функціонуванні системи. Визначено актуальність автоматизації обраної предметної області та сформульовано вимоги до інформаційної системи. Отримані результати стали основою для подальшого проєктування бази даних та реалізації аналітичних компонентів системи.

## **РОЗДІЛ 2. ПРОЕКТУВАННЯ БАЗИ ДАНИХ**

Проектування бази даних є ключовим етапом створення інформаційної системи, оскільки саме на цьому етапі визначається структура зберігання даних, логіка взаємозв'язків між ними та забезпечується основа для подальшої реалізації системи. Якісно спроектована база даних дозволяє не лише коректно зберігати інформацію, але й ефективно виконувати аналітичні запити, що є особливо важливим у системах, орієнтованих на обробку великих обсягів даних.

У межах даної курсової роботи проектування бази даних виконується для предметної області «Студент (Деканат)». Ця область характеризується значною кількістю сутностей, складними взаємозв'язками між ними, а також необхідністю зберігання історичних даних щодо навчального процесу, проживання студентів та користування бібліотекою. Проектування бази даних здійснюється поетапно та включає концептуальне і логічне моделювання.

### **2.1 Концептуальне проектування бази даних**

Концептуальне проектування є першим і найбільш узагальненим рівнем моделювання бази даних. На цьому етапі формується абстрактна модель предметної області без прив'язки до конкретної системи управління базами даних та фізичних аспектів зберігання інформації.

Основною метою концептуального проектування є визначення:

- основних сутностей предметної області;
- атрибутів, що характеризують кожну сутність;
- логічних зв'язків між сутностями;
- бізнес-правил та обмежень, які повинні виконуватися у системі.

Результатом концептуального проектування є ER-діаграма, яка наочно відображає структуру предметної області та слугує основою для подальших етапів проектування бази даних.

#### **2.1.1 Мета концептуального проектування**

Метою концептуального проектування у межах даної курсової роботи є створення узагальненої інформаційної моделі предметної області деканату, яка

відображає реальні об'єкти та процеси навчального закладу. Така модель дозволяє систематизувати інформацію про студентів, навчальні групи, факультети, результати навчання, проживання у гуртожитках та користування бібліотекою.

Концептуальна модель не визнає технічних обмежень СУБД, що дозволяє зосередитися виключно на логіці предметної області. Це значно спрощує узгодження вимог між розробником та кінцевими користувачами системи, а також дозволяє виявити потенційні логічні помилки ще до етапу реалізації.

### **2.1.2 Вибір нотації ER-діаграми**

Для побудови концептуальної моделі бази даних було обрано нотацію **Crow's Foot**, яка є однією з найпоширеніших нотацій ER-моделювання. Її використання зумовлене низкою переваг.

По-перше, нотація Crow's Foot забезпечує чітке та наочне відображення кардинальності зв'язків між сущностями, зокрема типів зв'язків 1:1, 1:N та M:N. По-друге, вона є інтуїтивно зрозумілою, що спрощує аналіз моделі не лише для розробників, але й для фахівців предметної області. По-третє, дана нотація широко використовується у практиці проєктування реляційних баз даних та легко трансформується у логічну модель.

Застосування нотації Crow's Foot дозволяє забезпечити відповідність концептуальної моделі вимогам реляційної СУБД Microsoft SQL Server, яка використовується у подальших етапах реалізації проєкту.

### **2.1.3 Основні сутності предметної області**

На основі аналізу предметної області «Деканат» було визначено перелік основних сущностей, які відображають ключові об'єкти та процеси навчального закладу.

Центральною сутністю системи є Студент (Student), оскільки більшість бізнес-процесів безпосередньо пов'язані саме з обліком студентів та аналізом їх навчальної діяльності. Сутність Student містить персональні та навчальні характеристики студентів, необхідні для формування звітності та аналітичного аналізу.

Сутність Факультет (Faculty) використовується для представлення організаційної структури університету. Вона є базовою для групування студентів та навчальних груп.

Сутність Навчальна група (Group) дозволяє об'єднувати студентів за роком та напрямом навчання, що є важливим для контролю навчального процесу та формування аналітичних зрізів.

Сутності Предмет (Subject) та Викладач (Teacher) описують навчальний процес з точки зору дисциплін та викладацького складу. Вони використовуються для аналізу результатів навчання та ефективності викладання.

Сутність Оцінка (Grade) відображає результати навчальної діяльності студентів і є основою для формування показників успішності, середніх балів та рейтингів.

Окрім навчального процесу, у системі враховано допоміжні сутності: Книга, Видача книги, Гуртожиток, Кімната, Поселення та Хобі, що дозволяє вести комплексний облік діяльності студентів.

#### **2.1.4 Зв'язки між сутностями та їх кардинальність**

Між сутностями предметної області встановлено логічні зв'язки, що відображають реальні взаємодії між об'єктами. Один факультет може мати декілька навчальних груп, а кожна група — багатьох студентів. Студент може отримувати багато оцінок з різних предметів у різні навчальні періоди.

Для реалізації зв'язків типу «багато-до-багатьох» використовуються проміжні сутності. Так, зв'язок між студентами та книгами реалізовано через сутність BookLoan, що дозволяє зберігати історію видачі книг. Аналогічно, зв'язок між студентами та хобі реалізовано через таблицю StudentHobby, а між викладачами та предметами — через TeacherSubject.

Особливу роль відіграє сутність Settlement, яка забезпечує зберігання історії проживання студентів у гуртожитках з урахуванням часових меж.

## **2.1.5 Узагальнення концептуальної моделі**

Побудована концептуальна модель бази даних у вигляді ER-діаграми повністю відображає структуру предметної області «Студент (Деканат)» та основні бізнес-процеси, що в ній відбуваються. Запропонована модель є логічно узгодженою, масштабованою та придатною для подальшого розвитку системи. Концептуальна модель створює надійну основу для переходу до наступного етапу — логічного проєктування бази даних, під час якого вона буде трансформована у реляційну структуру з визначенням ключів, обмежень та нормалізації.

## **2.2 Логічне проєктування бази даних**

Логічне проєктування бази даних є наступним етапом після концептуального моделювання та полягає у трансформації ER-моделі предметної області у логічну реляційну модель. На цьому етапі визначається структура таблиць, їх атрибути, первинні та зовнішні ключі, а також забезпечується дотримання принципів нормалізації.

На відміну від концептуального рівня, логічна модель вже орієтована на реляційну модель даних, однак все ще не враховує фізичних особливостей конкретної СУБД. Основною метою логічного проєктування є створення такої структури бази даних, яка забезпечує цілісність, узгодженість та мінімізацію надлишковості інформації.

### **2.2.1 Перехід від концептуальної до логічної моделі**

Перехід від концептуальної ER-діаграми до логічної моделі здійснюється шляхом відображенняожної сутності у вигляді окремої таблиці реляційної бази даних. Атрибути сутностей переносяться у вигляді колонок таблиць, а зв'язки між сутностями реалізуються за допомогою зовнішніх ключів.

Сутності предметної області «Студент (Деканат)» мають чітко визначені ідентифікатори, що дозволяє однозначно ідентифікувати кожен запис. Для спрощення реалізації та підвищення продуктивності в логічній моделі використано сурогатні первинні ключі числового типу.

Особливу увагу було приділено збереженню ієрархічних зв'язків, зокрема:

- факультет → група → студент;
- гуртожиток → кімната → поселення;
- навчальний процес → оцінки → часові періоди.

## **2.2.2 Нормалізація бази даних**

Логічна модель бази даних була приведена до третьої нормальної форми (ЗНФ), що дозволяє уникнути аномалій вставки, оновлення та видалення даних.

### **Перша нормальна форма (1НФ)**

База даних відповідає першій нормальній формі, оскільки всі атрибути є атомарними, а кожне поле таблиці містить лише одне значення. Наприклад, інформація про хобі студентів не зберігається у вигляді списку в таблиці Student, а винесена в окрему таблицю StudentHobby.

### **Друга нормальна форма (2НФ)**

Друга нормальна форма забезпечена за рахунок того, що всі неключові атрибути повністю залежать від первинного ключа таблиці. Довідникові дані, такі як факультети, предмети та викладачі, винесені в окремі таблиці, що усуває часткові залежності.

### **Третя нормальна форма (3НФ)**

Третя нормальна форма досягнута шляхом усунення транзитивних залежностей. Наприклад, назва факультету не дублюється в таблиці Student, а отримується через зв'язок із таблицею Faculty. Це забезпечує цілісність даних та спрощує їх оновлення.

