

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 «Кабельне телебачення»
Варіант 10

Виконала:
студентка групи ПМ-41
Анна-Софія ПАУК
Перевірив:
доцент кафедри ПМ
Богдан ЛЮБІНСЬКИЙ

Львів — 2025 р.

ЗМІСТ

ВСТУП	3-4
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	5-11
1.1 Детальний опис обраної предметної області	
1.2 Опис основних бізнес-процесів	
1.3 Функціональні вимоги до системи	
1.4 Бізнес-правила та обмеження	
1.5 Матеріально-технічне забезпечення	
РОЗДІЛ 2. КОНЦЕПТУАЛЬНЕ ПРОЄКТУВАННЯ БАЗИ ДАНИХ	12-20
2.1 Опис сутностей та їх атрибутив	
2.2 Логічне проєктування бази даних	
2.3 Фізичне проєктування бази даних	
2.4 Генерація та наповнення даних	
2.5 Перевірка коректності даних	
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ETL-ПРОЦЕСІВ	21-26
3.1 Проєктування та структура сховища даних (Data Warehouse)	
РОЗДІЛ 4. ПОБУДОВА OLAP-КУБА ТА АНАЛІТИЧНИХ ЗВІТІВ	27-37
4.1 Концепція та обґрунтування побудови OLAP-куба	
4.2 Створення вимірів (Dimensions)	
4.3 Створення мір (Measures) та груп мір (Measure Groups)	
4.4 Розгортання та процесинг OLAP-куба	
4.5 Аналітичні звіти та візуалізація даних	
ВИСНОВКИ	38 -30

ДОДАТКИ

Вступ

Сучасний етап розвитку телекомунікаційної галузі характеризується активним переходом від класичного лінійного мовлення до інтерактивних мультимедійних платформ. Кабельне телебачення перестає бути лише засобом трансляції телевізійних каналів і перетворюється на комплексний цифровий сервіс, який поєднує телебачення, відео за запитом, персоналізований контент та інтерактивні послуги. Така трансформація супроводжується стрімким зростанням обсягів даних, що накопичуються в інформаційних системах операторів.

Провайдери кабельного телебачення працюють з великими масивами інформації, які включають дані про абонентів, підключені пакети каналів, історію переглядів, замовлення фільмів, фінансові операції та взаємодію клієнтів із сервісами компанії. Особливо значні обсяги даних формуються в межах сервісів «відео за запитом» (VOD), де кожна дія користувача фіксується у вигляді транзакцій. З часом такі дані накопичуються за кілька років і досягають масштабів, при яких традиційні підходи до зберігання та обробки інформації стають малоефективними.

Операційні бази даних (OLTP-системи) добре підходять для забезпечення щоденної діяльності провайдера, зокрема реєстрації нових абонентів, обробки замовлень та проведення розрахунків. Водночас їх основне призначення полягає в підтримці швидких транзакцій, а не в аналітичній обробці великих обсягів історичних даних. При спробі виконання складних аналітичних запитів такі системи зазнають значного навантаження, що негативно впливає на продуктивність та стабільність роботи.

У зв'язку з цим виникає потреба у використанні технологій надвеликих баз даних (Very Large Data Bases, VLDB), які орієнтовані на зберігання та аналіз інформації в масштабах сотень мільйонів і мільярдів записів. Одним із ключових підходів у цій сфері є побудова сховищ даних (Data Warehouse), які дозволяють відокремити аналітичні процеси від операційних систем. Сховище даних забезпечує зберігання

узгодженої, очищеної та структурованої інформації, придатної для подальшого аналізу.

Невід'ємною складовою побудови сховищ даних є реалізація ETL-процесів, що включають витягування даних з різних джерел, їх очищення, трансформацію та завантаження у аналітичну базу. Для систем кабельного телебачення це особливо важливо, оскільки дані можуть надходити з різних підсистем, мати неоднорідну структуру та містити помилки або пропуски. Коректно побудовані ETL-процеси забезпечують цілісність даних і підвищують якість подальшого аналізу.

Подальший аналіз даних у сховищі здійснюється з використанням багатовимірних моделей та OLAP-технологій. OLAP-куби дозволяють аналізувати інформацію в різних розрізах, зокрема за часовими періодами, типами контенту, групами абонентів або фінансовими показниками. Для провайдера кабельного телебачення це створює можливості для оцінки популярності фільмів і каналів, аналізу динаміки абонентської бази, виявлення сезонних тенденцій та підтримки управлінських рішень.

Метою курсової роботи є розробка та аналіз архітектури надвеликої бази даних для системи управління кабельним телебаченням. Робота спрямована на створення повного циклу обробки даних — від проєктування реляційної структури та наповнення її великими обсягами реалістичних тестових даних до побудови сховища даних, реалізації ETL-процесів, багатовимірного аналізу та формування аналітичної звітності.

Об'єктом дослідження є процеси функціонування оператора кабельного телебачення, пов'язані з обліком абонентів, наданням цифрових послуг, обробкою замовлень фільмів та управлінням фінансовими розрахунками. Предметом дослідження є методи та технології проєктування, інтеграції та аналітичної обробки великих обсягів даних із використанням сервісів SSIS, SSAS та SSRS у середовищі Visual Studio.

Розділ 1. Аналіз предметної області

1.1 Детальний опис обраної предметної області

Предметною областю даної курсової роботи є діяльність оператора кабельного телебачення, який надає послуги з трансляції телевізійних каналів та доступу до цифрового медіаконтенту, зокрема сервісів «відео за запитом» (VOD). Сучасні оператори кабельного телебачення функціонують як складні інформаційні системи, що поєднують технічну інфраструктуру, програмні рішення та бізнес-процеси, орієнтовані на обслуговування великої кількості абонентів.

Основою діяльності оператора є абонентська база, яка включає інформацію про користувачів послуг, їхні підключення, тарифні плани та пакети каналів. Кожен абонент може бути підключений до одного або кількох пакетів, що відрізняються набором телеканалів, вартістю та умовами надання послуг. Крім того, абоненти мають можливість користуватися додатковими сервісами, зокрема замовляти окремі фільми або серіали у межах платформи VOD.

Важливим елементом предметної області є медіаконтент, який включає телевізійні канали та кінофільми. Контент може групуватися за жанрами, тематикою або іншими ознаками, що дозволяє формувати різні пакети послуг та аналізувати споживчі вподобання користувачів. Інформація про перегляди, замовлення та популярність контенту є цінним джерелом даних для подальшого аналітичного опрацювання.

Фінансова складова діяльності оператора кабельного телебачення охоплює процеси формування рахунків, обліку платежів та контролю за розрахунками з абонентами. Кожне замовлення або підключення послуги супроводжується відповідною фінансовою операцією, що фіксується в інформаційній системі. З часом такі дані накопичуються у великих обсягах та формують історію фінансової діяльності компанії.

У межах операційної діяльності всі зазначені процеси підтримуються реляційною базою даних, орієнтованою на швидке виконання транзакцій. Проте для аналізу діяльності оператора, оцінки ефективності надання послуг, виявлення тенденцій

у зміні абонентської бази та популярності контенту необхідне використання аналітичних підходів. Саме тому виникає потреба у побудові сховища даних, яке дозволяє інтегрувати інформацію з різних підсистем та зберігати її у зручному для аналізу вигляді.

1.2 Опис основних бізнес-процесів

Основні бізнес-процеси предметної області кабельного телебачення охоплюють повний цикл взаємодії абонента з інформаційною системою — від моменту реєстрації до здійснення фінансових операцій та подальшого аналітичного аналізу отриманих даних. Центральним бізнес-процесом у межах даної предметної області є процес підписки абонента на пакети каналів, оскільки саме він формує основу для надання послуг, нарахування абонентської плати та побудови фінансової аналітики.

Початковим етапом функціонування системи є реєстрація абонента. У процесі реєстрації в системі створюється запис абонента із збереженням персональних та контактних даних, а також визначенням його поточного статусу. Дані про абонентів використовуються для подальшого обліку підписок, замовлень і платежів, а також для аналізу динаміки абонентської бази та кількості активних користувачів.

Наступним важливим бізнес-процесом є управління пакетами каналів та окремими телеканалами. У межах цього процесу адміністратор системи формує перелік доступних пакетів каналів, визначає їх вартість та наповнення. Актуальна структура пакетів є основою для формування пропозиції абонентам і дозволяє виконувати аналітичні дослідження популярності каналів та ефективності пакетних пропозицій.

Ключовим бізнес-процесом предметної області є підключення абонента до пакету каналів, що реалізується у вигляді підписки. Підписка фіксує факт користування абонентом конкретним пакетом каналів у визначений період часу та містить інформацію про дату початку, можливу дату завершення та поточний статус активності. Наявність історії підписок дозволяє аналізувати тривалість

користування пакетами, кількість активних підписок у різні періоди та загальні тенденції споживання послуг.

