Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра ПІ

Дисципліна «Розробка хмарних застосувань в Azure»

Лабораторна робота №2

«РОЗРОБКА БД ХМАРНОГО ЗАСТОСУВАННЯ»

Виконала:

ст. гр. ПЗПІ-22-7

Бистрицька А.І

Прийняла:

Доц. кафедри ПІ

Кравець Н.С

Харків – 2025

Мета: Метою даної роботи є придбання практичних навичок розробки та масштабування БД, використання сховищ даних хмарного застосування, що буде реалізовано у Microsoft Azure.

Хід роботи

1. Базуючись на проекті, що розробили у 1-й лабораторній роботі визначте які базу даних (та сховище даних) ви будете використовувати для свого додатку. Обґрунтуйте свій вибір базуючись на класифікації даних (як структурованих, частково структурованих і неструктурованих) та способах використання даних у вашому додатку.

У рамках реалізації хмарного застосунку для бронювання консультацій було прийнято обґрунтоване рішення щодо використання різних сервісів зберігання даних, враховуючи їхню структуру та функціональні особливості. Для роботи зі структурованими даними, зокрема інформацією про користувачів (студентів, викладачів, адміністраторів), розклади консультацій та записи на них, було обрано Azure SQL Database. Цей сервіс є оптимальним для реалізації реляційної моделі даних, оскільки забезпечує підтримку транзакцій, відповідність вимогам ACID, а також надійне управління зв’язками типу «один до багатьох» між таблицями Users, Teachers, Slots і Bookings.

Для зберігання частково структурованих даних, таких як журнали дій користувачів і аналітична інформація, використовується Azure Cosmos DB з API для MongoDB або SQL. Цей варіант було обрано через гнучкість у роботі з даними у форматі JSON, високу швидкодію при операціях читання/запису, а також можливість масштабування на глобальному рівні. У межах цього сервісу зберігаються колекції Logs (системні події) та Analytics (звіти про навантаження викладачів).

Для роботи з неструктурованими даними, зокрема файлами навчальних матеріалів у форматах PDF, DOCX, а також вкладеннями до електронних сповіщень, використовується Azure Blob Storage. Це рішення дозволяє ефективно зберігати великі обсяги файлів, забезпечує контроль доступу до них та є економічно доцільним порівняно з використанням традиційних баз даних. У рамках цього сервісу створено контейнери materials і notifications. Загалом, обрана модель зберігання даних дозволяє забезпечити ефективну, масштабовану та надійну роботу хмарного застосунку відповідно до поставлених вимог.

2. Які варіанти зберігання даних у Azure найбільш підходять для великих даних?

У процесі вибору технологій для зберігання даних у хмарному застосунку було детально проаналізовано переваги та недоліки кожного з обраних сервісів Microsoft Azure, відповідно до характеру даних та специфіки їх використання.

- **Azure SQL Database**

Цей сервіс було обрано для зберігання структурованих реляційних даних, зокрема інформації про користувачів та консультації. Серед основних переваг варто виділити надійність, оскільки база даних повністю підтримує транзакції з дотриманням принципів ACID, що є критично важливим для цілісності даних під час бронювання. Інтеграція з іншими сервісами Azure, такими як App Service і Azure Active Directory, спрощує побудову повноцінної архітектури. Також значною перевагою є автоматизація — у сервіс вбудовано механізми резервного копіювання та масштабування. Водночас слід враховувати і недоліки, зокрема обмежену гнучкість при роботі з напівструктурованими даними (наприклад, JSON) та зростання вартості при збільшенні обсягу даних.

- **Azure Cosmos DB**

Цей сервіс було використано для зберігання частково структурованих даних, таких як події системи та аналітика. Його головна перевага — гнучкість, адже він підтримує різні моделі даних (документи, графи, пари «ключ–значення»). Завдяки глобальному масштабуванню Cosmos DB забезпечує низьку затримку доступу до даних у будь-якому регіоні. Продуктивність також знаходиться на високому рівні завдяки автоматичній індексації, яка пришвидшує виконання запитів. Проте сервіс має і недоліки: його вартість може бути досить високою при інтенсивному використанні, а для досягнення максимальної ефективності потрібно здійснити додаткові налаштування (наприклад, оптимізацію RU/s, партиціювання тощо).