## **2.2.3 Реалізація зв'язків типу «багато-до-багатьох»**

У логічній моделі всі зв'язки типу М:N були розв'язані шляхом створення окремих таблиць зв'язку. Такий підхід дозволяє не лише коректно відобразити кардинальність зв'язків, але й зберігати додаткові атрибути.

У межах проєкту було створено такі таблиці зв'язку:

- **StudentHobby** — для зв'язку студентів з хобі;
- **TeacherSubject** — для відображення викладання предметів різними викладачами;

- **BookLoan** — для реалізації зв'язку між студентами та книгами з урахуванням дат видачі та повернення;
- **Settlement** — для зберігання історії проживання студентів у гуртожитках. Завдяки цьому база даних підтримує історичність даних та дозволяє виконувати аналітичні запити за різні періоди часу.

#### **2.2.4 Первинні та зовнішні ключі**

Для кожної таблиці логічної моделі визначено первинний ключ, який забезпечує унікальність записів. Як первинні ключі використовуються ціличисельні поля з автоінкрементом, що спрощує генерацію даних та підвищує продуктивність запитів.

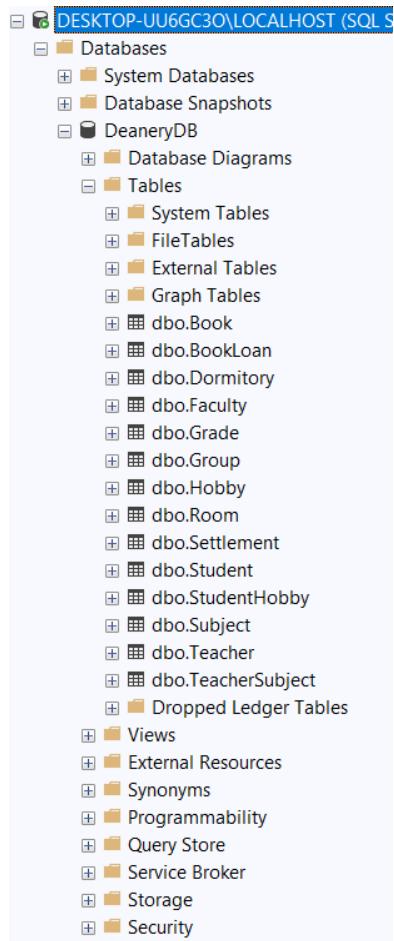
Зовнішні ключі використовуються для реалізації зв'язків між таблицями та забезпечення референтної цілісності. Усі зовнішні ключі відповідають логіці предметної області та узгоджені з концептуальною моделлю.

### **2.3 Фізичне проєктування бази даних**

Фізичне проєктування бази даних є завершальним етапом проєктування та полягає у безпосередній реалізації логічної моделі в конкретній системі управління базами даних. У межах даної курсової роботи фізична модель реалізується у СУБД Microsoft SQL Server.

На цьому етапі визначаються:

- типи даних колонок;
- механізми автоінкременту;
- індекси;
- обмеження цілісності;
- підготовка структури для подальшої генерації великих обсягів даних.



### 2.3.1 Вибір типів даних

Типи даних було підібрано з урахуванням обсягів інформації та вимог до продуктивності. Для ідентифікаторів використовуються типи INT та BIGINT, для текстових полів — NVARCHAR, для дат — DATE.

Такий вибір забезпечує:

- економне використання пам'яті;
- коректну обробку національних символів;
- підтримку великих обсягів даних.

### 2.3.2 Проектування індексів

Для підвищення швидкодії запитів у фізичній моделі було передбачено:

- кластерні індекси на первинні ключі;
- некластерні індекси на зовнішні ключі;
- індекси на поля, які часто використовуються у фільтрації та аналітичних запитах.

Особливу увагу приділено таблицям фактів, які у подальшому використовуються для побудови сховища даних та OLAP-куба.

### **2.3.3 Підготовка бази даних до генерації великих обсягів даних**

Фізична структура бази даних була спеціально спроектована з урахуванням подальшої генерації великих обсягів тестових даних. Для цього:

- використано автоінкрементні первинні ключі;
- забезпечено коректні зовнішні ключі між таблицями;
- уникнуто складних обмежень, що можуть ускладнювати масову вставку даних.

Це дозволяє безпечно та ефективно наповнювати базу даних сотнями тисяч і мільйонами записів без порушення цілісності.

### **2.4 Генерація та наповнення бази даних тестовими даними**

Після завершення етапів концептуального, логічного та фізичного проєктування база даних потребує наповнення інформацією для перевірки коректності структури, тестування запитів та подальшої аналітичної обробки.

Використання реальних даних у межах навчального проєкту є неможливим з огляду на вимоги конфіденційності та обмежений доступ до інформації. Тому виникає необхідність у генерації тестових, але водночас реалістичних даних.

Генерація даних дозволяє змоделювати реальну діяльність навчального закладу протягом тривалого періоду часу, перевірити працездатність зв'язків між таблицями та підготувати базу даних до побудови сховища даних і формування аналітичних звітів.

#### **2.4.1 Мета та завдання генерації даних**

Метою генерації даних у межах даного проєкту є створення великого обсягу структурованих даних, які відповідають бізнес-логіці предметної області «Студент (Деканат)» та забезпечують можливість виконання аналітичних операцій.

Основними завданнями генерації даних є:

- перевірка коректності логічної та фізичної структури бази даних;
- тестування обмежень цілісності та зовнішніх ключів;

- підготовка даних для реалізації ETL-процесів;
- забезпечення достатніх обсягів інформації для побудови OLAP-куба;
- формування основи для створення аналітичних звітів у SSRS.

Згенеровані дані повинні бути узгодженими між собою та відповідати реальним сценаріям функціонування інформаційної системи деканату.

#### **2.4.2 Вимоги до обсягів та реалістичності даних**

Обсяги згенерованих даних були визначені з урахуванням необхідності виконання аналітичних запитів над великими масивами інформації. Занадто малі обсяги даних не дозволяють повною мірою оцінити ефективність проектних рішень, тоді як великі обсяги забезпечують наближення до реальних умов експлуатації системи.

У межах даної курсової роботи було встановлено такі орієнтовні вимоги до обсягів даних:

- кількість студентів — не менше 10 000 записів;
- кількість оцінок — не менше 500 000 записів;
- період навчальної діяльності — не менше 5 років;
- таблиці фактів (Grade, BookLoan, Settlement) — сотні тисяч записів;
- таблиці довідників повинні містити достатню кількість унікальних значень.
- Окрім кількісних характеристик, важливою вимогою є реалістичність даних, зокрема:
  - оцінки повинні відповідати прийнятій шкалі оцінювання;
  - дата повернення книги не може бути меншою за дату видачі;
  - академічний рік і семестр мають відповідати даті оцінювання;
  - студент не може бути поселений у двох кімнатах одночасно.

#### **2.4.3 Обґрунтування вибору інструменту генерації даних**

Для автоматизованої генерації тестових даних було використано програмний продукт Redgate SQL Data Generator, який інтегрується з середовищем SQL Server Management Studio та дозволяє створювати великі обсяги даних з урахуванням структури бази даних і зв'язків між таблицями.

Вибір даного інструменту обумовлений такими перевагами:

- підтримка зовнішніх ключів та обмежень цілісності;
- можливість налаштування складних правил генерації;
- генерація даних відповідно до типів колонок;
- висока продуктивність при роботі з великими обсягами;
- зручний графічний інтерфейс.

У порівнянні з ручною генерацією SQL-скриптів або онлайн-сервісами, Redgate SQL Data Generator забезпечує більш високий рівень контролю якості та узгодженості даних.

#### **2.4.4 Налаштування правил генерації даних**

Процес генерації даних здійснюється з урахуванням логічної структури бази даних та бізнес-правил предметної області.

Для основних таблиць були застосовані такі правила генерації:

##### **Таблиця Student**

- імена та прізвища — випадкові значення з вбудованих словників;
- дата народження — у діапазоні, що відповідає типовому віку студентів;
- рік вступу — значення за останні п'ять років;
- статус студента — обмежений набір допустимих значень;
- зовнішні ключі — вибір існуючих значень із довідників таблиць.

##### **Таблиця Grade**

- значення оцінки — у допустимому діапазоні;
- дата оцінювання — узгоджена з навчальним семестром;
- академічний рік — автоматично пов'язаний із датою;
- забезпечення зв'язку зі студентом, предметом та викладачем.

##### **Таблиця BookLoan**

- дата видачі — випадкова дата в межах визначеного періоду;
- дата повернення — або коректна дата після видачі, або NULL;
- забезпечення логічної послідовності дат.

##### **Таблиця Settlement**

- періоди проживання студентів у гуртожитках;

- недопущення перетину періодів поселення для одного студента;
- коректний зв'язок із таблицями Room та Dormitory.