Окремим бізнес-процесом є замовлення фільмів. Абонент має можливість замовляти фільми незалежно від наявності активної підписки на пакет каналів. У результаті виконання цього процесу в системі створюється запис замовлення із зазначенням дати та вартості. Дані про замовлення використовуються для формування рейтингу популярності фільмів і аналізу споживчих уподобань абонентів.

Фінансовий блок бізнес-процесів включає формування рахунків та облік платежів. Рахунок формується на основі активних підписок абонента та замовлених фільмів за певний розрахунковий період і містить загальну суму до оплати. У подальшому абонент здійснює оплату рахунку повністю або частково, що фіксується у вигляді платежу із зазначенням суми, дати та способу оплати. Дані про рахунки та платежі є основою для фінансової аналітики, аналізу заборгованостей, платіжної дисципліни та структури доходів.

Таким чином, бізнес-процеси предметної області кабельного телебачення утворюють цілісну систему, у якій підписка виступає центральним елементом, що поєднує абонентів, пакети каналів, фінансові операції та аналітичні показники.

1.3 Функціональні вимоги до системи

Інформаційна система кабельного телебачення повинна забезпечувати ведення обліку абонентів із можливістю створення та редагування їхніх даних, а також збереження повної історії підписок кожного абонента. Система має підтримувати оформлення, зміну та завершення підписок на пакети каналів із фіксацією періодів їх дії та статусів активності.

Функціональні можливості системи повинні включати керування пакетами каналів і переліком каналів, що входять до кожного пакету. Також система повинна забезпечувати облік замовлень фільмів, формування рахунків за розрахункові періоди та реєстрацію платежів абонентів.

Окрему роль відіграють аналітичні функції системи, які повинні забезпечувати можливість аналізу популярності пакетів і каналів, формування фінансових звітів за обрані періоди, а також дослідження динаміки абонентської бази.

1.4 Бізнес-правила та обмеження

У межах предметної області кабельного телебачення визначено низку бізнес-правил та обмежень, які регламентують порядок надання послуг абонентам і забезпечують коректність обліку даних у системі. Дотримання цих правил є необхідним для забезпечення цілісності інформації, коректного формування фінансових нарахувань та подальшого аналітичного аналізу.

Одним з основних бізнес-правил є те, що абонент може мати декілька підписок у різні періоди часу, проте одночасно допускається наявність не більше однієї активної підписки. Це обмеження дозволяє уникнути дублювання абонентської плати та забезпечує однозначність визначення набору послуг, які надаються абоненту в конкретний момент часу. Наявність історії підписок дає можливість аналізувати зміну вподобань абонентів, частоту переходів між пакетами каналів та тривалість користування окремими тарифними пропозиціями.

Кожна підписка обов'язково пов'язується з конкретним абонентом та одним пакетом каналів. Підписка має фіксовану дату початку, яка визначає момент підключення послуг, тоді як дата завершення може бути відсутньою для активної підписки. Такий підхід дозволяє коректно вести облік активних підписок у будь-який момент часу та забезпечує можливість формування часової аналітики, зокрема підрахунку кількості активних підписок за місяцями або роками.

Абонентська плата нараховується виключно за активні підписки. Це правило забезпечує коректність фінансових розрахунків і виключає нарахування платежів за неактивні або завершені підписки. Завдяки цьому система дозволяє точно визначати доходи від абонентської плати та виконувати аналіз фінансових показників у розрізі пакетів каналів і періодів часу.

Замовлення фільмів у межах сервісу відео на вимогу можливе лише для зареєстрованих абонентів системи. Це обмеження забезпечує узгодженість даних та дозволяє коректно пов'язувати замовлення з конкретними абонентами для подальшого аналізу їхніх споживчих уподобань. Дані про замовлення використовуються для формування рейтингів популярності фільмів і оцінки структури доходів від додаткових послуг.

Формування рахунків у системі здійснюється для кожного абонента за визначений розрахунковий період. Рахунок може включати нарахування за активну підписку, а також вартість замовлених фільмів. Такий підхід дозволяє агрегувати фінансові дані та забезпечує прозорість розрахунків між абонентом і постачальником послуг.

Кожен платіж у системі обов'язково прив'язується до конкретного рахунку, що забезпечує можливість обліку повних і часткових оплат. Це дозволяє аналізувати платіжну дисципліну абонентів, рівень заборгованості та ефективність процесу стягнення платежів. Сукупність зазначених бізнес-правил і обмежень формує логічно цілісну модель предметної області, яка забезпечує коректну роботу системи та створює надійну основу для подальшого проєктування бази даних і аналітичної обробки інформації.

1.5 Матеріально-технічне забезпечення

Для реалізації та дослідження предметної області кабельного телебачення у межах курсової роботи було використано комплекс програмних і технічних засобів, які забезпечують повний цикл роботи з надвеликими обсягами даних. Застосоване матеріально-технічне забезпечення дозволяє виконувати зберігання, обробку, інтеграцію та аналітичний аналіз даних відповідно до вимог технічного завдання та специфіки предметної області.

Основною платформою для зберігання та обробки даних є система управління базами даних Microsoft SQL Server. Вибір даної СУБД обумовлений її широким

використанням у корпоративних інформаційних системах, підтримкою великих обсягів даних, наявністю розвинених механізмів забезпечення цілісності, безпеки та продуктивності, а також можливістю інтеграції з аналітичними інструментами. Microsoft SQL Server використовується для зберігання операційних даних, що описують абонентів, підписки, пакети каналів, замовлення, рахунки та платежі.

Для адміністрування бази даних та виконання SQL-запитів використовувалося середовище SQL Server Management Studio (SSMS). Даний інструмент забезпечує зручний доступ до структури бази даних, дозволяє створювати та модифікувати таблиці, переглядати та перевіряти дані, виконувати аналітичні та агрегатні запити, а також здійснювати контроль коректності наповнення бази даних. Використання SSMS є доцільним у контексті курсової роботи, оскільки воно дозволяє наочно перевіряти результати реалізації бізнес-процесів і відповідність структури бази даних вимогам предметної області.

Для реалізації процесів інтеграції та обробки великих обсягів даних застосовувалися засоби SQL Server Integration Services (SSIS). Даний інструмент використовується для побудови ETL-процесів, які забезпечують витягування даних, їх трансформацію та завантаження у цільову структуру сховища даних. У межах предметної області кабельного телебачення SSIS дозволяє обробляти значні масиви інформації про підписки, фінансові операції та історичні дані за тривалий період часу, що є необхідним для подальшого аналітичного використання.

Розробка та налаштування інтеграційних процесів виконувалися у середовищі Microsoft Visual Studio з використанням SQL Server Data Tools. Це середовище дозволяє створювати проекти інтеграційних сервісів, керувати з'єднаннями з джерелами даних та реалізовувати логіку обробки інформації у межах єдиного рішення. Використання Visual Studio у поєднанні з SSIS забезпечує гнучкість та масштабованість інтеграційних процесів.

Для аналітичної обробки даних та побудови звітів використовуються інструменти платформи Microsoft SQL Server, зокрема SQL Server Analysis Services (SSAS) та SQL Server Reporting Services (SSRS). Вони дозволяють реалізувати багатовимірний аналіз даних, створювати OLAP-куб і формувати аналітичні звіти щодо динаміки абонентської бази, структури підписок, фінансових показників та популярності контенту.

Матеріально-технічне забезпечення також включає апаратні ресурси персонального комп'ютера, на якому розгорнуто середовище розробки, сервер бази даних та аналітичні інструменти. Достатній обсяг оперативної пам'яті та продуктивність процесора є важливими факторами для коректної роботи з великими наборами даних, виконання складних запитів і обробки інтеграційних процесів. Хоча курсова робота виконується у навчальному середовищі, використані програмні засоби дозволяють моделювати роботу реальної інформаційної системи кабельного телебачення.

Таким чином, використане матеріально-технічне забезпечення утворює цілісне середовище для реалізації предметної області курсової роботи. Поєднання Microsoft SQL Server, SQL Server Management Studio, Visual Studio, SQL Server Integration Services та аналітичних сервісів забезпечує можливість роботи з надвеликими обсягами даних, реалізації бізнес-процесів та виконання аналітичних задач відповідно до вимог дисципліни.

Розділ 2 Концептуальне проєктування бази даних

2.1 Опис сущностей та їх атрибутив

Сутність Subscriber (Абонент)

Сутність **Subscriber** призначена для зберігання інформації про фізичних осіб, які користуються послугами кабельного телебачення. Абонент є базовим об'єктом системи, оскільки всі фінансові та сервісні операції виконуються від його імені.

Сутність описується такими характеристиками:

- SubscriberID
- FullName
- Email
- Address
- RegistrationDate
- Status

Ідентифікатор абонента використовується для його однозначної ідентифікації в системі та застосовується у зв'язках з іншими сущностями. Персональні та контактні дані дозволяють ідентифікувати користувача та забезпечують взаємодію з ним. Дата реєстрації та статус абонента використовуються для аналізу динаміки абонентської бази та визначення активних користувачів.