- **Azure Blob Storage**

Цей сервіс обрано для зберігання неструктурованих даних, зокрема файлів матеріалів консультацій. Його основна перевага — економічність, адже зберігання великих обсягів даних тут значно дешевше, ніж у реляційних базах. Blob Storage також відзначається високою доступністю та легко інтегрується через REST API або відповідні SDK. Крім того, підтримуються різні рівні доступу до файлів (гарячий, прохолодний, архівний), що дозволяє оптимізувати витрати залежно від частоти звернень до даних. Водночас варто зазначити деякі обмеження, зокрема відсутність підтримки складних запитів до вмісту файлів, а також необхідність підключення додаткових сервісів (наприклад, Azure Functions) для обробки та обробки файлів.

3. Розробіть схему БД вашого додатку (та опишіть дані, які будуть зберігатися у сховищі).

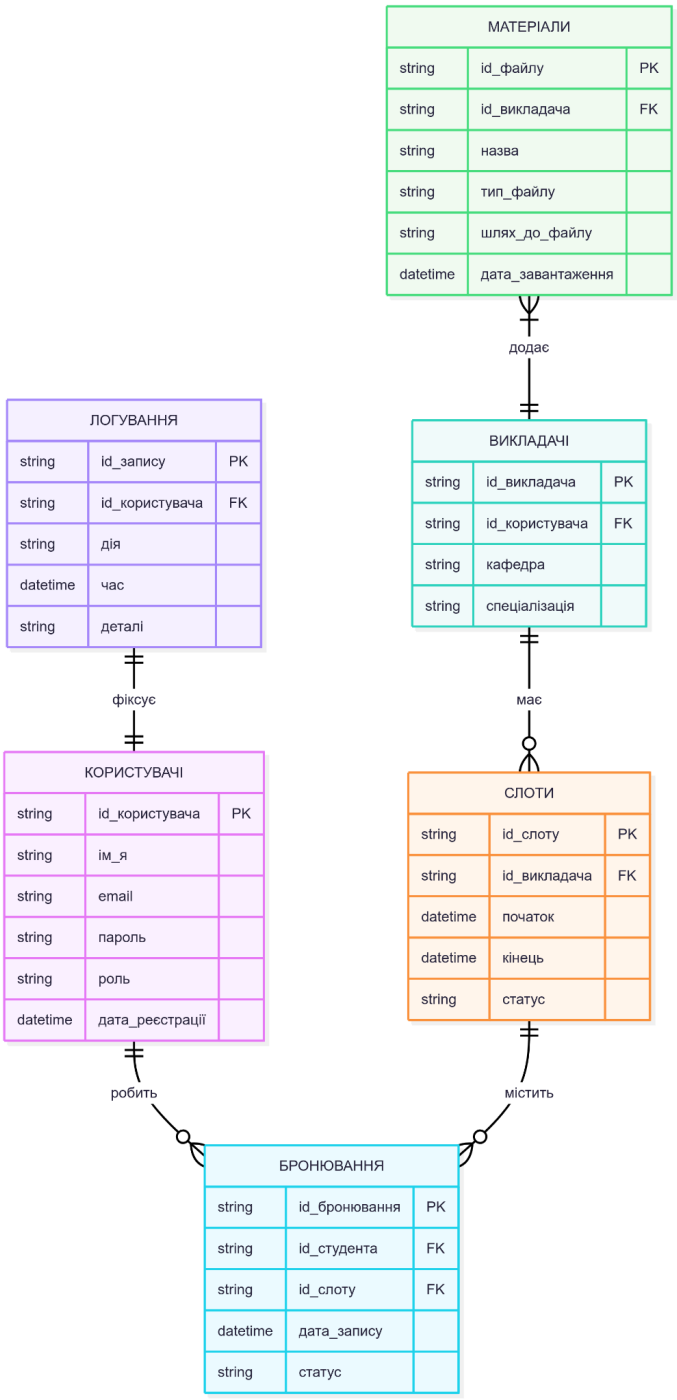

Рис 2.1 - ER - діаграма бази даних

Запропонована ER-модель бази даних, відображає логічну структуру даних хмарного додатка, розробленого для організації консультацій у навчальному закладі. Модель є результатом попереднього аналізу функціональних вимог системи та враховує основні принципи побудови реляційних баз даних, зокрема нормалізацію до третьої нормальної форми (3NF), що забезпечує відсутність транзитивних залежностей та використання атомарних атрибутів.