The screenshot shows the SQL Data Generator application. In the left pane, under 'Tables to populate', many tables from the 'DeaneryDB' database are listed with checkboxes. Several checkboxes are checked, including [dbo].[Bookloan], [dbo].[Book], [dbo].[Dormitory], [dbo].[Faculty], [dbo].[Grade], [dbo].[Group], [dbo].[Hobby], [dbo].[Room], [dbo].[Settlement], [dbo].[StudentHobby], [dbo].[Student], [dbo].[Subject], [dbo].[TeacherSubject], and [dbo].[Teacher]. In the right pane, under 'Table generation settings' for the [dbo].[Bookloan] table, the 'Source of data' dropdown is set to 'Generate data'. The 'Number of rows to generate' is set to 1,000. A preview table at the bottom shows the first 100 rows of generated data, which includes columns: LoanID, BookID, StudentID, LoanDate, and ReturnDate. The preview table has 14 rows of sample data.

LoanID	BookID	StudentID	LoanDate	ReturnDate
1			05.07.1971	14.02.2009
2			24.05.1964	09.07.1985
3			08.04.1998	05.08.1976
4			06.01.2000	20.05.2023
5			30.06.2016	02.01.1977
6			27.09.1956	30.08.2019
7			04.05.1957	07.08.1989
8			22.05.2001	06.07.1965
9			29.06.1955	26.05.1964
10			22.08.2023	10.03.1978
11			27.10.2018	13.11.1999
12			19.10.1971	04.01.1956
13			08.06.1999	06.12.1976
14			24.07.2008	08.01.1974

#### 2.4.5 Забезпечення цілісності та перевірка згенерованих даних

Після завершення генерації даних було виконано перевірку їх коректності та узгодженості. Для цього здійснювалися:

- перевірка кількості записів у кожній таблиці;
- контроль виконання обмежень зовнішніх ключів;
- тестові SQL-запити для аналізу вибірок;
- перевірка логічних залежностей між датами та значеннями.

Завдяки використанню автоматизованого інструменту генерації всі записи були коректно інтегровані у структуру бази даних без порушення цілісності.

(No column name)
1 20000
(No column name)
1 19996
(No column name)
1 500000
(No column name)
1 180

## **2.5 Висновки до розділу**

У другому розділі курсової роботи було виконано повний цикл проєктування бази даних, який включає концептуальне, логічне та фізичне моделювання, а також процес генерації та наповнення бази даних тестовими даними. Спроєктована база даних є структуровано узгодженою, масштабованою та готовою до подальшого використання в аналітичних задачах.

Згенеровані дані забезпечують можливість реалізації ETL-процесів, побудови сховища даних, OLAP-куба та створення аналітичних звітів, що розглядається у наступному розділі курсової роботи.

## **РОЗДІЛ 3. РЕАЛІЗАЦІЯ ETL-ПРОЦЕСІВ**

### **3.1 Загальна характеристика сховища даних**

У процесі розробки інформаційних систем, орієнтованих на аналітичну обробку інформації, особливе значення має відокремлення транзакційних процесів від аналітичних. Традиційні реляційні бази даних, що використовуються для щоденної роботи деканату, бібліотеки чи адміністрації гуртожитків, призначені насамперед для швидкого збереження та оновлення даних. Водночас такі бази даних є малоекективними для виконання складних аналітичних запитів, які включають агрегацію великих обсягів історичних даних, багатовимірний аналіз та побудову звітів.

З метою розв'язання цієї проблеми у даній курсовій роботі було спроектовано та реалізовано сховище даних (Data Warehouse). Сховище даних являє собою окрему інформаційну систему, що акумулює дані з операційної бази даних, зберігає їх у структурованому та узгодженному вигляді і оптимізована для аналітичної обробки.

Основним призначенням сховища даних у межах предметної області «Студент (Деканат)» є забезпечення можливості комплексного аналізу:

- результатів навчальної діяльності студентів;
- користування бібліотечним фондом;
- заселення студентів у гуртожитки;
- динаміки змін показників у часовому розрізі.

Сховище даних створює інформаційну основу для побудови OLAP-куба та формування аналітичних звітів у системі SQL Server Reporting Services.

### **3.2 Архітектура побудови сховища даних**

Архітектура сховища даних у даному проекті базується на класичній трирівневій моделі, яка включає джерело даних, проміжний рівень обробки та аналітичний рівень.

На першому рівні знаходиться транзакційна база даних, що містить нормалізовані таблиці, розроблені на попередніх етапах курсової роботи. Ця

база даних використовується для зберігання поточних даних про студентів, оцінки, книги, поселення та інші об'єкти предметної області.

Другий рівень представлений ETL-процесами, які виконують витяг даних із транзакційної бази, їх очищенння, трансформацію та завантаження до сховища даних. Саме на цьому етапі відбувається підготовка даних до аналітичного використання.

Третій рівень складається зі сховища даних, OLAP-куба та інструментів аналітики й звітності. Така архітектура забезпечує масштабованість, гнучкість та високу продуктивність системи.

### **3.3 Вибір моделі сховища даних**

Для організації сховища даних у даному проекті було обрано зіркоподібну схему (Star Schema). Данна схема є однією з найбільш поширених у системах бізнес-аналітики завдяки своїй простоті, наочності та високій швидкодії аналітичних запитів.

Зіркоподібна схема передбачає наявність центральних таблиць фактів, які містять числові показники, та набору таблиць вимірів, що описують контекст цих показників. Така структура дозволяє легко виконувати операції агрегації, групування та фільтрації даних.

Вибір зіркоподібної схеми є обґрунтованим, оскільки предметна область курсової роботи передбачає аналіз даних за багатьма вимірами, зокрема за часом, факультетами, предметами, студентами, викладачами та іншими характеристиками.

### **3.4 Таблиці фактів сховища даних**

У межах даного проекту було реалізовано кілька таблиць фактів, кожна з яких відповідає окремому бізнес-процесу.

#### **3.4.1 Фактова таблиця успішності студентів**

Таблиця фактів, що відображає результати навчання студентів, призначена для зберігання інформації про отримані оцінки. Вона акумулює дані з

транзакційної таблиці Grade та містить посилання на відповідні таблиці вимірів.

Фактова таблиця дозволяє аналізувати:

- середній бал студентів;
- динаміку успішності за роками та семестрами;
- порівняння результатів навчання між факультетами та групами;
- ефективність викладання окремих предметів і викладачів.

Числовими показниками в цій таблиці є значення оцінок та кількість виставлених оцінок.

### **3.4.2 Фактова таблиця користування бібліотекою**

Фактова таблиця видачі книг відображає процес користування бібліотечним фондом студентами. Вона формується на основі таблиці BookLoan та дозволяє аналізувати активність студентів у бібліотеці.

Завдяки цій таблиці можна досліджувати:

- популярність окремих книг та авторів;
- кількість виданих книг у різні періоди часу;
- навантаження на бібліотечний фонд;
- поведінку студентів різних факультетів щодо користування бібліотекою.

### **3.4.3 Фактова таблиця поселення студентів**

Окрему роль у сховищі даних відіграє фактова таблиця поселення студентів (FactSettlement), яка відображає процес розміщення студентів у гуртожитках.

Ця таблиця дозволяє зберігати історію проживання студентів у різних кімнатах та гуртожитках.

На основі даних цієї таблиці можна виконувати аналіз:

- заповненості гуртожитків;
- тривалості проживання студентів;
- динаміки поселення за роками;
- ефективності використання житлового фонду.

### **3.5 Таблиці вимірів**

Таблиці вимірів у сховищі даних містять описову інформацію та використовуються для формування аналітичних звітів. Вони є денормалізованими порівняно з транзакційною базою даних, що дозволяє зменшити кількість з'єднань при виконанні запитів.

Особливе значення має часовий вимір, який використовується практично у всіх аналітичних запитах. Таблиця часу дозволяє аналізувати дані за роками, семестрами, місяцями та окремими датами.

Інші виміри описують студентів, факультети, групи, предмети, викладачів, книги, гуртожитки та кімнати, що забезпечує гнучкість і багатовимірність аналізу.