Сутність ChannelGroup (Пакет каналів)

Сутність **ChannelGroup** використовується для опису пакетів телеканалів, які пропонуються абонентам за фіксовану щомісячну плату. Вона дозволяє структурувати тарифні пропозиції та спрощує управління асортиментом послуг.

Сутність містить такі характеристики:

- ChannelGroupID
- GroupName

- MonthlyFee

Ідентифікатор пакета забезпечує його унікальність у системі. Назва пакета використовується для відображення інформації у звітах і під час оформлення підписок. Щомісячна вартість застосовується для формування рахунків і аналізу доходів.

Сутність Channel (Канал)

Сутність **Channel** представляє окремий телевізійний канал, що входить до складу певного пакета каналів. Вона дозволяє деталізувати структуру пакетів і аналізувати контентну пропозицію.

Сутність описується такими характеристиками:

- ChannelID
- ChannelName

Ідентифікатор каналу забезпечує його унікальність, а назва використовується для формування переліків каналів і аналітичних звітів.

Сутність Subscription (Підписка)

Сутність **Subscription** є ключовою в предметній області, оскільки фіксує факт підключення абонента до конкретного пакета каналів у певний період часу.

Сутність містить такі характеристики:

- SubscriptionID
- StartDate
- EndDate
- Status

Інформація про період дії підписки дозволяє коректно нараховувати абонентську плату та зберігати історію змін тарифних планів. Статус підписки

використовується для визначення активних послуг у системі та побудови часової аналітики.

Сутність Movie (Фільм)

Сутність **Movie** призначена для зберігання інформації про фільми, доступні для замовлення в сервісі відео на вимогу.

Сутність описується такими характеристиками:

- MovieID
- Title
- Genre
- ReleaseYear
- Price

Ці дані використовуються для формування каталогу фільмів, аналізу споживчих уподобань абонентів і розрахунку вартості замовлень.

Сутність Order (Замовлення)

Сутність **Order** фіксує факт замовлення абонентом конкретного фільму.

Сутність містить такі характеристики:

- OrderID
- OrderDate
- OrderAmount

Інформація про замовлення використовується для обліку операцій сервісу VOD та формування фінансової аналітики.

Сутність Invoice (Рахунок)

Сутність **Invoice** використовується для зберігання інформації про фінансові нарахування абоненту за визначений розрахунковий період.

Сутність описується такими характеристиками:

- InvoiceID
- InvoiceDate
- TotalAmount
- DueDate
- PaymentStatus

Рахунок може включати нарахування за активну підписку та замовлені фільми, що забезпечує прозорість фінансових розрахунків.

Сутність Payment (Платіж)

Сутність **Payment** призначена для обліку оплат рахунків абонентами.

Сутність містить такі характеристики:

- PaymentID
- PaymentDate
- PaymentAmount
- PaymentMethod

Ці дані дозволяють враховувати повні та часткові оплати, а також аналізувати платіжну дисципліну абонентів.

2.2 Логічне проєктування бази даних

На етапі логічного проєктування здійснюється перехід від концептуальної моделі предметної області до логічної структури бази даних. Основною метою цього етапу є формування таблиць, визначення первинних і зовнішніх ключів, а також забезпечення цілісності та узгодженості даних. Логічне проєктування виконується

з урахуванням правил нормалізації, що дозволяє зменшити надлишковість даних і запобігти аномаліям вставки, оновлення та видалення.

Побудована логічна модель бази даних відповідає вимогам перших трьох нормальних форм.

Перша нормальна форма (1НФ)

Усі таблиці бази даних приведені до першої нормальної форми, оскільки кожне поле містить атомарні, неподільні значення. У таблицях відсутні повторювані групи полів і багатозначні атрибути. Наприклад, інформація про пакети каналів, підписки, замовлення та платежі зберігається у вигляді окремих записів, що забезпечує коректну обробку даних.

Друга нормальна форма (2НФ)

Друга нормальна форма забезпечується тим, що всі неключові поляожної таблиці повністю функціонально залежать від первинного ключа. У моделі відсутні часткові залежності, оскільки всі таблиці мають прості первинні ключі. Це дозволяє уникнути дублювання інформації та спрощує підтримку цілісності даних.

Третя нормальна форма (3НФ)

Третя нормальна форма досягається завдяки відсутності транзитивних залежностей між неключовими полями. Дані, що логічно належать до різних об'єктів предметної області, винесені в окремі таблиці. Наприклад, інформація про абонентів, пакети каналів, підписки та фінансові операції зберігається в окремих сутностях, що забезпечує гнучкість структури та спрощує подальший аналітичний аналіз.

Первинні та зовнішні ключі

Для забезпечення унікальної ідентифікації записів у кожній таблиці визначено первинні ключі. У таблиці **Subscriber** первинним ключем є поле *SubscriberID*. Аналогічно, первинними ключами відповідних таблиць є *ChannelGroupID*, *ChannelID*, *SubscriptionID*, *MovieID*, *OrderID*, *InvoiceID* та *PaymentID*.

Зовнішні ключі використовуються для реалізації зв'язків між таблицями та забезпечення посилальної цілісності. Зокрема, таблиця **Subscription** містить зовнішній ключ, що посилається на таблиці **Subscriber** та **ChannelGroup**, що дозволяє однозначно визначити абонента та пакет каналів для кожної підписки. Таблиця **Order** пов'язується з таблицями **Subscriber** та **Movie**, що забезпечує коректний облік замовлень фільмів. Таблиця **Invoice** містить посилання на таблицю **Subscriber**, а таблиця **Payment** — на таблицю **Invoice**, що дозволяє реалізувати облік фінансових нарахувань і платежів.

Логічна модель бази даних реалізує зв'язки типу один-до-багатьох між основними сутностями предметної області. Один абонент може мати декілька підписок, замовлень і рахунків, проте кожен із цих об'єктів належить лише одному абоненту. Один пакет каналів може бути використаний у багатьох підписках, а кожна підписка пов'язується лише з одним пакетом. Один рахунок може бути оплачений кількома платежами, що дозволяє враховувати часткові оплати.

2.3 Фізичне проектування бази даних

На етапі фізичного проектування було виконано реалізацію логічної моделі бази даних у середовищі системи управління базами даних. Для кожної сутності предметної області створено відповідні таблиці з визначенням типів даних, первинних і зовнішніх ключів, а також обмежень цілісності. Фізична структура бази даних повністю відповідає логічній моделі, розробленій на попередньому етапі.

SQL-скрипти створення таблиць, ключів і зв'язків між ними наведено в додатках до курсової роботи. Використання скриптів дозволяє відтворити структуру бази даних та перевірити коректність реалізації моделі.

2.4 Генерація та наповнення даних

Для наповнення розробленої бази даних було використано програмний скрипт, реалізований мовою програмування Python із застосуванням бібліотек **Pandas** та **SQLAlchemy**. Такий підхід дозволив автоматизувати процес створення великих обсягів взаємопов'язаних даних та забезпечити їх коректне завантаження безпосередньо в реляційну базу даних за допомогою SQL-запитів.

Генерація даних виконувалася в середовищі операційної системи Linux, що накладало певні обмеження на використання оперативної пам'яті та обчислювальних ресурсів. У зв'язку з цим процес наповнення бази даних було оптимізовано та поділено на кілька послідовних етапів, що дозволило стабільно обробляти значні обсяги інформації без перевантаження системи.

На першому етапі формувалися довідниківі дані, які є основою для подальших бізнес-процесів. Було згенеровано інформацію про пакети каналів і телеканали, що входять до їх складу. Також на цьому етапі було створено каталог фільмів сервісу відео на вимогу, який містив десятки та сотні тисяч записів, що дозволило зmodелювати реалістичний обсяг контенту.

На другому етапі здійснювалася генерація даних про абонентів системи. Було створено інформацію про **щонайменше 100 000 абонентів**, для яких генерувалися персональні та контактні дані, адреса підключення, дата реєстрації та статус активності. Період генерації даних охоплював **п'ятирічний часовий інтервал**, що дозволило зmodелювати довготривалу історію функціонування системи та подальший аналіз динаміки абонентської бази.

Третій етап був присвячений формуванню підписок, які є ключовою сутністю предметної області. Для кожного абонента створювалися записи підписок на пакети каналів із датами початку та завершення в межах п'ятирічного періоду. Алгоритм генерації забезпечував дотримання бізнес-правила, згідно з яким у кожен момент часу абонент може мати не більше однієї активної підписки, що гарантувало коректність історії користування послугами.

На наступному етапі виконувалася генерація замовлень фільмів у сервісі відео за запитом (VOD). Загальна кількість згенерованих записів становила **приблизно один мільйон замовлень**, що дозволило змоделювати інтенсивне використання додаткових послуг і створити значне навантаження на базу даних. Для кожного замовлення визначався абонент, обраний фільм, дата замовлення та вартість перегляду.