У центрі моделі знаходиться уніфікована сутність "Користувачі", яка акумулює базову інформацію про всіх учасників системи — студентів, викладачів і адміністраторів. Кожен користувач має унікальний ідентифікатор, персональні дані (ім’я, електронна адреса), зашифрований пароль, роль у системі та дату реєстрації. Рольовий підхід дозволяє реалізувати механізми авторизації й обмеження доступу відповідно до функціональних повноважень користувачів.

На основі сутності "Користувачі" побудовано спеціалізоване розширення — "Викладачі", яке зберігає додаткову інформацію про академічний профіль викладача, зокрема кафедру та спеціалізацію. Між цими сутностями встановлено зв’язок типу "один до одного", що забезпечує логічне відокремлення загальних та специфічних атрибутів.

Організація розкладу консультацій представлена сутністю "Слоти", яка фіксує часові межі консультацій, статус доступності, а також містить зовнішній ключ до викладача, який проводить консультацію. Записи на консультації реалізуються через сутність "Бронювання", що моделює відношення між студентом і конкретним слотом. Кожне бронювання містить інформацію про статус підтвердження та дату створення, що дозволяє вести облік та контроль над виконанням записів.

Навчальні матеріали, пов’язані з консультаціями, представлені окремою сутністю "Матеріали". Вона містить метадані файлів, дату завантаження та шлях

зберігання у службі Azure Blob Storage. Додавання матеріалів виконується викладачами, про що свідчить відповідний зв’язок між цими сутностями.

Для забезпечення аудиту та контролю активності користувачів у системі використовується сутність "Логування", яка фіксує типи операцій, опис подій та точний час їх виконання. Зв’язок цієї сутності з користувачами дозволяє здійснювати повне трасування дій, що є важливим елементом інформаційної безпеки.

Архітектурна реалізація даної моделі передбачає розміщення основних структурованих сутностей у реляційній базі даних Azure SQL Database. Напівструктуровані дані з журналу подій зберігаються у Azure Cosmos DB, що дозволяє забезпечити масштабованість і високу швидкодію при аналізі подій. Файли з навчальними матеріалами, відповідно до своєї природи, розміщуються у Azure Blob Storage, що є ефективним та економічно доцільним рішенням для зберігання неструктурованих

4. Обрати метод масштабування БД. Обґрунтуйте свій вибір.

У ході проєктування хмарного застосунку для бронювання консультацій було здійснено детальний аналіз вимог до масштабування системи, з урахуванням специфіки її використання та характеру навантажень. Основною особливістю даного типу застосунку є нерівномірність навантаження: інтенсивність використання системи значно зростає в періоди перед початком семестру або під час сесій, коли студенти масово записуються на консультації. Крім того, з часом відбувається накопичення історичних даних — записів бронювання, навчальних матеріалів і логів, що також вимагає відповідної адаптації інфраструктури. Не менш важливою є вимога до високої доступності, оскільки система має залишатися стабільною і функціональною навіть за умов пікового навантаження.

У процесі оцінки можливих стратегій масштабування було розглянуто два основні підходи — вертикальне та горизонтальне масштабування. Вертикальне масштабування, яке передбачає підвищення ресурсної потужності одного сервера (шляхом збільшення обсягу оперативної пам’яті, кількості ядер процесора чи дискового простору), характеризується відносною простотою реалізації та підходить для обробки помірного навантаження. Втім, цей підхід має ряд обмежень: він упирається у фізичні межі можливостей обладнання, потребує значних фінансових витрат у разі переходу до високопродуктивних конфігурацій, а також може спричинити простої системи під час оновлення інфраструктури.

На відміну від цього, горизонтальне масштабування полягає у додаванні додаткових серверів та розподілі навантаження між ними. Залежно від обраної стратегії, можуть застосовуватися механізми шардування (розподілу даних за певною логікою, наприклад, по факультетах) або реплікації (створення копій даних для зменшення навантаження на головний вузол). Горизонтальне масштабування забезпечує лінійне зростання продуктивності системи, не обмежене фізичними ресурсами одного сервера, а також сприяє досягненню високої доступності. Разом із тим, впровадження такого підходу потребує складнішої архітектурної реалізації та використання додаткових програмних засобів для координації розподілу навантаження.