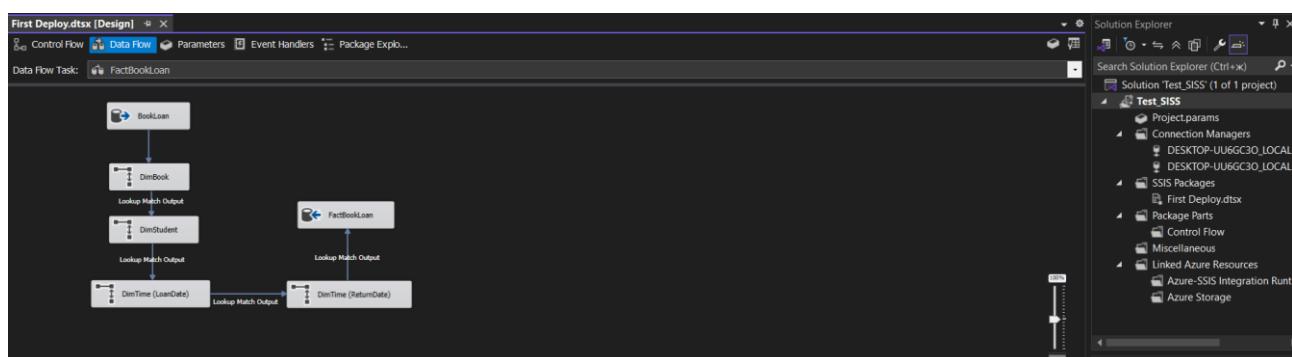
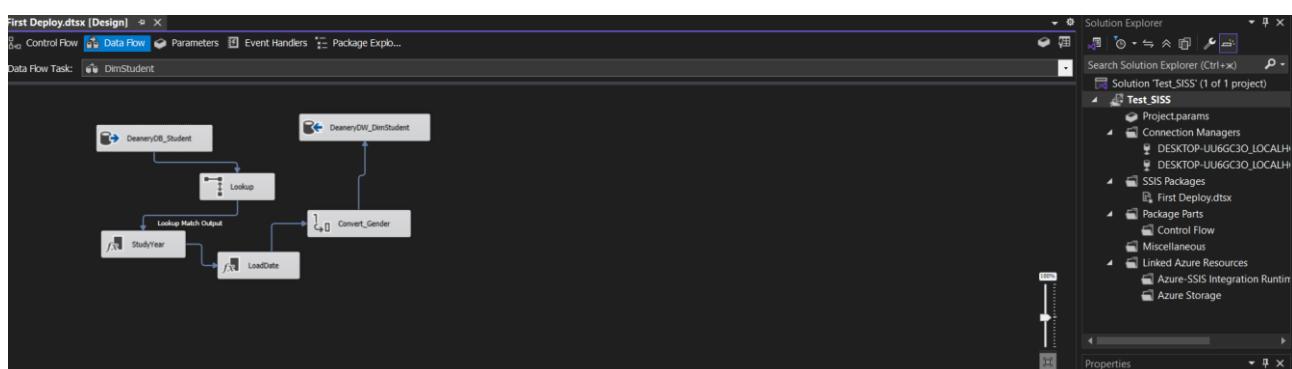
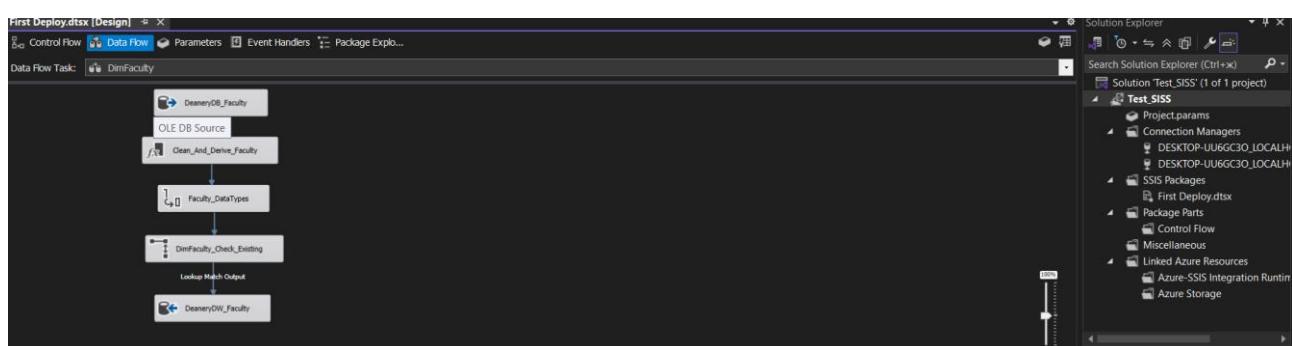
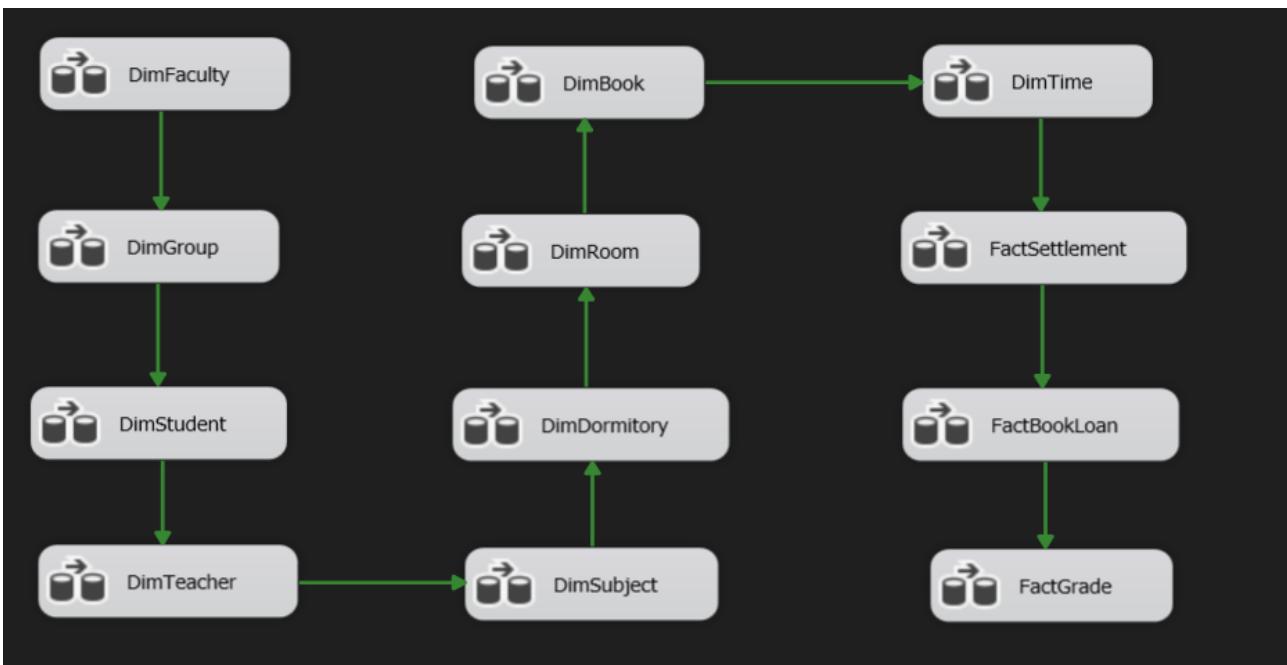
### **3.6 Реалізація ETL-процесів**

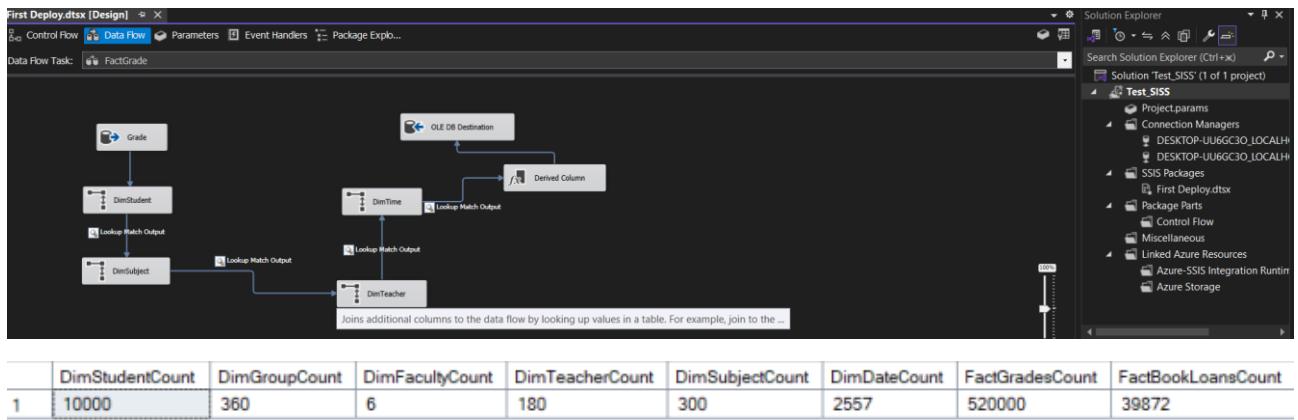
ETL-процеси є ключовим етапом побудови сховища даних, оскільки саме вони забезпечують перетворення операційних даних у аналітичні. У даній курсовій роботі ETL-процеси реалізовано за допомогою SQL Server Integration Services (SSIS).

