# Вимоги

Назва файлу: Номер практичної роботи + \_ + Група + \_ + Прізвище + \_ + Ініціали.docx

В іменах файлів використовуємо тільки латинський алфавіт.

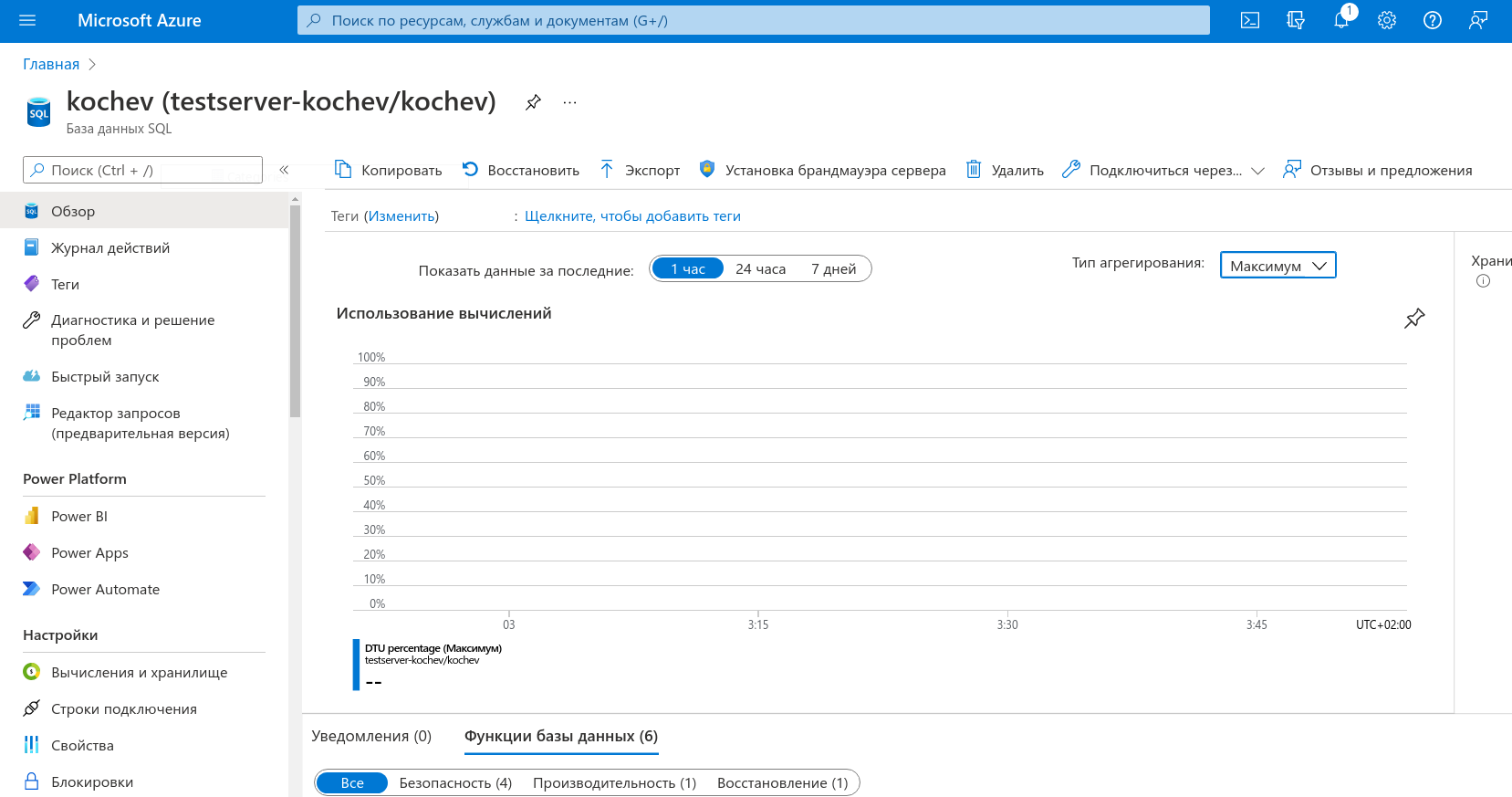
Приклад: 5\_IP96\_Nedashkivskyi\_YA.docx

Відповідь пишеться відразу після завдання курсивом. Питання не видаляються.

# Завдання:

Використовуючи Azure SQL Database\*:

1. Створити базу даних, де назвою буде ваше прізвище. Впевнитись, що кодування бази даних підтримує українську мову.



1. Створити таблицю music, що матиме колонки з назвою виконавця, пісні, альбому та року релізу, самостійно обравши типи даних.

*CREATE TABLE music (*

*singer\_name VARCHAR(50),*

*song VARCHAR(50),*

*album VARCHAR(50),*

*release\_year INT*

*);*

1. Відредагувати створену таблицю додавши нову колонку з ідентифікатором, що буде первинним ключем та матиме властивість автоінкременту.

*ALTER TABLE music*

*ADD music\_id INT IDENTITY PRIMARY KEY;*

1. Записати в таблицю music п’ять рядкова з піснями, що ви зараз слухаєте найчастіше. Серед них повинна бути хоча б одна з назвою чи виконавцем українською мовою.

*INSERT INTO music (singer\_name, song, album, release\_year)*

*VALUES*

*('Kino', 'Stuk', 'Zvezda', 1989),*

*('Imagine Dragons', 'Radioactive', 'Night Visions', 2012),*

*('Lyapis-98', 'Kapital', 'Kapital', 2007),*

*('DDT', 'Veter', 'Eto vse', 1995),*

*('Павло Тичина', 'Державний гімн Української Радянської Соціалістичної Республіки', 'Творчість', 1949);*

1. Створити таблицю artists, що містить назви виконавців (музичних гуртів), країн їх походження, та рік заснування.  
   Додати до таблиці music нову колонку з ідентифікатором виконавця.  
   Заповнити відсутні значення шляхом пошуку відповідностей в назвах виконавців між таблицями.  
   Видалити назву виконавця з таблиці music.  
   Впевнитись, що неможливо додати посилання не неіснуючого виконавця.