Після цього формувалися рахунки для абонентів за розрахункові періоди, до яких включалися нарахування за активні підписки та замовлені фільми. На завершальному етапі генерувалися платежі, які могли покривати рахунки повністю або частково, що дозволило змоделювати реальні сценарії фінансової поведінки абонентів.

Застосування бібліотек Pandas і SQLAlchemy, поетапна генерація даних та використання великого обсягу історичної інформації забезпечили коректність зв'язків між таблицями, дотримання обмежень цілісності та можливість подальшої аналітичної обробки даних у межах курсової роботи.

2.5 Перевірка коректності даних

З метою перевірки коректності згенерованих та завантажених даних було виконано низку SQL-запитів для контролю обсягів і часових меж даних у ключових таблицях операційної бази даних. Перевірка кількісних показників підтвердила відповідність фактичних обсягів даних вимогам технічного завдання.

Зокрема, у таблиці **Subscribers** було згенеровано рівно **100 000 записів**, що відповідає заданій кількості абонентів. Аналогічно, у таблиці **Orders** зафіковано **1 000 000 записів**, що підтверджує коректну генерацію замовлень фільмів у необхідному обсязі.

Додатково було виконано перевірку часових меж даних у таблиці замовлень. Результати запиту показали, що мінімальна дата замовлення відповідає початку дослідженого періоду, а максимальна дата — його завершенню. Це підтверджує коректне формування п'ятирічної історії даних без пропусків та часових зсувів.

Отримані результати свідчать про те, що дані були згенеровані та завантажені коректно, відповідають заданим кількісним і часовим обмеженням та є придатними для подальшого використання у сховищі даних, OLAP-кубі та аналітичних звітах.

```
SELECT COUNT(*) AS TotalSubscribers  
FROM Subscribers;
```

Results	
TotalSubscribers	
1	100000
TotalOrders	
1	1000000

```
SELECT COUNT(*) AS TotalOrders  
FROM [Orders];
```

```
SELECT  
    MIN(OrderDate) AS FirstOrderDate,  
    MAX(OrderDate) AS LastOrderDate  
FROM [Orders];
```

Results		
	FirstOrderDate	LastOrderDate
1	2021-01-04 19:15:11.193	2026-01-03 19:13:30.977

Розділ 3. Реалізація ETL-процесів

ETL (Extract – Transform – Load) — це підхід до обробки даних, який передбачає поетапне вилучення даних із джерел, їх перетворення відповідно до бізнес-правил предметної області та подальше завантаження у структури, призначенні для аналітичної обробки. ETL-процеси є базовим елементом при побудові сховищ даних та багатовимірних аналітичних систем.

У межах даної курсової роботи ETL-процеси використовуються для підготовки даних з операційної бази даних кабельного телебачення до подальшого аналізу. Реалізація ETL виконувалася з орієнтацією на **схему зірки**, що є однією з найбільш поширених моделей організації даних у сховищах даних.

3.1 Проєктування та структура сховища даних (Data Warehouse)

У межах курсової роботи було спроектовано та реалізовано сховище даних (Data Warehouse), призначене для аналітичної обробки інформації про діяльність системи кабельного телебачення. Основною метою створення сховища даних є зберігання узгоджених, очищених та історичних даних, отриманих з операційної бази, а також забезпечення можливості виконання агрегованих аналітичних запитів і побудови звітів.

Сховище даних спроектоване відповідно до **схеми зірки**, яка передбачає наявність таблиць вимірів та таблиць фактів. Така структура дозволяє спростити логіку аналізу даних, підвищити продуктивність запитів і забезпечити наочне представлення предметної області.

Таблиця виміру абонентів (DimSubscriber)

Таблиця виміру абонентів використовується для зберігання довідникової інформації про абонентів системи кабельного телебачення. Вона містить дані, що описують абонента як користувача послуг, зокрема ідентифікаційні та контактні

відомості, а також статус активності. Даний вимір використовується для аналізу підписок, замовлень фільмів і фінансових операцій у розрізі абонентів.

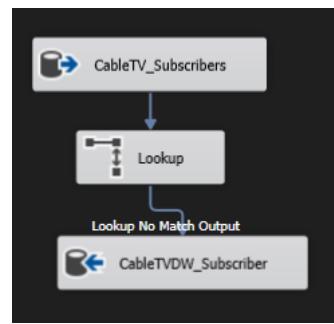


Рис. 1 Таблиця виміру абонентів

Таблиця виміру пакетів каналів (DimChannelGroup)

Таблиця виміру пакетів каналів призначена для зберігання інформації про доступні пакети телеканалів. Вона містить назви пакетів та їхні основні характеристики, зокрема вартість підписки. Даний вимір використовується при аналізі популярності пакетів каналів, структури підписок та доходів від абонентської плати.

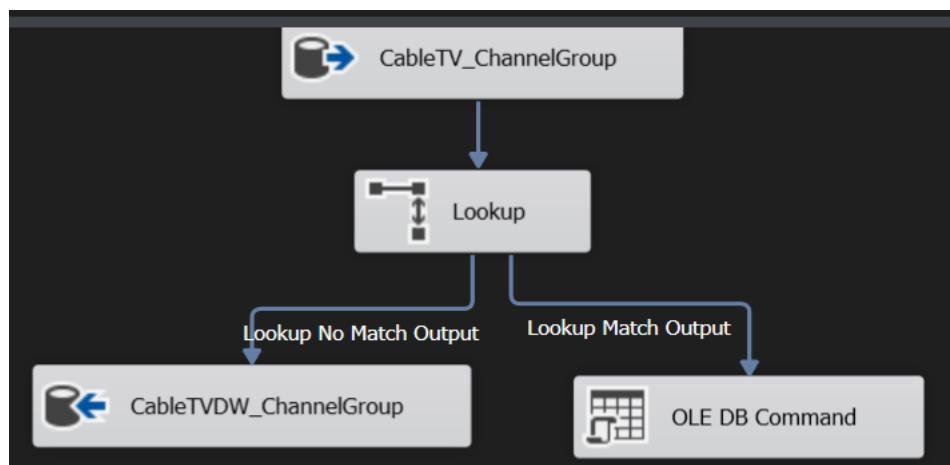


Рис. 2 Таблиця виміру пакетів каналів

Таблиця виміру каналів (DimChannel)

Таблиця виміру каналів містить довідникову інформацію про окремі телеканали, які входять до складу пакетів каналів. Вона використовується для аналізу

структурі пропозиції кабельного телебачення та дослідження взаємозв'язку між каналами і пакетами каналів у межах аналітичних звітів.

Таблиця виміру фільмів (DimMovie)

Таблиця виміру фільмів використовується для зберігання інформації про фільми сервісу відео за запитом. Вона містить дані про назви фільмів, жанри, рік випуску та вартість перегляду. Даний вимір використовується для аналізу популярності фільмів, споживчої активності абонентів і доходів від VOD-сервісу.

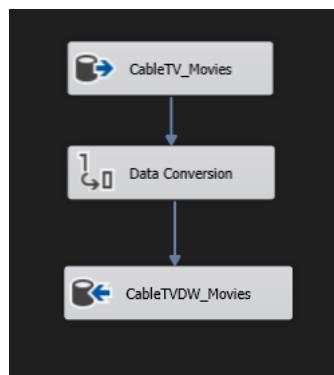


Рис. 3 Таблиця виміру фільмів

Таблиця виміру способів оплати (DimPaymentMethod)

Таблиця виміру способів оплати призначена для зберігання інформації про доступні методи оплати рахунків. Вона використовується для аналізу платіжної дисципліни абонентів, структури платежів та популярності окремих способів оплати.

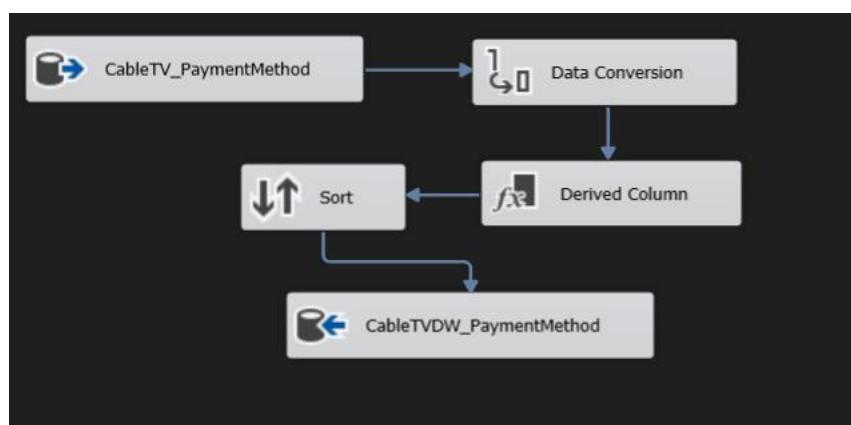


Рис. 4 Таблиця виміру фільмів

Таблиця виміру часу (DimDate)

Таблиця виміру часу використовується для зберігання календарних характеристик, які застосовуються при аналізі даних у часовому розрізі. Вона формується на основі дат підписок, замовень, рахунків і платежів та дозволяє виконувати аналіз за роками, місяцями та іншими часовими інтервалами. Часовий вимір є ключовим елементом сховища даних і використовується у всіх таблицях фактів.