Процес витягування даних передбачає отримання інформації з транзакційної бази даних. На цьому етапі виконуються SQL-запити до таблиць джерела з урахуванням необхідних атрибутів.

На етапі трансформації здійснюється очищення даних, приведення типів, обчислення похідних показників, формування бізнес-ключів та узгодження форматів дат. Особлива увага приділяється перевірці коректності значень та усуненню логічних помилок.

Завантаження даних виконується у таблиці вимірів і фактів сховища даних. Процес завантаження оптимізований для роботи з великими обсягами інформації та передбачає можливість повторного запуску у разі збоїв.





### 3.7 Контроль якості та надійності ETL

Для забезпечення надійності ETL-процесів було реалізовано механізми контролю якості даних. Вони включають перевірку кількості завантажених записів, контроль виконання обмежень цілісності та логування помилок.

Завдяки цьому забезпечується коректність даних у сховищі та можливість їх подальшого використання в аналітичних задачах.

## **РОЗДІЛ 4. ПОБУДОВА OLAP-КУБА ТА АНАЛІТИЧНІ ЗВІТИ**

### **4.1 Призначення OLAP-аналізу в інформаційній системі деканату**

У сучасних інформаційних системах, що обробляють великі обсяги історичних даних, традиційні реляційні бази даних не завжди забезпечують достатню продуктивність для аналітичних запитів. Особливо це стосується задач, пов'язаних з багатовимірним аналізом, агрегуванням даних за різними періодами часу та побудовою рейтингів.

У межах даної курсової роботи OLAP-аналіз використовується для підтримки прийняття управлінських рішень у навчальному закладі, зокрема для:

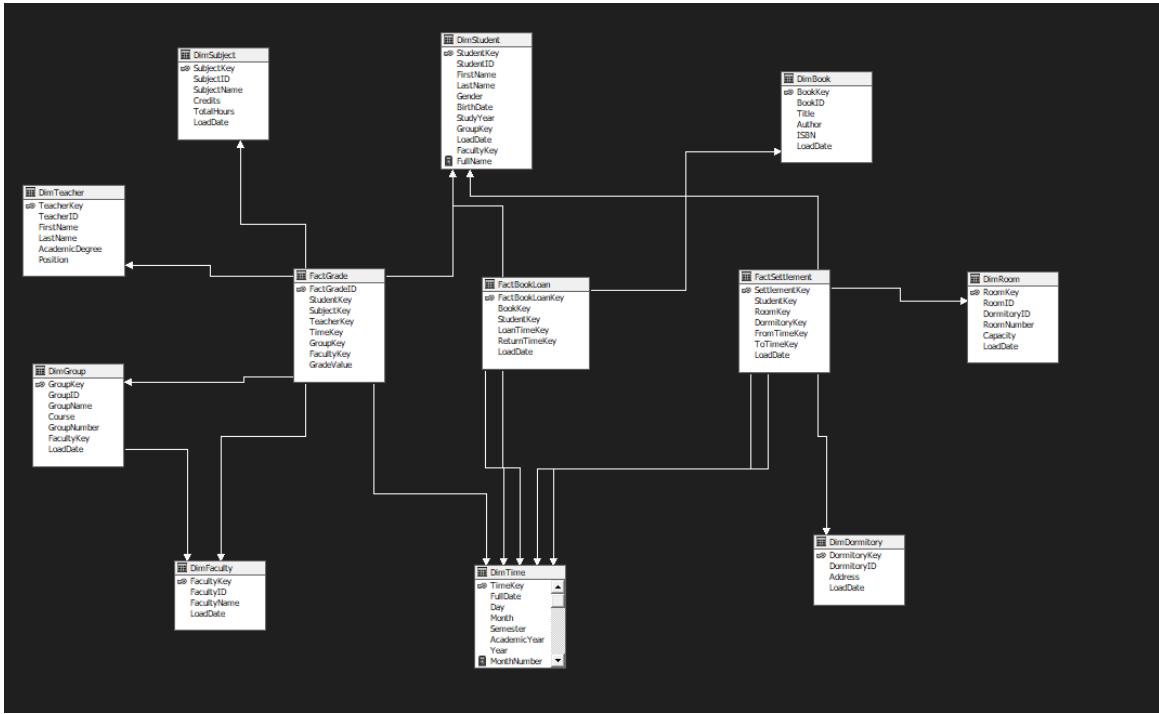
- аналізу успішності студентів;
- порівняння результатів між факультетами та групами;
- відстеження динаміки змін оцінок у часі;
- формування рейтингів студентів;
- аналізу додаткових показників, таких як користування бібліотекою або проживання в гуртожитку.

Для реалізації багатовимірного аналізу було використано SQL Server Analysis Services (SSAS) у режимі Multidimensional.

### **4.2 Архітектура сховища даних**

Сховище даних, створене в межах курсової роботи, побудоване за принципом багатовимірного аналізу та орієнтоване на зберігання історичних даних за кілька років. Для забезпечення гнучкого аналізу була використана схема типу “зірка”, у якій центральне місце займають таблиці фактів, пов’язані з таблицями вимірів.

У сховищі даних реалізовано три основні таблиці фактів, кожна з яких відповідає окремому бізнес-процесу предметної області деканату.



#### 4.2.1 Таблиці фактів

##### *FactGrade*

Таблиця FactGrade використовується для зберігання інформації про результати навчання студентів. Вона містить числові показники, що дозволяють аналізувати успішність у різних розрізах.

Основні показники:

- значення оцінки;
- кількість отриманих оцінок;
- семестр та навчальний рік.

Таблиця пов'язана з такими вимірами: студент, предмет, викладач, час, група та факультет.

##### *FactBookLoan*

Таблиця FactBookLoan відображає процес користування бібліотекою студентами та використовується для аналізу дисципліни повернення книг.

На її основі виконуються такі види аналізу:

- кількість виданих книг;
- тривалість користування книгами;
- виявлення прострочених повернень (більше року).

Таблиця фактів пов'язана з вимірами студентів, книг та часу.

### ***FactSettlement***

Таблиця FactSettlement відображає процес поселення студентів у гуртожитках та містить інформацію про періоди проживання.

Основні аналітичні задачі, які вирішуються за допомогою цієї таблиці:

- аналіз кількості студентів, що проживають у гуртожитках;
- аналіз заповнюваності кімнат;
- відстеження періодів проживання студентів у часі;
- визначення тривалості проживання.

Таблиця містить такі ключові дані:

- студент;
- кімната;
- гуртожиток;
- дата початку проживання;
- дата завершення проживання.

FactSettlement пов'язана з вимірами студентів, кімнат, гуртожитків та часу.

#### **4.2.2 Взаємозв'язок таблиць фактів**

Кожна таблиця фактів у сховищі даних відповідає окремому бізнес-процесу:

- навчальний процес (FactGrade),
- бібліотечне обслуговування (FactBookLoan),
- проживання у гуртожитку (FactSettlement).
- Такий підхід дозволяє:
  - аналізувати процеси незалежно один від одного;
  - виконувати комплексний аналіз студента за різними аспектами його діяльності;
  - підвищити масштабованість OLAP-куба.

### **4.3 Створення проекту SSAS**

Для побудови OLAP-куба було створено новий проект типу Analysis Services Multidimensional and Data Mining Project у середовищі SQL Server Data Tools (SSDT).

Основні етапи створення проєкту:

1. Налаштування підключення до сховища даних.
2. Створення Data Source, який визначає джерело даних для куба.
3. Створення Data Source View (DSV), що містить логічну модель таблиць.

Data Source View дозволяє візуально перевірити зв'язки між таблицями та створювати додаткові обчислення без зміни фізичної структури бази даних.

## 4.4 Проектування вимірів (Dimensions)

### 4.4.1 Часовий вимір (DimTime)

Часовий вимір є одним з ключових компонентів OLAP-куба. Він дозволяє аналізувати дані в розрізі років, семестрів, місяців та днів.

У вимірі DimTime було реалізовано ієрархію:

Рік → Місяць → День

Додатково були створені такі атрибути:

- номер семестру;
- день тижня;
- навчальний рік.