З огляду на вищезазначені особливості, для реалізації масштабування у розроблюваному застосунку було обрано комбінований підхід із використанням вбудованих можливостей платформи Microsoft Azure. У контексті Azure SQL Database реалізовано горизонтальне масштабування за рахунок реплікації читання: один головний сервер відповідає за запис даних, а до чотирьох реплік — за виконання операцій читання. Такий підхід дозволяє рівномірно розподіляти навантаження між вузлами без зниження продуктивності. Крім того, застосовано модель автоматичного вертикального масштабування на основі DTU (Database

Transaction Units), що забезпечує автоматичне збільшення обчислювальних ресурсів у разі зростання навантаження.

Для Azure Cosmos DB, який використовується для зберігання частково структурованих даних, масштабування реалізовано за допомогою автоматичного шардування та підтримки глобального розподілення даних. Гнучка модель керування пропускною спроможністю (вимірюваної в RU/s — Request Units per second) дозволяє адаптувати продуктивність бази до поточних потреб без потреби зупинки системи. Зокрема, передбачено шардування даних за ідентифікатором викладача, що дозволяє ефективно балансувати навантаження між розділами даних.

Що стосується Azure Blob Storage, він надає можливість автоматичного масштабування обсягів зберігання без обмежень. Завдяки використанню різних рівнів доступу — наприклад, "холодного" для архівних матеріалів — досягається оптимізація витрат на зберігання даних. Крім того, для прискорення завантаження найбільш затребуваних файлів можлива інтеграція з Content Delivery Network (CDN), що сприяє підвищенню швидкості доступу до ресурсів.

Запропонований підхід має низку суттєвих переваг. По-перше, він забезпечує еластичність — система автоматично адаптується до змін у навантаженні, що особливо важливо в умовах сезонних піків. По-друге, відмовостійкість досягається за рахунок геореплікації даних у рамках глобальної інфраструктури Azure. По-третє, модель є економічно ефективною, оскільки передбачає оплату лише за реально використані ресурси, що знижує витрати на підтримку інфраструктури. І нарешті, управління масштабуванням значно спрощується завдяки інтегрованим інструментам Azure, які дозволяють здійснювати моніторинг, аналіз продуктивності та адаптацію системи в режимі реального часу.

5. Обрати метод захисту даних. Обґрунтуйте свій вибір.

- ***захист від зовнішніх атак;***
- ***захист даних;***
- ***виявлення в своїх мережах загроз безпеці.***

Метод шифрування даних було обрано як основний засіб захисту інформації у системі через його здатність забезпечувати високий рівень конфіденційності та цілісності даних на всіх етапах їх обробки. Шифрування гарантує, що навіть у разі несанкціонованого доступу або перехоплення інформації, її зміст залишатиметься недоступним для сторонніх осіб. Застосування технологій Transparent Data Encryption для захисту даних у стані спокою та протоколів TLS/SSL для захисту даних під час передачі відповідає сучасним стандартам інформаційної безпеки та нормативним вимогам. Таким чином, цей метод забезпечує комплексний захист від зовнішніх атак і витоків інформації, що є критично важливим для підтримання довіри користувачів і стабільної роботи системи.

6. Створити заповнити та продемонструвати працездатність БД (та сховища даних) в Azure із реалізацією обраних методів масштабування та захисту даних. Вимоги до ступеню реалізації вашого додатка: 1) додаток повинен існувати, 2) додаток повинен запускатися без помилок.

У рамках реалізації хмарного застосунку було створено базу даних *ConsultationDB* за допомогою сервісу Azure SQL Database. Для налаштування було використано підписку Azure for Students та обрано відповідні параметри: логічний сервер, регіон розміщення, автентифікацію, а також базовий тарифний план, придатний для цілей навчального проєкту. Базу даних створено успішно, що підтверджує її наявність у ресурсній групі середовища Azure.