*CREATE TABLE artists (*

*band\_name VARCHAR(50),*

*country VARCHAR(50),*

*found\_year INT,*

*);*

*ALTER TABLE artists*

*ADD artist\_id INT IDENTITY PRIMARY KEY;*

*ALTER TABLE music*

*DROP COLUMN singer\_name;*

*ALTER TABLE music*

*ADD artist\_id INT FOREIGN KEY REFERENCES artists(artist\_id);*

*INSERT INTO music (song, album, release\_year, artist\_id)*

*VALUES*

*('Stuk', 'Zvezda', 1989, 1),*

*('Radioactive', 'Night Visions', 2012, 2),*

*('Kapital', 'Kapital', 2007, 3),*

*('Veter', 'Eto vse', 1995, 4);*

*INSERT INTO artists (band\_name, country, found\_year)*

*VALUES*

*('Kino', 'USSR', 1981),*

*('Imagine Dragons', 'USA', 2008),*

*('Lyapis-98', 'Belarus', 1989),*

*('DDT', 'USSR', 1980);*

1. Створити представлення (view), що виводить всю інформацію про музичні треки виконавців зі Східної Європи.

*CREATE VIEW music\_eastern\_europe*

*AS*

*SELECT a.band\_name, a.country, a.found\_year, m.song, m.album, m.release\_year*

*FROM music m*

*JOIN artists a*

*ON m.artist\_id = a.artist\_id*

*WHERE country = 'USSR';*

1. Створити збережену процедуру, що у якості параметрів приймає назву треку та виконавця і заповнює відповідні таблиці. Не додавати записи, якщо вони вже є в базі.

CREATE PROCEDURE add\_track\_and\_singer (@song NVARCHAR(50), @band\_name NVARCHAR(50))

AS

BEGIN

IF NOT EXISTS (SELECT band\_name FROM artists

WHERE band\_name = @band\_name)

BEGIN

INSERT INTO artists (band\_name)

VALUES (@band\_name)

END

IF NOT EXISTS (SELECT song FROM music

WHERE song = @song)

BEGIN

INSERT INTO music (song, artist\_id)

VALUES (@song, (SELECT artist\_id FROM artists

WHERE band\_name = @band\_name))

END

END

1. Створити скалярну функцію, що приймає ідентифікатори пісні та виконавця та повертає назву треку з конкатенацією альбому та року за зразком:  
   Shine On You Crazy Diamond (Wish You Were Here, 1975)

CREATE FUNCTION full\_track\_name(@music\_id INT, @artist\_id INT)

RETURNS NVARCHAR(50)

AS

BEGIN

RETURN CONCAT(

(

SELECT m.song FROM music m

JOIN artists a ON a.artist\_id = m.artist\_id

WHERE m.music\_id = @music\_id AND a.artist\_id = @artist\_id

),

'(',

(

SELECT m.album FROM music m

JOIN artists a ON a.artist\_id = m.artist\_id

WHERE m.music\_id = @music\_id AND a.artist\_id = @artist\_id

),

', ',

(

SELECT m.release\_year FROM music m

JOIN artists a ON a.artist\_id = m.artist\_id

WHERE m.music\_id = @music\_id AND a.artist\_id = @artist\_id

),

')'

)

END;

1. Створити табличну функцію, що повертає ті ж колонки, що і створене представлення, проте фільтрація по країнам відбувається згідно вхідного параметру.

CREATE FUNCTION music\_by\_country (@country NVARCHAR(50))

RETURNS TABLE

AS

RETURN

SELECT m.\* from Music m

JOIN artists a on m.artist\_id = a.artist\_id

where a.country = @country

1. Створити тригер, що при додаванні пісні буде перевіряти, що рік її релізу не менше за рік початку діяльності виконавця.

CREATE OR ALTER TRIGGER check\_release\_year

ON music

INSTEAD OF INSERT

AS

BEGIN

IF (SELECT release\_year FROM inserted)

>= (SELECT found\_year FROM artists a

WHERE a.artist\_id = (SELECT artist\_id FROM inserted))

BEGIN

INSERT INTO music (song, album, release\_year, artist\_id)

SELECT song, album, release\_year, artist\_id FROM inserted

END

ELSE

BEGIN

SELECT 'Release year cannot be less than found year.'

END

END

*\*будь-ласка впевніться, що ви використовуєте Basic tier та видалили всі ресурси в кінці роботи*

**Матеріали для повторення:**

[How to change an Azure SQL Database Collation](https://www.sqlshack.com/how-to-change-an-azure-sql-database-collation/)

[CREATE TABLE (Transact-SQL) IDENTITY (Property)](https://docs.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql-identity-property?view=sql-server-ver15)

[Cloud Databases (practice) pt6](https://youtu.be/0QNTyaiUjYc) (INSERT, UPDATE, DELETE, TRUNCATE, CREATE TABLE)

[Cloud Databases (practice) pt7](https://youtu.be/3ymEzDdgAN0) (AZURE, PROCEDURE, UDF, SCALAR & TABLE FUNCTIONS)

[Cloud Databases (practice) pt8](https://youtu.be/OkqmeUfMVZg) (CURSOR, TRIGGER, QUERY OPTIMIZATION)

[Оператори INSERT, UPDATE, DELETE](http://fcit.tneu.edu.ua/navchannja/pidhotovka-do-pratsevlashtuvannia/56-web-rozrobka/mysql/579-mysql-insert)