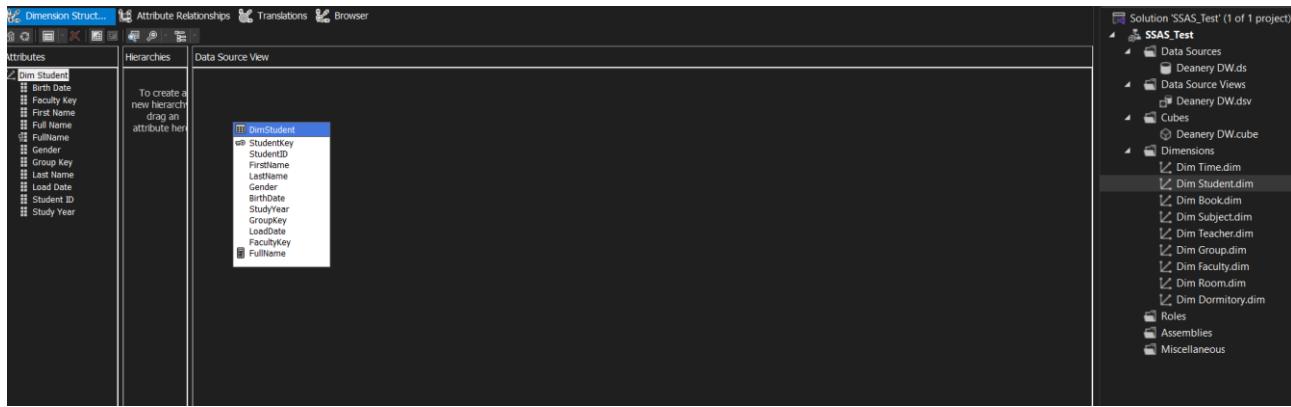
Це дозволяє виконувати часові обчислення, зокрема YTD, PY та YoY аналіз.

The screenshot displays the Microsoft Analysis Services Dimension Designer. On the left, the 'Attributes' pane lists attributes such as Dim Time, Academic Year, Day, Day Of Week, Day Of Week Number, Day Name, Is Weekend, Month, Month Number, Semester, Time Key, Week Number, and Year. In the center, the 'Hierarchies' pane shows a hierarchy for 'Year' with levels 'Year', 'Month', and 'Day'. A tooltip indicates: 'To create a new hierarchy, drag an attribute here.' Below this is a '<new level>' button. The 'Data Source View' pane on the right shows the DimTime table with columns: TimeKey, FullDate, Day, Month, Semester, AcademicYear, Year, MonthNumber, DayOfWeek, DayOfWeekNumber, WeekNumber, and IsWeekend. The navigation pane on the right shows the project structure under 'SSAS Test': Data Sources (Deanery DW.ds), Cubes (Deanery DW.cube), Dimensions (Dim Time.dim, Dim Student.dim, Dim Book.dim, Dim Subject.dim, Dim Teacher.dim, Dim Group.dim, Dim Faculty.dim, Dim Room.dim, Dim Dormitory.dim), Roles, Assemblies, and Miscellaneous.

### 4.4.2 Вимір студентів (DimStudent)

Вимір студентів містить персональні та навчальні характеристики студентів:

- ім'я та прізвище;
- рік вступу;
- статус студента;
- група та факультет.



## 4.5 Міри та групи мір

На основі таблиць фактів у кубі створено кілька груп мір:

### *Grade Measures*

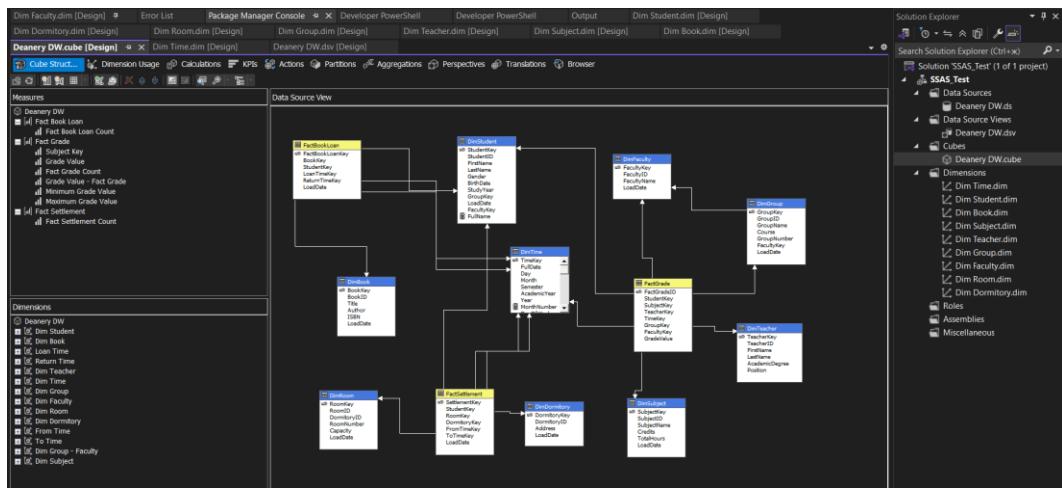
- Загальна сума оцінок;
- Кількість оцінок;
- Мінімальна та максимальна оцінка;
- Середня оцінка;
- Рейтинг студента.

### *Book Loan Measures*

- Кількість виданих книг;
- Кількість неповернених книг;
- Середня тривалість користування книгою.

### *Settlement Measures*

- Кількість поселень;
- Кількість студентів, що проживали у гуртожитку;
- Середня тривалість проживання;
- Кількість активних поселень за період.



## 4.7 Призначення аналітичних звітів у системі

Заключним етапом розробки інформаційно-аналітичної системи деканату є створення аналітичних звітів, які забезпечують зручний доступ до узагальнених та детальних даних для прийняття управлінських рішень. На цьому етапі результати, отримані під час побудови сховища даних та OLAP-куба, використовуються для візуалізації показників навчального процесу.

Для реалізації аналітичної звітності було використано SQL Server Reporting Services (SSRS) — інструмент платформи Microsoft SQL Server, призначений для створення інтерактивних звітів з підтримкою параметрів, ієархій, drill-down та різноманітних типів візуалізації.

Звіти, розроблені у межах даної курсової роботи, дозволяють аналізувати:

- склад студентів за факультетами та групами;
- рівень успішності студентів;
- рейтинги студентів;
- заборгованість по бібліотечних книгах;
- ефективність викладання дисциплін різними викладачами.

## 4.8 Створення проекту SQL Server Reporting Services

### 4.8.1 Створення проекту звітності

Для розробки аналітичних звітів у середовищі SQL Server Data Tools (SSDT) було створено окремий проект типу Report Server Project. Такий підхід дозволяє централізовано керувати звітами, джерелами даних та параметрами, а також спрощує процес подальшого розгортання на сервері звітів.

Проект звітності було створено з назвою Deanery\_SSRS, що відповідає предметній області курсової роботи.

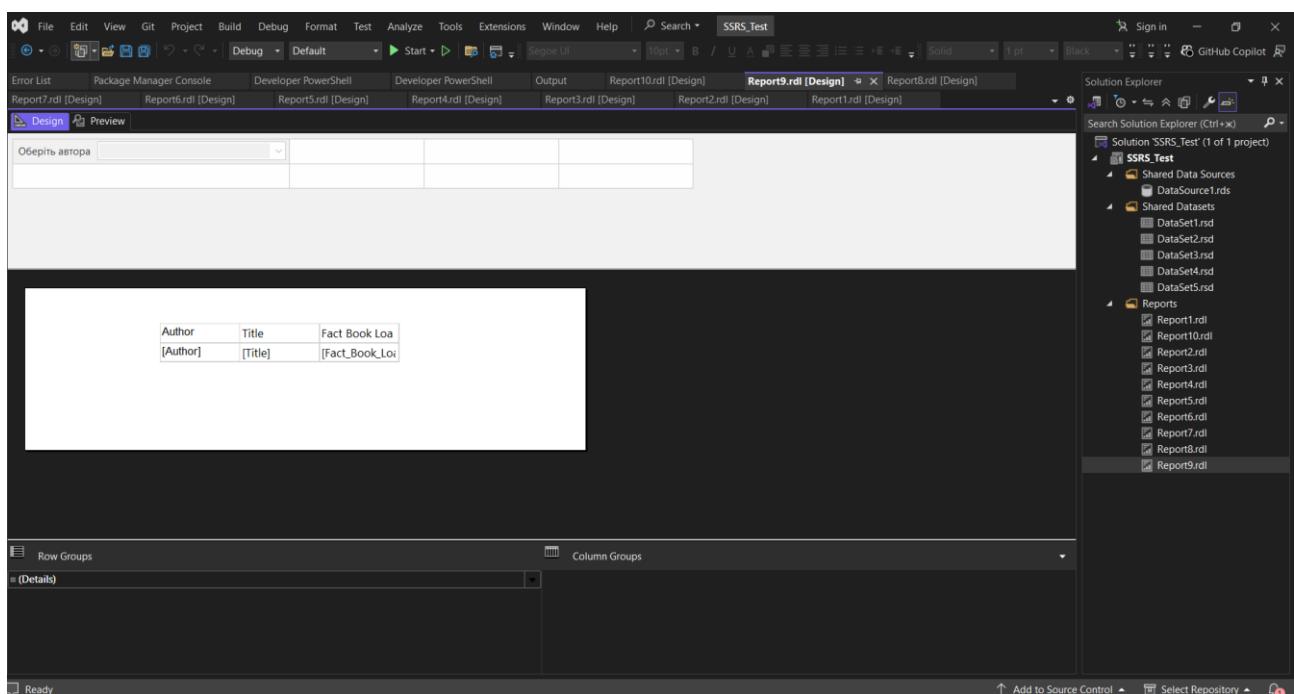
#### 4.8.2 Налаштування підключення до OLAP-куба

Оскільки всі аналітичні обчислення виконуються на рівні OLAP-куба, джерелом даних для звітів виступає SQL Server Analysis Services Multidimensional Cube.