Таблиця фактів підписок (FactSubscriptions)

Таблиця фактів підписок містить інформацію про оформлення та тривалість підписок абонентів на пакети каналів. Вона використовується для аналізу кількості активних підписок, динаміки підключень і відключень, а також доходів від абонентської плати у часовому розрізі.

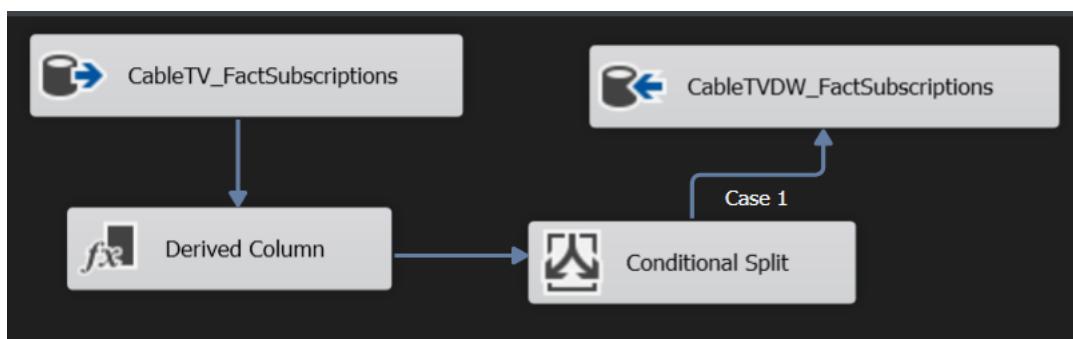


Рис. 5 Таблиця фактів підписок

Таблиця фактів замовлень фільмів (FactOrders)

Таблиця фактів замовлень фільмів відображає операції замовлення фільмів у сервісі відео за запитом. Вона містить числові показники, пов'язані з вартістю замовлень, та ключі вимірів, що дозволяють аналізувати популярність фільмів, активність абонентів і обсяги доходів від VOD.

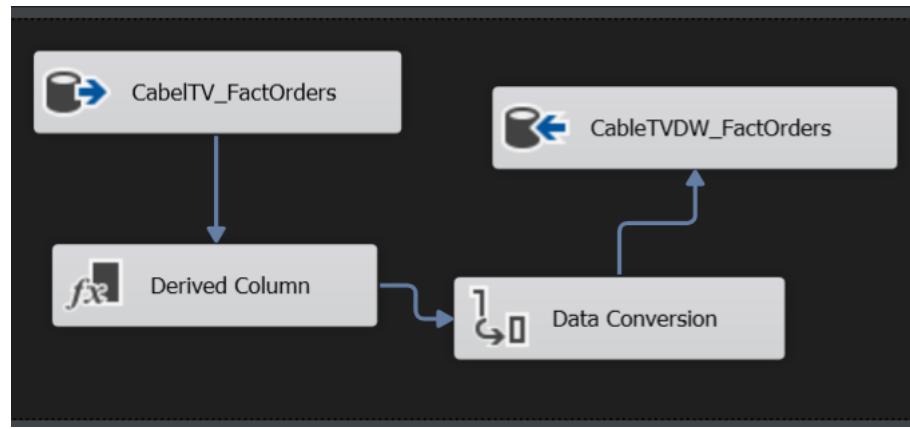


Рис. 6 Таблиця фактів замовлень фільмів

Таблиця фактів рахунків (FactInvoices)

Таблиця фактів рахунків використовується для зберігання інформації про нарахування абонентам за розрахункові періоди. Вона дозволяє аналізувати загальні суми нарахувань, структуру доходів та взаємозв'язок між підписками і фінансовими показниками.

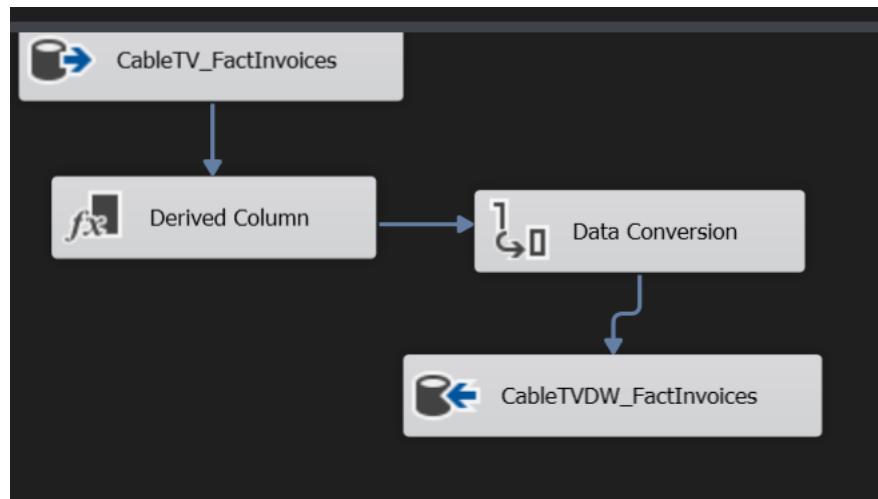


Рис. 7 Таблиця фактів рахунків

Таблиця фактів платежів (FactPayments)

Таблиця фактів платежів містить інформацію про фактичні оплати рахунків абонентами. Вона використовується для аналізу платіжної дисципліни, часткових і повних оплат, а також для оцінки заборгованості у часовому розрізі.

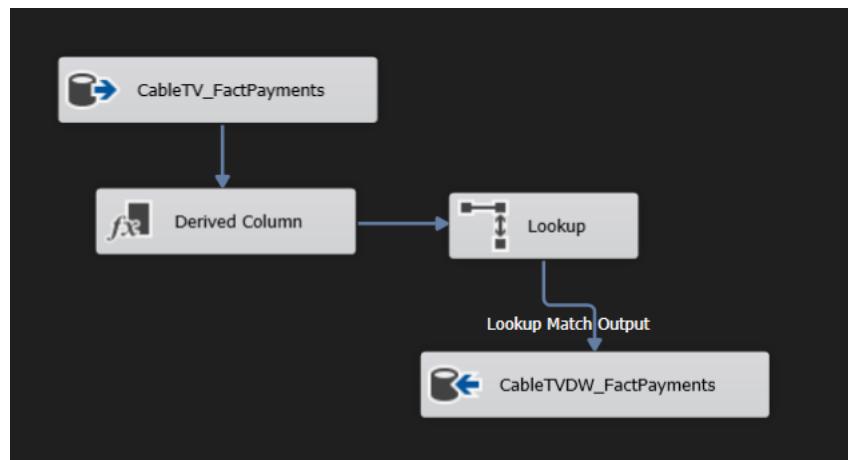


Рис. 8 Таблиця фактів платежів

Розділ 4. Побудова OLAP-куба та аналітичних звітів

4.1 Концепція та обґрунтування побудови OLAP-куб

Спроектоване сховище даних забезпечує централізоване зберігання узгодженої та історичної інформації, підготовленої до аналітичної обробки. Проте для виконання складних багатовимірних аналітичних запитів, швидкого агрегування показників та інтерактивного аналізу даних у різних розрізах необхідне використання спеціалізованих аналітичних інструментів.

З цією метою на основі сховища даних було побудовано OLAP-куб, який дозволяє реалізувати багатовимірну модель аналізу даних. OLAP-куб використовує таблиці фактів і таблиці вимірів сховища даних як джерело інформації, забезпечуючи логічне об'єднання показників та вимірів у єдину аналітичну структуру.

Використання OLAP-куба дозволяє виконувати аналіз показників діяльності системи кабельного телебачення у різних аналітичних розрізах, зокрема за абонентами, пакетами каналів, фільмами, способами оплати та часовими періодами. Такий підхід забезпечує можливість швидко отримувати агреговані дані, аналізувати динаміку показників та виявляти закономірності, які є складними для аналізу за допомогою звичайних SQL-запитів.

Таким чином, сховище даних виконує роль надійного джерела аналітичної інформації, а OLAP-куб — інструмента для її багатовимірної обробки. Побудова OLAP-куба є логічним продовженням етапів проєктування бази даних і реалізації ETL-процесів та створює основу для формування аналітичних звітів і показників, розглянутих у наступному підрозділі.

4.2 Створення вимірів (Dimensions)

1. Вимір часу (DimDate)

Вимір **DimDate** використовується для аналізу показників у часовому розрізі та є ключовим виміром OLAP-куба. Він забезпечує можливість агрегації та фільтрації

фактів за різними календарними характеристиками, що є необхідним для аналізу динаміки підписок, замовлень фільмів, рахунків і платежів у межах визначених періодів.