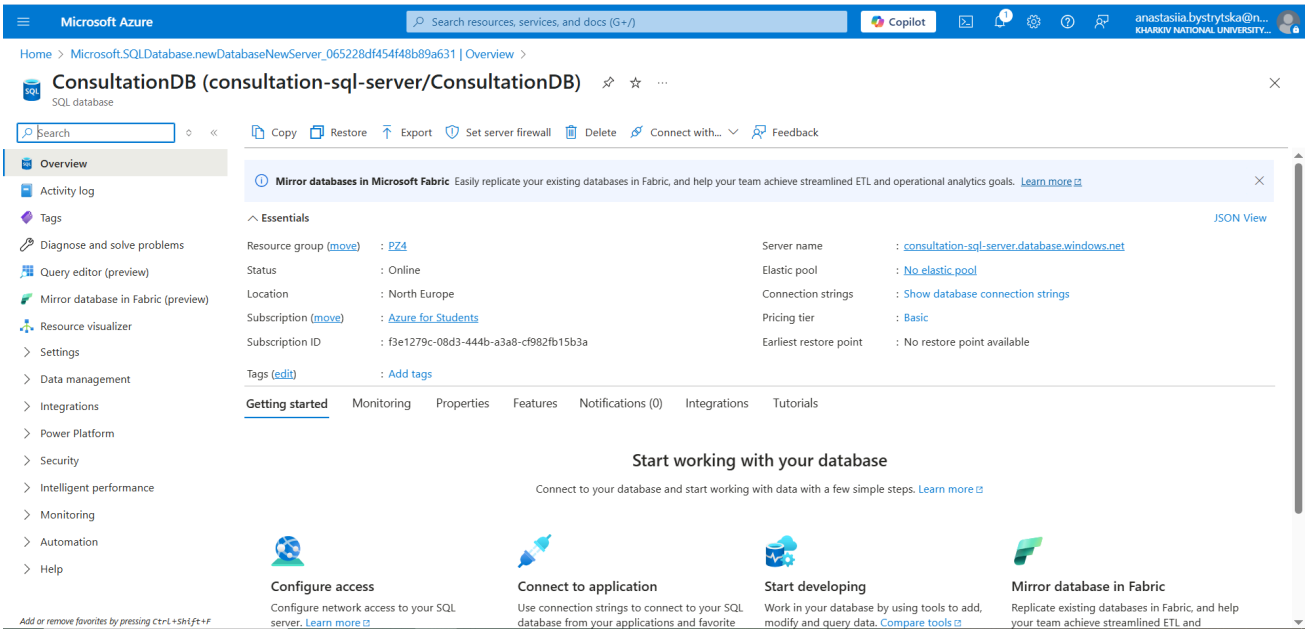

Рис 3.2 - Створення бази даних ConsultationDB у хмарному середовищі Microsoft Azure

Для підтвердження успішного створення структури бази даних було виконано SQL-запит, який виводить список усіх таблиць у базі даних ConsultationDB. Це допомагає переконатися, що всі необхідні таблиці (Users, Teachers, Slots, Bookings, Materials, Logs) були створені правильно і доступні для роботи.