Для забезпечення повторного використання підключення було створено спільне джерело даних (Shared Data Source), яке підключається безпосередньо до OLAP-куба *Deanery DW*, розгорнутого в базі даних SSAS\_Test.

Використання OLAP-куба як джерела даних дозволяє:

- уникнути дублювання бізнес-логіки у звітах;
- використовувати ієрархії та виміри куба;
- забезпечити високу продуктивність запитів.



#### 4.9 Загальна характеристика розроблених аналітичних звітів

У межах курсової роботи було реалізовано комплекс аналітичних звітів різних типів із використанням SQL Server Reporting Services. Під час розробки звітності було зроблено акцент не лише на табличному відображені даних, а й на використанні матричних структур, графічних візуалізацій та

інтерактивних дашбордів, що дозволяє забезпечити зручний та наочний аналіз даних для працівників деканату.

Розроблені звіти базуються на даних OLAP-куба та використовують попередньо розраховані міри, що забезпечує високу продуктивність і коректність аналітичних обчислень. У звітах активно застосовуються параметри, фільтрація, групування, агрегування показників, а також умовне форматування, що підвищує інформативність та зручність використання звітів.

Загалом у межах роботи було створено такі типи звітів:

- табличні звіти з багаторівневим групуванням;
- матричні (перехресні) звіти для аналізу показників у кількох вимірах;
- звіти з діаграмами різних типів;
- інтегрований дашборд, який поєднує таблиці та графіки в одному інтерфейсі.

#### **4.10 Табличний звіт «Список студентів по групах»**

Даний звіт призначений для відображення повного переліку студентів з ієрархічним групуванням за факультетами та академічними групами. Він дозволяє отримати структуровану інформацію про контингент студентів та використовується для контролю чисельності груп і аналізу розподілу студентів між факультетами.

Звіт реалізовано у вигляді класичного табличного звіту з багаторівневим групуванням:

- перший рівень — факультет;
- другий рівень — академічна група;
- третій рівень — окремі студенти.

Для підвищення наочності було застосовано сортування студентів за середнім балом, а також умовне форматування значень успішності.

Faculty Name	Rating	First Name	Last Name	Avr Garde
Medicine	1	Phillip	Werner	83,25
	2	Dante	Gallagher	78,625
	3	Daphne	Marquez	78,25
	4	Heidi	Beltran	77,75
	5	Lawrence	Tanner	77,75
	6	Taryn	Anthony	77,625
	7	Sylvia	Stafford	77,5
	8	Desiree	Mccall	77,375
	9	Nick	Knox	77,25
	10	Jessica	Parks	77,1538461538462

#### 4.11 Матричний звіт (Matrix Report)

Одним із ключових елементів аналітичної звітності є матричний звіт, який реалізує перехресне представлення даних (cross-tab). Даний тип звіту дозволяє одночасно аналізувати показники у двох незалежних вимірах.

У межах курсової роботи матричний звіт використовується для аналізу успішності студентів за навчальними предметами та семестрами. У рядках матриці відображаються предмети, у стовпцях — семестри або навчальні роки, а в комірках — агреговані значення середнього бала.

Такий підхід дозволяє:

- порівнювати результати навчання між різними предметами;
- відстежувати динаміку змін успішності у часі;
- швидко виявляти проблемні дисципліни.

## Матричний звіт активно використовує агрегування даних та обчислювані поля OLAP-куба.

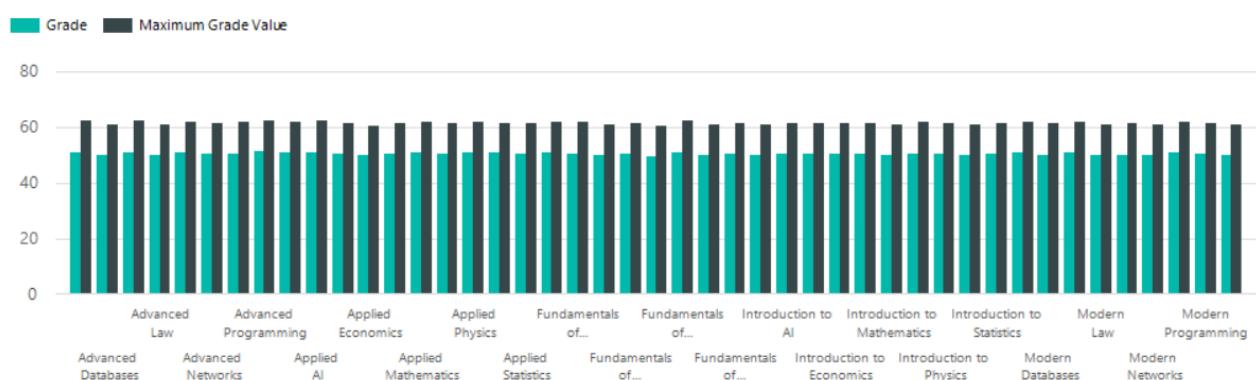
Full Name	Subject Name								
Abbott Alma	Applied Networks						74,5		
	Fundamentals of Law						63,0		
	Introduction to Statistics							27,5	
	Modern Economics								11,0
	Modern Statistics						72,0		
Abbott Charles	Applied Economics						88,0		
	Fundamentals of Networks	70,0							
	Modern Law							47,5	
	Modern Mathematics						48,0		
	Modern Physics						51,5		
Abbott Jeanine	Advanced Networks						87,0		
	Fundamentals of Programming		56,5						
	Introduction to AI							47,0	
	Modern Networks								72,0
	Advanced Networks						68,0		
Abbott Jeannie	Fundamentals of Programming		51,0						
	Introduction to							32,0	

### 4.12 Звіти з діаграмами (Chart Reports)

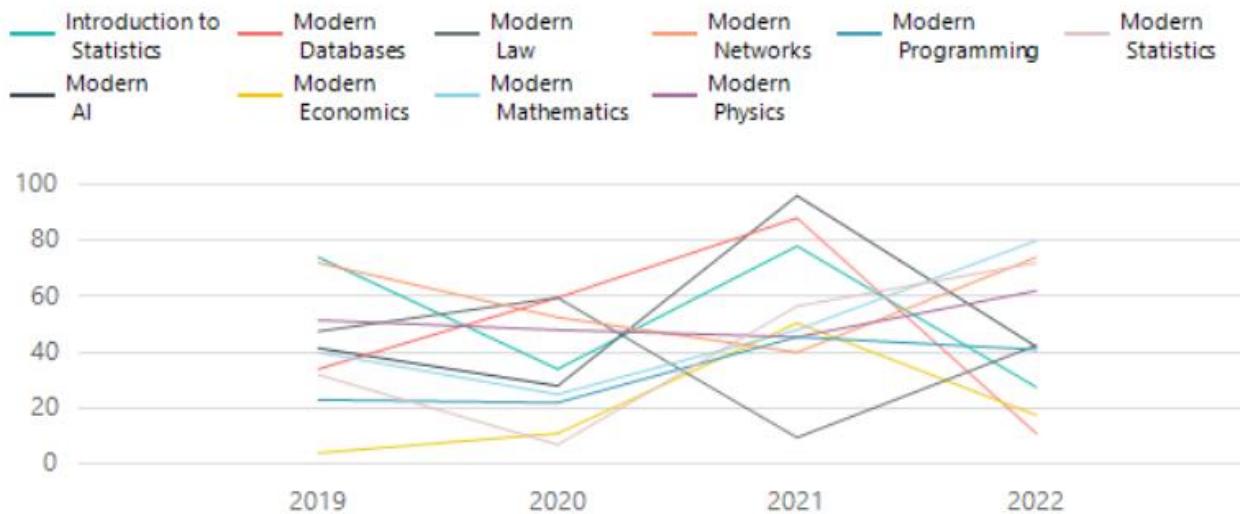
Для підвищення наочності аналітичної інформації у курсовій роботі було реалізовано кілька звітів з використанням різних типів діаграм. Графічне представлення даних дозволяє значно спростити сприйняття інформації та швидко виявляти тенденції.

