Azure Setup Guide

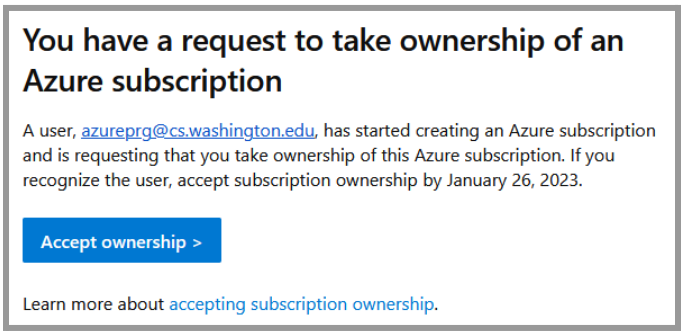
Introduction

In the following assignments, you will learn how to set up and use a cloud-hosted database. Thus, your first step will be to set up a SQLServer database in the Azure cloud service and import the flight data. This step is tedious but important: we want you to be able to continue using Azure after the class ends! So, you need to know how to set up the entire system starting from nothing.

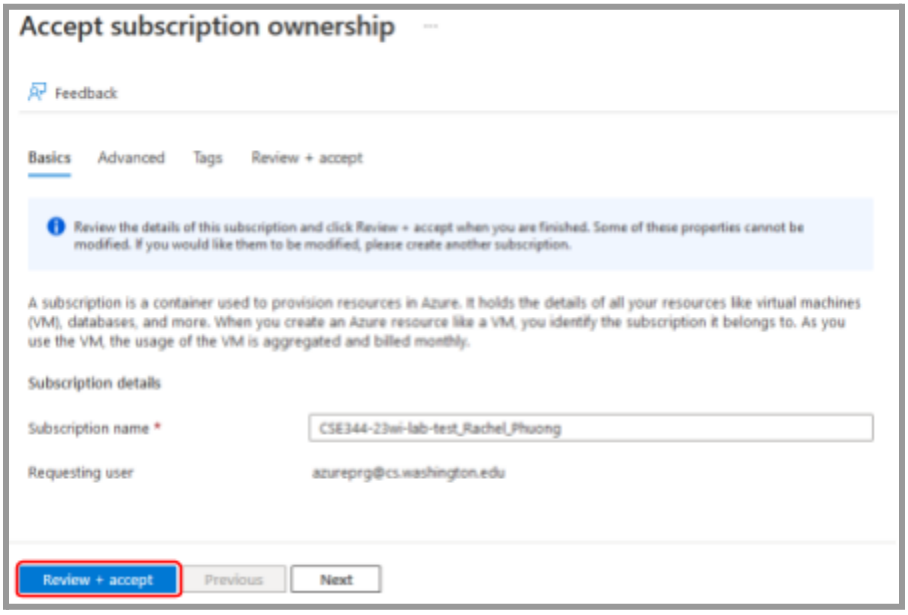
NOTE: These steps take several hours to complete, so start early!

Instructions

Step 1: Create an Azure account and login to Azure portal



A. You should have an email with the subject "You've been asked to accept Azure subscription ownership". Make sure you are logged into that account, **and only that account**, when you click on the link.
    a. Eg, if you have an @cs and an @uw account and the mail was sent to your @uw, then you should ONLY be signed into your @uw account.
B. Click "Accept ownership" to be forwarded to the Azure portal.
C. Click "Review + accept"



D. Click "Accept"

The above screenshot shows an example of a subscription; your information will be different. If you have any questions, please contact the course staff.

Step 2: Learn about Azure and SQLServer

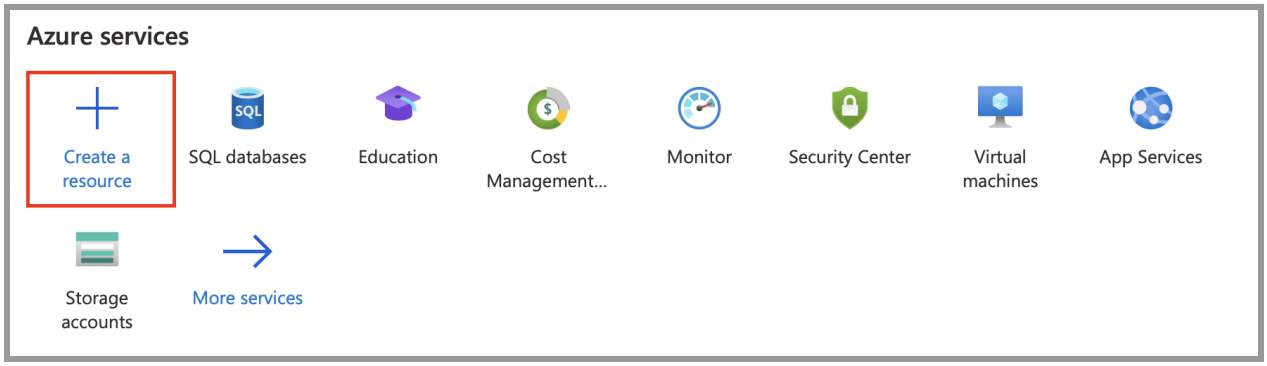
Spend some time clicking around, reading documentation, watching tutorials, and generally familiarizing yourself with Azure (the cloud service) and SQLServer (the RDMS).