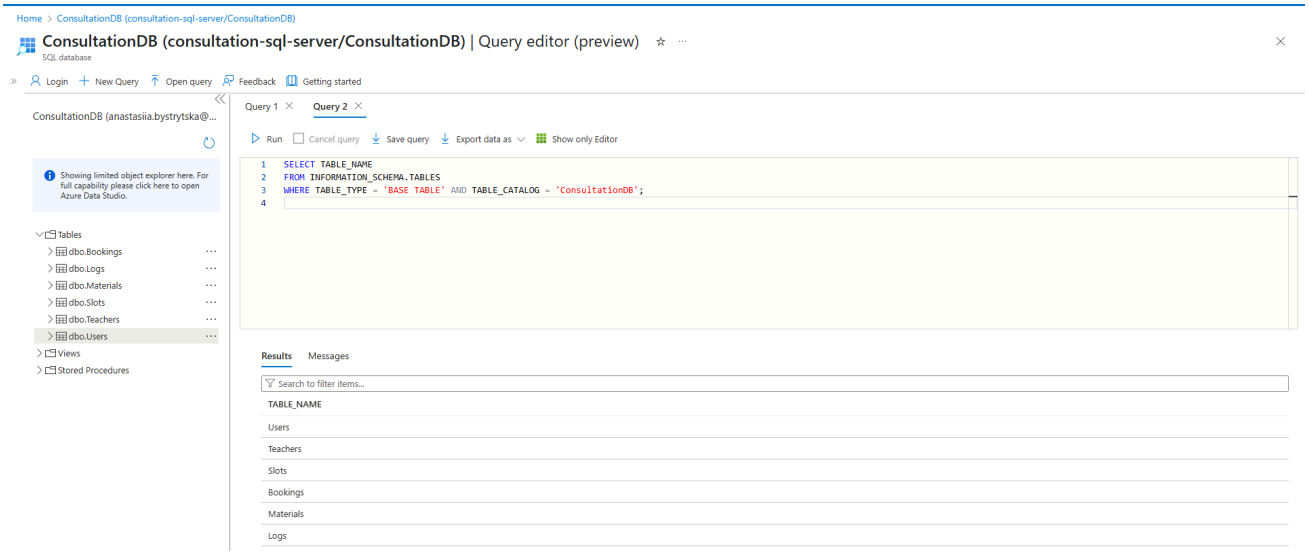

Рис 3.3 - Вивід списку таблиць бази даних ConsultationDB після створення структури

SQL - запит для створення таблиць:

```
CREATE TABLE Users (
    user_id NVARCHAR(50) PRIMARY KEY,
    first_name NVARCHAR(100),
    email NVARCHAR(100) NOT NULL UNIQUE,
    password NVARCHAR(255) NOT NULL,
    role NVARCHAR(50),
    registration_date DATETIME2 DEFAULT SYSUTCDATETIME()
);

CREATE TABLE Teachers (
    teacher_id NVARCHAR(50) PRIMARY KEY,
    user_id NVARCHAR(50) NOT NULL,
    department NVARCHAR(100),
    specialization NVARCHAR(100),
    CONSTRAINT FK_Teachers_Users FOREIGN KEY (user_id) REFERENCES
Users(user_id) ON DELETE CASCADE
);

CREATE TABLE Slots (
    slot_id NVARCHAR(50) PRIMARY KEY,
    teacher_id NVARCHAR(50) NOT NULL,
    start_time DATETIME2 NOT NULL,
    end_time DATETIME2 NOT NULL,
    status NVARCHAR(50),
    CONSTRAINT FK_Slots_Teachers FOREIGN KEY (teacher_id) REFERENCES
Teachers(teacher_id) ON DELETE CASCADE
);

CREATE TABLE Bookings (
    booking_id NVARCHAR(50) PRIMARY KEY,
    student_id NVARCHAR(50) NOT NULL,
    slot_id NVARCHAR(50) NOT NULL,
    booking_date DATETIME2 DEFAULT SYSUTCDATETIME(),
    status NVARCHAR(50),
    CONSTRAINT FK_Bookings_Users FOREIGN KEY (student_id) REFERENCES
Users(user_id) ON DELETE CASCADE,
    CONSTRAINT FK_Bookings_Slots FOREIGN KEY (slot_id) REFERENCES
Slots(slot_id) ON DELETE NO ACTION
);

CREATE TABLE Materials (
    file_id NVARCHAR(50) PRIMARY KEY,
    teacher_id NVARCHAR(50) NOT NULL,
    name NVARCHAR(255) NOT NULL,
```

```
    file_type NVARCHAR(50) NOT NULL,
    file_path NVARCHAR(255) NOT NULL,
    upload_date DATETIME2 DEFAULT SYSUTCDATETIME(),
    CONSTRAINT FK_Materials_Teachers FOREIGN KEY (teacher_id) REFERENCES
Teachers(teacher_id) ON DELETE CASCADE
);

CREATE TABLE Logs (
    log_id NVARCHAR(50) PRIMARY KEY,
    user_id NVARCHAR(50) NOT NULL,
    action NVARCHAR(255) NOT NULL,
    timestamp DATETIME2 DEFAULT SYSUTCDATETIME(),
    details NVARCHAR(MAX),
    CONSTRAINT FK_Logs_Users FOREIGN KEY (user_id) REFERENCES
Users(user_id) ON DELETE CASCADE
);
```

Для підтвердження заповнення таблиць тестовими даними було виконано SQL-запит, який рахує кількість записів у кожній таблиці бази даних ConsultationDB. Результат демонструє, що таблиці містять дані, необхідні для подальшої роботи додатка.