Основним ключем виміру є **DateKey**, який використовується для зв'язку з таблицями фактів. Атрибут **FullDate** зберігає повну календарну дату та застосовується для коректного відображення значень часу у звітах. Для деталізованого аналізу у вимірі присутні атрибути **Day**, **Month**, **MonthName**, **Quarter** та **Year**, що дозволяє виконувати аналіз на різних рівнях часової деталізації — від дня до року.

Для виміру **DimDate** була створена ієрархія **Calendar**, яка включає рівні *Year* → *Quarter* → *Month* → *Day*. Наявність ієрархії забезпечує зручну навігацію у багатовимірному аналізі та підвищує ефективність виконання агрегованих запитів.

2. Вимір абонентів (**DimSubscriber**)

Вимір **DimSubscriber** призначений для аналізу даних у розрізі абонентів системи кабельного телебачення. Він дозволяє виконувати аналітику підписок, замовлень фільмів та фінансових операцій з урахуванням характеристик клієнтів, а також досліджувати динаміку абонентської бази.

Основним ключем виміру є **SubscriberKey**, який використовується для зв'язку з таблицями фактів. У вимірі зберігаються основні ідентифікаційні та довідникові атрибути абонента, зокрема **FullName**, **Email**, **Address**, **RegistrationDate**. Використання цих атрибутів у звітах дозволяє аналізувати показники у розрізі абонентів та їхнього статусу, а також оцінювати активність клієнтів упродовж часу. У межах даної роботи основний акцент виміру **DimSubscriber** — забезпечення коректного зв'язку фактів із конкретними абонентами та можливість формувати клієнтську аналітику.

3. Вимір фільмів (DimMovie)

Вимір **DimMovie** використовується для аналізу замовлень фільмів у сервісі відео за запитом (VOD). Він дозволяє оцінювати популярність фільмів, аналізувати структуру замовлень та досліджувати доходи від VOD-сервісу у різних розрізах.

Основним ключем виміру є **MovieKey**, який використовується для зв'язку з таблицею фактів замовлень. Атрибут **Title** використовується для ідентифікації фільму у звітах. Для поглибленого аналізу у вимірі використовуються атрибути **Genre**, **ReleaseYear** та **Price** (wartість перегляду), що дозволяє оцінювати попит за жанрами, аналізувати популярність фільмів за роком випуску та досліджувати фінансові показники.

4. Вимір способів оплати (DimPaymentMethod)

Вимір **DimPaymentMethod** використовується для аналізу платежів у розрізі методів оплати. Він дозволяє оцінювати популярність різних способів оплати та аналізувати платіжну дисципліну абонентів залежно від обраного методу.

Основним ключем виміру є **PaymentMethodKey**, який використовується для зв'язку з таблицею фактів платежів. Атрибут **PaymentMethodName** (або аналогічний) зберігає назву способу оплати та використовується у звітах для порівняння структури платежів.

Вимір застосовується переважно у фінансових звітах, де необхідно визначати частку кожного способу оплати у загальній сумі платежів або кількості транзакцій.

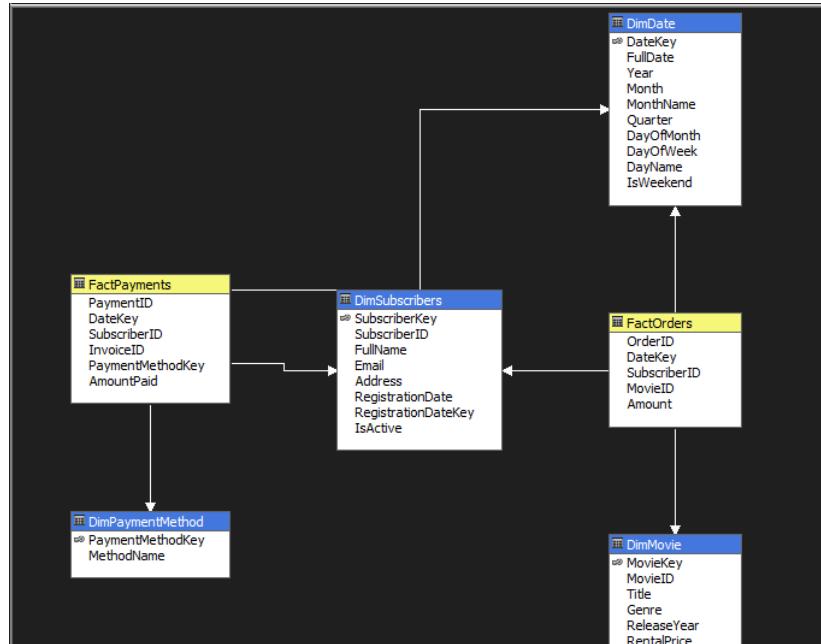


Рис. 9 OLAP-куб

4.3 Створення мір (Measures) та груп мір (Measure Groups)

На етапі проєктування OLAP-куба було створено групи мір та набір аналітичних мір, які забезпечують можливість комплексного аналізу діяльності системи кабельного телебачення. Міри були згруповани відповідно до таблиць фактів сховища даних, що дозволило логічно розділити показники та спростити роботу з аналітичною моделлю.

У межах OLAP-куба було сформовано чотири основні групи мір: **FactSubscriptions**, **FactOrders**, **FactInvoices** та **FactPayments**. Кожна група мір відповідає окремому бізнес-процесу та містить показники, необхідні для його аналізу.

1. Група мір FactSubscriptions

Група мір **FactSubscriptions** використовується для аналізу процесу підключення абонентів до пакетів каналів та тривалості користування послугами. Вона дозволяє оцінювати динаміку підписок, кількість активних абонентів і доходи від абонентської плати.

У межах цієї групи було реалізовано такі основні міри:

- **Subscription Count** — міра типу *Count*, яка використовується для підрахунку кількості підписок;
- **Active Subscriptions Count** — кількісна міра, що дозволяє визначити кількість активних підписок у вибраному періоді;
- **Subscription Revenue** — сумарна міра типу *Sum*, яка відображає загальний дохід від абонентської плати.

Дані міри дозволяють аналізувати структуру підписок у часовому розрізі, порівнювати популярність пакетів каналів та оцінювати фінансову ефективність підписок.

2. Група мір FactOrders

Група мір **FactOrders** призначена для аналізу замовлень фільмів у сервісі відео за запитом (VOD). Вона дозволяє досліджувати споживчу активність абонентів, популярність фільмів та обсяги доходів від додаткових послуг.

До основних мір цієї групи належать:

- **Order Count** — міра типу *Count*, яка використовується для підрахунку кількості замовлень фільмів;
- **Total Order Amount** — сумарна міра типу *Sum*, що відображає загальну вартість замовлень;
- **Average Order Amount** — міра типу *Average*, яка використовується для аналізу середньої вартості одного замовлення.

Використання цих мір дозволяє аналізувати динаміку замовлень, оцінювати популярність фільмів і жанрів, а також визначати частку доходів від VOD-сервісу у загальній структурі доходів.

3. Група мір FactInvoices

Група мір **FactInvoices** використовується для аналізу процесу формування рахунків абонентам. Вона дозволяє оцінювати обсяги нарахувань, структуру доходів та взаємозв'язок між підписками і фінансовими показниками.

У межах цієї групи було створено такі міри:

- **Invoice Count** — міра типу *Count*, що використовується для підрахунку кількості сформованих рахунків;
- **Total Invoice Amount** — сумарна міра типу *Sum*, яка відображає загальну суму нарахувань;
- **Average Invoice Amount** — міра типу *Average*, що дозволяє аналізувати середній розмір рахунку.

Ці міри використовуються для фінансового аналізу, оцінки динаміки доходів та порівняння фінансових показників за різні періоди.

4. Група мір **FactPayments**

Група мір **FactPayments** призначена для аналізу фактичних оплат рахунків абонентами. Вона дозволяє досліджувати платіжну дисципліну, структуру платежів і рівень заборгованості.

До основних мір цієї групи належать:

- **Payment Count** — міра типу *Count*, яка використовується для підрахунку кількості платежів;
- **Total Payment Amount** — сумарна міра типу *Sum*, що відображає загальну суму оплачених коштів;
- **Average Payment Amount** — міра типу *Average*, яка використовується для аналізу середнього розміру платежу.

Використання цих мір дозволяє аналізувати фінансову поведінку абонентів та оцінювати ефективність процесу оплати рахунків.

4.4 Розгортання та процесинг OLAP-куба

Етап розгортання (Deployment) став фінальною стадією фізичної реалізації розробленої багатовимірної моделі. На цьому етапі було виконано передачу метаданих OLAP-куба, включаючи структуру вимірів, ієархій, груп мір та обчислюваних показників, на сервер **SQL Server Analysis Services (SSAS)**.

Під час розгортання на сервері було створено відповідну аналітичну базу даних, яка використовує таблиці сховища даних як джерело інформації. Усі зв'язки між таблицями фактів і вимірів, визначені на етапі проєктування, були коректно перенесені в аналітичне середовище.