- SQL Server Tutorial For Beginners
- SQL Server Technical Documentation
- SQL Server Tutorial
- Transact-SQL Reference

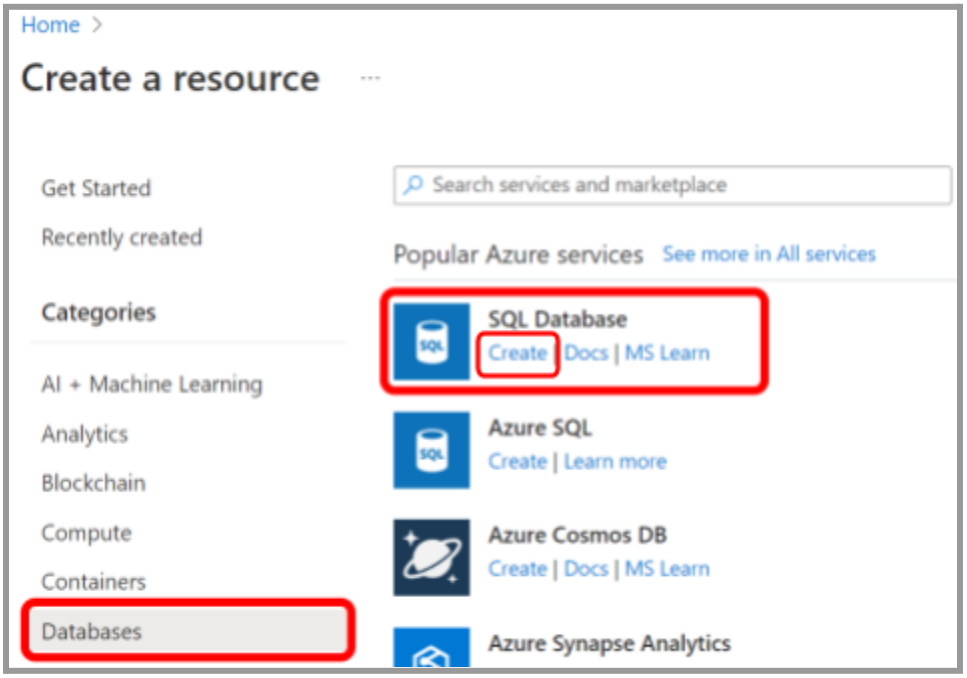
Step 3: Create the database

A. Navigate to the Azure portal.

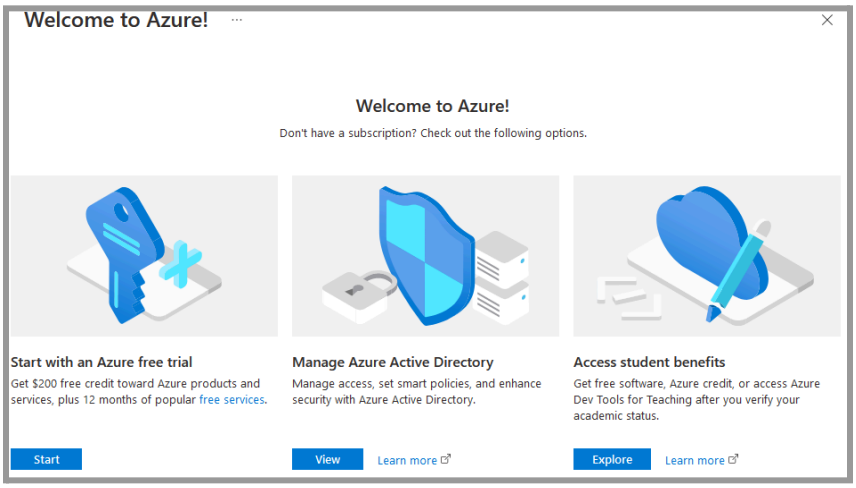
B. Under “Azure services”, select “Create a resource”



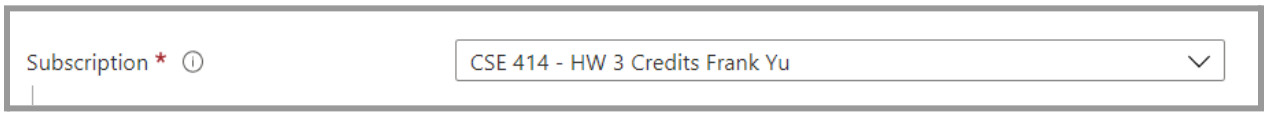
C. Select “Databases” under Categories, and then select “Create” for “SQL Database”.