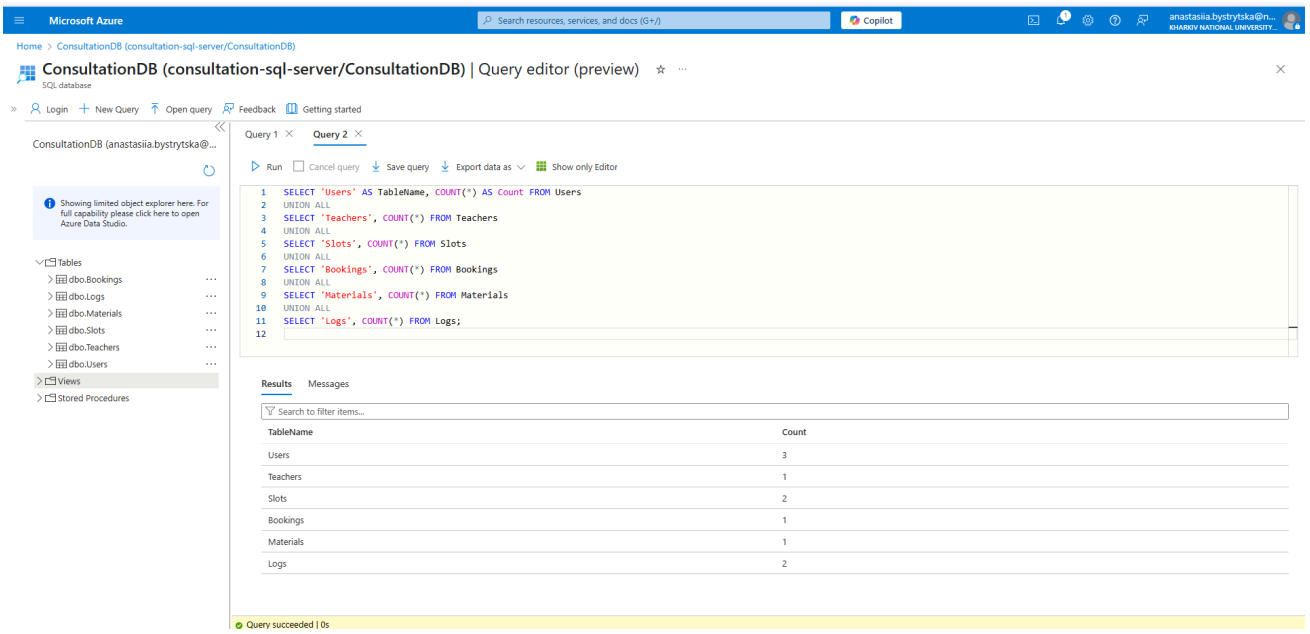

Рис 3.4 - Результат виконання SQL-запиту для перевірки кількості записів у таблицях бази даних ConsultationDB.

Для підтвердження коректності зв’язків між таблицями і логіки роботи бази даних було виконано складний SQL-запит, який об’єднує інформацію про студентів, викладачів, слоти бронювання, статус бронювання, матеріали викладачів та загальну кількість бронювань на кожен слот. Запит демонструє, що

дані коректно пов’язані і можуть використовуватися для розширеного аналітичного звіту.