Після завершення розгортання було ініційовано процес **Processing**, у ході якого система виконала завантаження та обробку даних для побудови багатовимірної структури. Обробка куба включав такі основні етапи:

1. Вилучення даних.

OLAP-куб звернувся до сховища даних та завантажив інформацію про понад 100 000 абонентів, п'ятирічну історію підписок, а також близько 1 000 000 замовлень фільмів і пов'язані з ними фінансові операції (рахунки та платежі).

2. Агрегація даних.

Сервер Analysis Services виконав розрахунок та збереження агрегованих показників у багатовимірному сховищі типу MOLAP. Було сформовано сумарні, кількісні та середні значення для всіх перетинів вимірів, що дозволяє виконувати аналіз даних у різних аналітичних розрізах без додаткових обчислень на стороні клієнта.

4.4 Аналітичні звіти та візуалізація даних

На основі розгорнутого OLAP-куба було реалізовано набір аналітичних звітів за допомогою **SQL Server Reporting Services (SSRS)**. Розроблені звіти використовують багатовимірну модель даних та забезпечують можливість аналізу діяльності системи кабельного телебачення у різних аналітичних розрізах. Основною метою створення звітів є надання узагальненої та наочної інформації для підтримки управлінських рішень.

У межах курсової роботи було реалізовано три основні аналітичні звіти.

Фінансовий звіт за місяць

Фінансовий звіт за місяць призначений для аналізу надходжень коштів від абонентів у вибраному часовому періоді. Звіт базується на даних таблиці фактів **FactPayments** та виміру **DimDate** і дозволяє оцінити щоденну динаміку платежів у межах конкретного місяця.

У звіті використовується параметр вибору місяця, що забезпечує гнучкість аналізу та можливість перегляду фінансових показників за будь-який період. Основними атрибутами звіту є рік, місяць, повна дата та сума сплачених коштів. Така структура дозволяє відстежувати коливання доходів упродовж місяця та виявляти періоди підвищеної або зниженої платіжної активності.

Фінансовий звіт може використовуватися для контролю платіжної дисципліни абонентів, аналізу стабільності доходів та оцінки ефективності фінансової діяльності системи.

The screenshot shows a software interface for generating reports. At the top, there's a search bar labeled 'Select month' with the value '4' and a 'View Report' button. Below the header are standard navigation buttons for back, forward, and search. The main title of the report is 'Фінансовий звіт за місяць'. The report displays a table with the following data:

Year	Month	Full Date	Amount Paid
2021	4	2021-04-01	104589
2021	4	2021-04-02	110646
2021	4	2021-04-03	98879
2021	4	2021-04-04	115541
2021	4	2021-04-05	103117
2021	4	2021-04-06	96615
2021	4	2021-04-07	104815
2021	4	2021-04-08	99678
2021	4	2021-04-09	102325
2021	4	2021-04-10	110988
2021	4	2021-04-11	106974

Рис. 10 Фінансовий звіт за місяць

Звіт з розподілу фільмів за жанрами

Звіт з розподілу фільмів за жанрами призначений для аналізу структури замовень у сервісі відео за запитом (VOD). Звіт базується на даних таблиці фактів **FactOrders** та виміру **DimMovie** і дозволяє визначити популярність різних жанрів фільмів серед абонентів.

У звіті використовується кругова діаграма, яка наочно відображає частку кожного жанру у загальній кількості замовень. Такий тип візуалізації дозволяє швидко оцінити структуру споживчого попиту та визначити найбільш популярні напрями контенту. Результати даного звіту можуть бути використані для формування контентної політики, оптимізації каталогу фільмів та планування розвитку сервісу відео за запитом.

Розподіл фільмів за жанрами

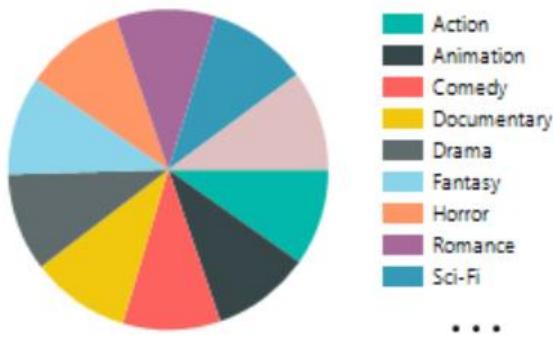


Рис. 11 Розподіл фільмів за жанрами

Звіт зі списку абонентів

Звіт зі списку абонентів призначений для перегляду довідкової інформації про клієнтів системи кабельного телебачення. Він базується на даних виміру **DimSubscriber** та дозволяє отримати інформацію про повне ім'я абонента, електронну адресу, дату реєстрації та пов'язані рахунки.

Звіт подається у табличному вигляді та може використовуватися для аналізу складу абонентської бази, перевірки коректності даних, а також для адміністративних і контрольних цілей. Наявність сортування та навігації забезпечує зручну роботу з великим обсягом записів.

Full Name	Email	Registration Date	Invoice ID
Софія Теліженко	user0_sgerus@example.net	2023-12-20	22688429
Богуслав Гавриш	user1_klyment04@example.com	2024-07-31	18756019
Лілія Панчук	user3_panasty_mchuk@example.net	2024-05-21	28416467
Марина Стус	user4_avreliipel_ekh@example.com	2023-07-10	24518563
Давид Якименко	user6_xiakovenko@example.com	2024-12-02	17010513

Рис. 12 Звіт по списку абонентів

Загальна характеристика аналітичних звітів

Розроблені аналітичні звіти демонструють можливості використання OLAP-куба для вирішення практичних аналітичних задач. Вони охоплюють фінансову, контентну та клієнтську складові діяльності системи кабельного телебачення та дозволяють отримувати узагальнену аналітичну інформацію у зручному для користувача вигляді.

Використання SSRS у поєднанні з багатовимірною моделлю даних забезпечує швидкий доступ до агрегованих показників, наочну візуалізацію результатів та гнучкість аналітичного аналізу.

Висновок

У ході виконання даної курсової роботи було розроблено повнофункціональну аналітичну систему для дослідження діяльності системи кабельного телебачення, яка охоплює всі основні етапи побудови сховища даних, OLAP-куба та аналітичної звітності. Робота виконувалася з урахуванням сучасних підходів до організації аналітичних систем і була орієнтована на практичне використання отриманих результатів для аналізу підписок, фінансових показників та споживчої активності абонентів.

На початковому етапі курсової роботи було виконано аналіз предметної області та визначено ключові бізнес-процеси системи, зокрема реєстрацію абонентів, управління підписками на пакети каналів, замовлення фільмів у сервісі відео за запитом, формування рахунків та облік платежів. На основі цього було сформовано перелік інформаційних потреб і визначено основні аналітичні звіти, зокрема фінансовий звіт за місяць, аналіз розподілу фільмів за жанрами та звіт зі списку абонентів. Це дозволило чітко окреслити вимоги до структури даних і майбутньої аналітичної моделі.

У процесі проєктування реляційної бази даних було створено нормалізовану структуру, що відповідає вимогам третьої нормальної форми. Було визначено основні сутності предметної області, їх атрибути, первинні та зовнішні ключі, а також забезпечені цілісність і узгодженість даних. На основі операційної бази даних було спроектовано сховище даних із використанням зіркової схеми, яка є оптимальною для аналітичних задач. Структура сховища включає таблиці фактів і таблиці вимірів, що забезпечують можливість багатовимірного аналізу даних.

Особливу увагу у роботі було приділено процесу ETL (Extract, Transform, Load), у межах якого здійснювалося витягування даних із джерел, їх очищення, трансформація та завантаження у сховище даних. Під час трансформації застосовувалися типові механізми обробки даних, зокрема приведення типів, обчислення похідних атрибутів, робота з довідниками та перевірка коректності

даних. Це дозволило забезпечити аналітичну придатність даних і коректну роботу сховища при великих обсягах інформації, зокрема для понад 100 000 абонентів, п'ятирічної історії підписок та близько 1 000 000 замовлень фільмів.

У рамках курсової роботи було спроектовано та реалізовано OLAP-куб за допомогою SQL Server Analysis Services. Для куба було створено виміри часу, абонентів, пакетів каналів, каналів, фільмів та способів оплати, а також визначено відповідні ієрархії, що забезпечують зручну навігацію та ефективний багатовимірний аналіз. Побудована модель дозволяє аналізувати фінансові та операційні показники у різних аналітичних розрізах і часовій динаміці.

У кубі було сформовано групи мір відповідно до таблиць фактів, а також реалізовано різні типи мір, зокрема сумарні, кількісні, середні та унікальні показники. Окрім базових мір, було реалізовано обчислювані міри з використанням MDX, включаючи показники Year To Date, Previous Year, відсоткову зміну Year Over Year, а також відносні показники частки у загальному обсязі. Це суттєво розширило аналітичні можливості системи та дозволило виконувати глибокий аналіз динаміки показників у часовому розрізі.