У роботі використано такі типи діаграм:

- Column Chart (стовпчикова діаграма) — для порівняння середніх оцінок за предметами;



- Line Chart (лінійна діаграма) — для відображення динаміки успішності студентів у часі;



- Pie Chart (кругова діаграма) — для аналізу часток, наприклад розподілу предметів з найвищими середніми балами;

Розподіл студентів за факультетами



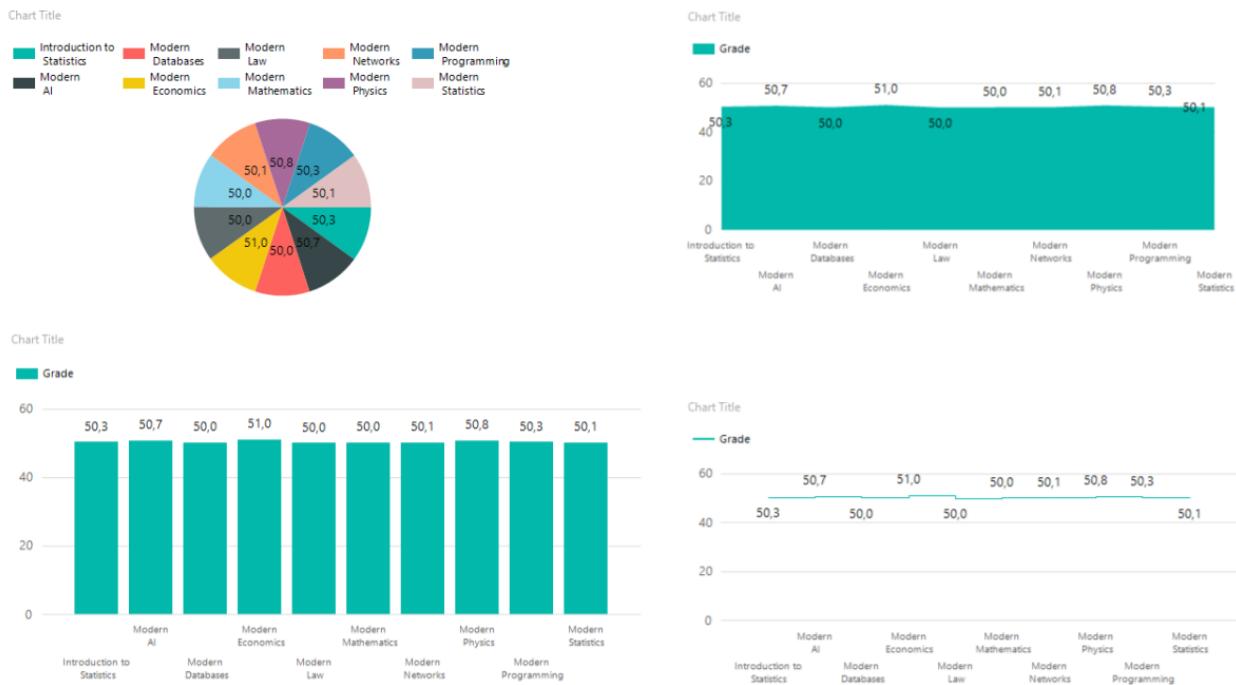
Для уникнення перевантаження графіків було реалізовано фільтрацію даних (наприклад, відображення лише топ-10 значень), що значно покращує читабельність діаграм.

#### 4.13 Інтерактивний дашборд

Окрему увагу в курсовій роботі приділено створенню інтегрованого дашборду, який поєднує кілька візуальних компонентів на одному екрані. Дашборд включає:

- табличне представлення агрегованих даних;
- одну або кілька діаграм, побудованих на основі того ж набору даних;
- параметри для інтерактивної фільтрації інформації.

Такий підхід дозволяє користувачу одночасно бачити числові значення та їх графічне представлення, що значно підвищує ефективність аналізу. Дашборд може використовуватися для швидкого огляду ключових показників навчального процесу.



## 4.14 Параметризовані звіти

У межах звітності активно використовуються параметри, які дозволяють обмежувати або змінювати вибірку даних без зміни структури звіту. Реалізовано параметри вибору факультету, студента, автора книги або іншого аналітичного об'єкта.

Використання параметрів забезпечує:

- зменшення обсягу відображуваних даних;
- підвищення продуктивності звітів;
- персоналізацію аналітичної інформації.

First Name	Last Name
Ruben	Strong

## **ВИСНОВКИ**

У межах курсової роботи було розглянуто процес проєктування, реалізації та аналізу надвеликої бази даних для інформаційної системи деканату закладу вищої освіти. Предметна область «Студент (Деканат)» характеризується значними обсягами різномірних даних, складними взаємозв'язками між сутностями та необхідністю зберігання історичної інформації за тривалий період часу.

У ході виконання роботи було проведено детальний аналіз предметної області, визначено основні бізнес-процеси, сутності та інформаційні потоки. На основі отриманих результатів побудовано концептуальну ER-модель, логічну реляційну модель та фізичну структуру бази даних у середовищі Microsoft SQL Server.

Для забезпечення аналітичної обробки великих обсягів даних було спроектовано сховище даних за зіркоподібною схемою та реалізовано ETL-процеси з використанням SQL Server Integration Services. Побудоване сховище даних забезпечує зберігання історичних даних та є основою для багатовимірного аналізу.

З використанням SQL Server Analysis Services було створено OLAP-куб, який дозволяє виконувати багатовимірний аналіз даних за факультетами, групами, студентами, предметами та часовими періодами. Для візуалізації результатів аналізу розроблено аналітичні звіти та дашборди у SQL Server Reporting Services.

Результати курсової роботи підтверджують ефективність використання технологій Microsoft SQL Server для побудови масштабованих інформаційно-аналітичних систем та аналізу надвеликих обсягів даних. Запропоновані рішення можуть бути використані як основа для впровадження подібних систем у реальних закладах вищої освіти.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Microsoft. SQL Server Documentation. <https://learn.microsoft.com/sql>
2. Microsoft. SQL Server Integration Services (SSIS) Documentation. <https://learn.microsoft.com/sql/integration-services>
3. Microsoft. SQL Server Analysis Services (SSAS) Documentation. <https://learn.microsoft.com/analysis-services>
4. Microsoft. SQL Server Reporting Services (SSRS) Documentation. <https://learn.microsoft.com/sql/reporting-services>
5. Kimball R., Ross M. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling.* – 3rd Edition. – Wiley, 2013.
6. Inmon W. H. *Building the Data Warehouse.* – 4th Edition. – John Wiley & Sons, 2005.
7. Date C. J. *An Introduction to Database Systems.* – 8th Edition. – Pearson Education, 2003.
8. Elmasri R., Navathe S. *Fundamentals of Database Systems.* – 7th Edition. – Pearson, 2016.
9. Redgate Software. SQL Data Generator Documentation. – <https://www.red-gate.com>
10. Connolly T., Begg C. *Database Systems: A Practical Approach to Design, Implementation, and Management.* – Pearson, 2015.
11. Golfarelli M., Rizzi S. *Data Warehouse Design: Modern Principles and Methodologies.* – McGraw-Hill, 2009.
12. Larson B. *Delivering Business Intelligence with Microsoft SQL Server.* – Microsoft Press, 2013.
13. Russo M., Ferrari A. *Introducing Microsoft SQL Server Analysis Services.* – Microsoft Press, 2012.
14. Kimball Group. Dimensional Modeling Techniques. – <https://www.kimballgroup.com>
15. Chaudhuri S., Dayal U. An Overview of Data Warehousing and OLAP Technology. – ACM SIGMOD Record, 1997.

- 16.Han J., Kamber M., Pei J. *Data Mining: Concepts and Techniques*. – Morgan Kaufmann, 2011.
- 17.Oracle Corporation. Data Warehousing Concepts. – <https://docs.oracle.com>
- 18.ISO/IEC 9075. Information technology — Database languages — SQL.
- 19.Visual Studio Documentation. SQL Server Data Tools (SSDT).  
– <https://learn.microsoft.com/visualstudio>