D. If you see the “Welcome to Azure” screen, your subscription is not currently active. Please ensure that you have completed Step 1 and contact the course staff if you have questions.



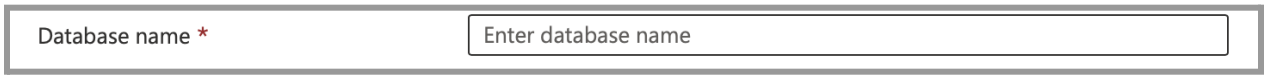
E. Make sure that your “Subscription” is **NOT** set to “Free Trial” or “Azure for Students”. The correct subscription name will either:
   a. mention “Homework 3”
   b. or mention “Microsoft Azure Sponsorship”
   c. or mention CSE 344 or CSE 414



The above screenshot shows an example of a subscription; your information will be different. If you have any questions, please contact the course staff.

F. Under Subscription, create a new resource group with a name of your choice.

G. Choose a database name.



H. Create a new server.

a. Choose a server name

*****Do not use the @ symbol in your server name!*****

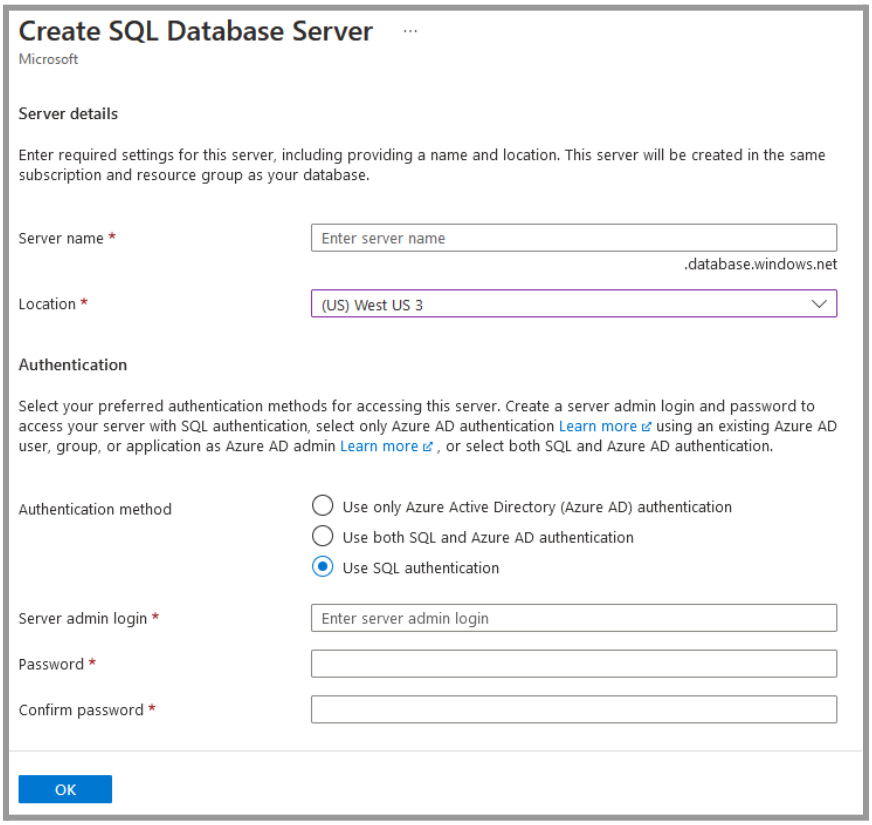
This naming requirement is strictly necessary for future homeworks. If you do not follow this advice, you'll have to reset your login information later.

b. Choose the server location that's closest to you
   i. For example, if you're located in Seattle and one of the "US West" locations is available, choose that location
   ii. If none of the US West locations are available, you can pick another location

c. Set the authentication method to **"Use SQL authentication"**.

d. Select an admin login and password; ***don't lose these values***! Once again,

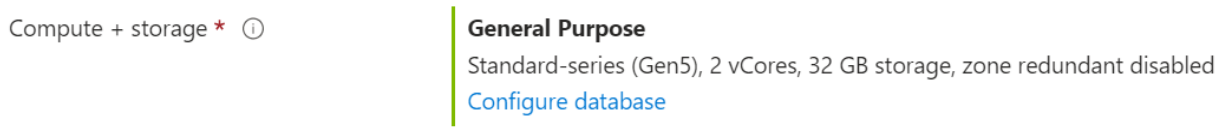
*****Do not use the @ symbol in your admin login or password!*****