Для зберігання та обробки даних у кубі було використано режим MOLAP, який забезпечує високу продуктивність аналітичних запитів за рахунок попереднього обчислення агрегатів та зберігання даних безпосередньо в Analysis Services. Такий підхід є оптимальним для аналітичних систем і повністю відповідає характеру розробленої моделі.

Завершальним етапом курсової роботи стала розробка аналітичних звітів за допомогою SQL Server Reporting Services. Було реалізовано табличні та графічні звіти з підтримкою параметрів і візуалізацій, які дозволяють аналізувати фінансові показники, структуру замовлень фільмів і дані про абонентів у зручному для користувача вигляді.

У результаті виконання курсової роботи було здобуто практичні навички у сфері проектування та реалізації аналітичних систем, зокрема зі створенням сховищ даних, побудови OLAP-кубів, використання MDX-обчислень та розробки аналітичної звітності. Отримано практичний досвід роботи з інструментами Microsoft SQL Server, Analysis Services та Reporting Services, що є актуальними і широко застосовуються у сучасних інформаційних системах.

Розроблена аналітична система є логічно завершеною, коректно функціонує та повністю відповідає поставленим вимогам. Отримані результати можуть бути використані як основа для подальшого розширення системи, зокрема шляхом збільшення обсягів даних, додавання нових вимірів, удосконалення аналітичних показників або створення додаткових сценаріїв аналізу. Таким чином, курсова робота досягла своєї мети, а отримані знання та навички мають суттєву практичну цінність.

Додаток

SQL-скрипт створення структури операційної бази даних:

```
CREATE TABLE ChannelGroups (
    GroupID INT IDENTITY(1,1) PRIMARY KEY,
    GroupName NVARCHAR(50) NOT NULL UNIQUE,
    MonthlyPrice DECIMAL(10,2) NOT NULL
        CHECK (MonthlyPrice >= 0),
    ParentGroupID INT NULL,
    CONSTRAINT FK_ChannelGroups_Parent
        FOREIGN KEY (ParentGroupID)
        REFERENCES ChannelGroups(GroupID)
);

CREATE TABLE Channels (
    ChannelID INT IDENTITY(1,1) PRIMARY KEY,
    ChannelName NVARCHAR(100) NOT NULL,
    GroupID INT NOT NULL,
    CONSTRAINT FK_Channels_ChannelGroups
        FOREIGN KEY (GroupID)
        REFERENCES ChannelGroups(GroupID)
);
ALTER TABLE Channels
ADD Genre NVARCHAR(50);

CREATE TABLE Subscribers (
    SubscriberID INT IDENTITY(1,1) PRIMARY KEY,
    FullName NVARCHAR(200) NOT NULL,
    Email NVARCHAR(100) NOT NULL UNIQUE,
    Address NVARCHAR(255),
    RegistrationDate DATE NOT NULL DEFAULT GETDATE(),
    IsActive BIT NOT NULL DEFAULT 1
);
CREATE TABLE Subscriptions (
    SubscriptionID INT IDENTITY(1,1) PRIMARY KEY,
    SubscriberID INT NOT NULL,
    GroupID INT NOT NULL,
    StartDate DATE NOT NULL,
    EndDate DATE NULL,
    IsActive BIT NOT NULL DEFAULT 1,
    CONSTRAINT FK_Subscriptions_Subscriber
        FOREIGN KEY (SubscriberID)
        REFERENCES Subscribers(SubscriberID),
    CONSTRAINT FK_Subscriptions_ChannelGroup
        FOREIGN KEY (GroupID)
        REFERENCES ChannelGroups(GroupID),
    CONSTRAINT CK_Subscriptions_Dates
        CHECK (EndDate IS NULL OR EndDate >= StartDate)
);
CREATE TABLE Movies (
    MovieID INT IDENTITY(1,1) PRIMARY KEY,
    Title NVARCHAR(255) NOT NULL,
    Genre NVARCHAR(50),
    ReleaseYear INT
        CHECK (ReleaseYear BETWEEN 1900 AND YEAR(GETDATE())),
    RentalPrice DECIMAL(10,2) NOT NULL
```

```

        CHECK (RentalPrice > 0)
);

CREATE TABLE Orders (
    OrderID INT IDENTITY(1,1) PRIMARY KEY,
    SubscriberID INT NOT NULL,
    MovieID INT NOT NULL,
    OrderDate DATETIME NOT NULL DEFAULT GETDATE(),
    Amount DECIMAL(10,2) NOT NULL
        CHECK (Amount > 0),

    CONSTRAINT FK_Orders_Subscriber
        FOREIGN KEY (SubscriberID)
        REFERENCES Subscribers(SubscriberID),

    CONSTRAINT FK_Orders_Movie
        FOREIGN KEY (MovieID)
        REFERENCES Movies(MovieID)
);

CREATE TABLE Invoices (
    InvoiceID INT IDENTITY(1,1) PRIMARY KEY,
    SubscriberID INT NOT NULL,
    InvoiceDate DATE NOT NULL,
    TotalAmount DECIMAL(10,2) NOT NULL
        CHECK (TotalAmount >= 0),
    DueDate DATE NOT NULL,
    IsPaid BIT NOT NULL DEFAULT 0,

    CONSTRAINT FK_Invoices_Subscriber
        FOREIGN KEY (SubscriberID)
        REFERENCES Subscribers(SubscriberID)
);

CREATE TABLE Payments (
    PaymentID INT IDENTITY(1,1) PRIMARY KEY,
    InvoiceID INT NOT NULL,
    PaymentDate DATETIME NOT NULL DEFAULT GETDATE(),
    AmountPaid DECIMAL(10,2) NOT NULL
        CHECK (AmountPaid > 0),
    PaymentMethod NVARCHAR(50) NOT NULL,

    CONSTRAINT FK_Payments_Invoice
        FOREIGN KEY (InvoiceID)
        REFERENCES Invoices(InvoiceID)
);

CREATE INDEX IX_Subscriptions_Active
    ON Subscriptions (SubscriberID, IsActive);

CREATE INDEX IX_Orders_OrderDate
    ON Orders (OrderDate);

CREATE INDEX IX_Invoices_InvoiceDate
    ON Invoices (InvoiceDate);

```

Пайтон код для генерації даних для Invoice:

```

import pandas as pd
import random
from datetime import timedelta
from sqlalchemy import create_engine
import urllib

```

```

SERVER_NAME = 'DESKTOP-A95Q7PS'
DATABASE_NAME = 'CableTV_OLTP'

params = urllib.parse.quote_plus(
    f"DRIVER={{ODBC Driver 17 for SQL Server}};" +
    f"SERVER={SERVER_NAME};" +
    f"DATABASE={DATABASE_NAME};" +
    f"Trusted_Connection=yes;"
)

engine = create_engine(f"mssql+pyodbc:///?odbc_connect={params}")
print(" Connected to DB")

subscriptions_df = pd.read_sql("""
    SELECT
        s.SubscriptionID,
        s.SubscriberID,
        s.StartDate,
        g.MonthlyPrice
    FROM Subscriptions s
    JOIN ChannelGroups g ON s.GroupID = g.GroupID
    WHERE s.IsActive = 1
""", engine)

print(f"Active subscriptions: {len(subscriptions_df)}")
orders_df = pd.read_sql("""
    SELECT
        SubscriberID,
        OrderDate,
        Amount
    FROM Orders
""", engine)

orders_df["OrderMonth"] = orders_df["OrderDate"].dt.to_period("M")

orders_monthly = (
    orders_df
    .groupby(["SubscriberID", "OrderMonth"])["Amount"]
    .sum()
    .reset_index()
)
END_DATE = pd.Timestamp.today().to_period("M")
START_DATE = END_DATE - 59

all_months = pd.period_range(
    start=START_DATE,
    end=END_DATE,
    freq="M"
)
duration_choices = [
    (60, 0.30),
    (36, 0.30),
    (24, 0.20),
    (18, 0.20)
]
invoices_data = []

for _, sub in subscriptions_df.iterrows():
    active_months = random.choices(
        [d[0] for d in duration_choices],
        weights=[d[1] for d in duration_choices],
        k=1
    )[0]

    start_month = pd.to_datetime(sub["StartDate"]).to_period("M")

```

```

valid_months = all_months[all_months >= start_month][:active_months]

for month in valid_months:
    movie_sum = orders_monthly[
        (orders_monthly["SubscriberID"] == sub["SubscriberID"]) &
        (orders_monthly["OrderMonth"] == month)
    ]["Amount"].sum()

    total = sub["MonthlyPrice"] + movie_sum

    invoices_data.append({
        "SubscriberID": sub["SubscriberID"],
        "InvoiceDate": month.to_timestamp(),
        "TotalAmount": round(total, 2),
        "IsPaid": 0,
        "DueDate": (month + 1).to_timestamp()
    })
df_invoices = pd.DataFrame(invoices_data)
print(f"Generated invoices: {len(df_invoices)}")

df_invoices.to_sql(
    name="Invoices",
    con=engine,
    if_exists="append",
    index=False,
    chunksize=1000
)

print("Invoices inserted")

```