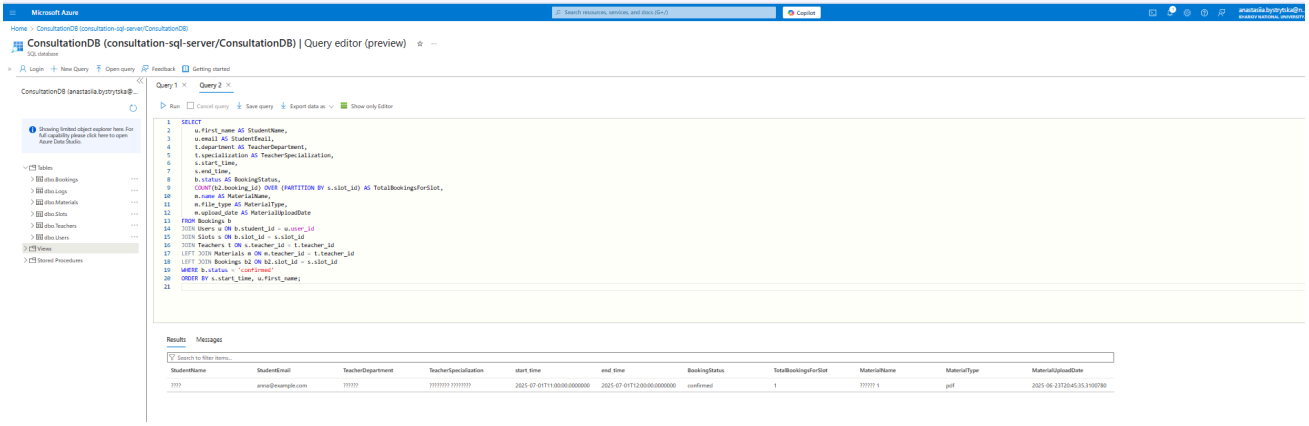

Рис 3.5 - Результат виконання складного JOIN-запиту, що демонструє інформацію про підтверджені бронювання, матеріали викладача та статистику бронювань у базі даних ConsultationDB.

Висновки:У ході виконання роботи було розроблено та реалізовано структуру реляційної бази даних для системи бронювання консультацій. Було створено основні таблиці, визначено зв’язки між ними через зовнішні ключі, заповнено тестовими даними. За допомогою складних SQL-запитів перевірено коректність зв’язків та логіку роботи бази даних. Робота демонструє здатність забезпечувати надійне збереження, обробку та отримання інформації, що є основою для подальшої розробки додатку з функціоналом бронювання та управління матеріалами.

Контрольні запитання:

1. **Які дані відносяться до структурованих, частково структурованих, неструктурованих?**

Структуровані дані — це інформація, яка зберігається в чітко визначених таблицях з фіксованою схемою, наприклад, у реляційних базах даних. Частково структуровані дані мають певну організацію, але не жорстку схему, як у випадку з JSON або XML файлами. Неструктуровані дані — це

різного роду файли без чіткої структури, наприклад, зображення, відео чи текстові документи.

2. Які варіанти зберігання даних у Azure найбільш підходять для великих даних?

Для зберігання великих обсягів даних в Azure найбільш підходять рішення типу Data Lake Storage, яке дозволяє зберігати різні типи даних у великому масштабі, Blob Storage для неструктурованих даних, а також аналітичні платформи на кшталт Synapse Analytics, що забезпечують потужний аналіз великих даних.

3. Які варіанти зберігання даних у Azure найбільш підходять для бізнес даних, що потребують виконання складних аналітичних запитів?

Якщо ж мова йде про бізнес-дані, де потрібні складні аналітичні запити, то Azure Synapse Analytics або Azure SQL Data Warehouse є найбільш ефективними, оскільки вони спеціалізовані на обробці складних запитів і аналітиці.

4. Які варіанти зберігання даних у Azure найбільш підходять для даних додатку з жорсткими вимогами до швидкості виконання запитів?

Коли додаток потребує дуже швидкого виконання запитів, варто звернути увагу на Azure SQL Database з планами високої продуктивності або Azure Cosmos DB, яка пропонує глобальну розподіленість і низьку затримку. Для додаткового прискорення часто використовують кешування через Azure Cache for Redis.

5. Назвіть методи масштабування БД у Azure.

Щодо масштабування баз даних у Azure, існують два основні підходи: вертикальне масштабування, коли збільшуються ресурси окремого сервера, і горизонтальне масштабування, яке передбачає додавання нових екземплярів бази

для розподілу навантаження. Крім того, для ефективного використання ресурсів застосовують еластичні пули, де кілька баз даних ділять ресурси, а також реплікацію для читання, що зменшує навантаження на основну базу.

6. Назвіть методи захисту даних у Azure.

Для захисту даних в Azure використовуються різні методи, серед яких шифрування даних у стані спокою за допомогою Transparent Data Encryption, а також шифрування при передачі через протоколи TLS або SSL. Важливу роль відіграють брандмауери і мережеві правила, які контролюють доступ, а також система ролей (RBAC) для керування правами користувачів. Окрім цього, здійснюється аудит і моніторинг активності для виявлення підозрілих дій, а також регулярно робляться резервні копії для можливості відновлення даних у разі потреби.