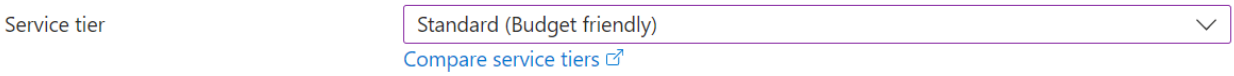
I. Ensure that "Want to use SQL elastic pool?" is set to **NO.**

J. Under Workload Environment, select Production

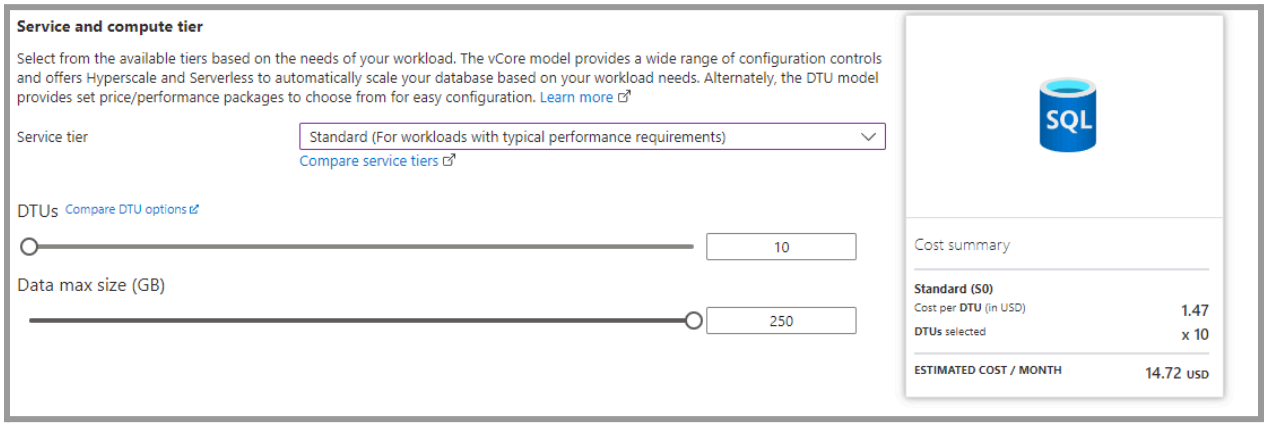
K. **VERY IMPORTANT:** Under “Compute + storage”, do **NOT** use the default “General Purpose” as this costs about $400 per month and you will burn through your credits before you finish the assignment. (We want Standard S0: 10 DTUs, and 250 GB storage.)



Click on “Configure database“ and in the next screen, set the Service Tier to “Standard”.



The estimated cost should be around $15 per month. If you don’t see this price, go back to the previous step and make sure you have configured the service tier correctly.

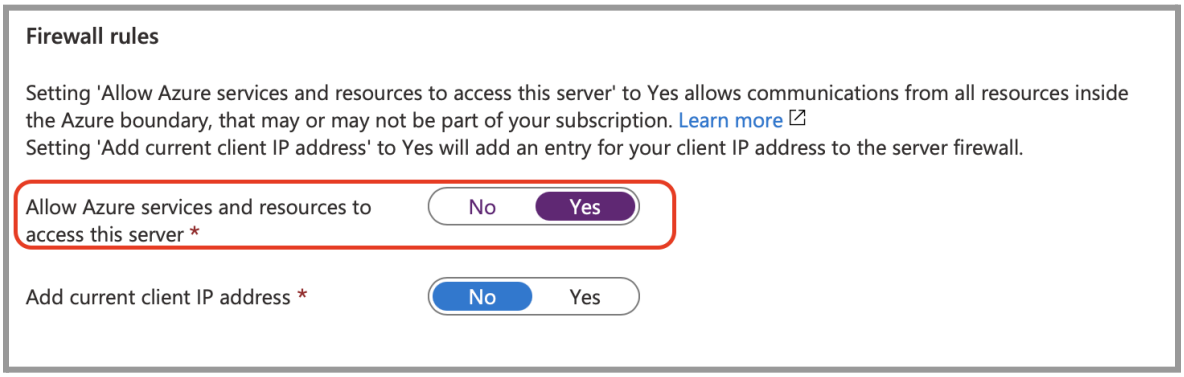


L. Click on "Apply" at the bottom of the page.

M. Click on “Next: Networking” at the bottom of the page.

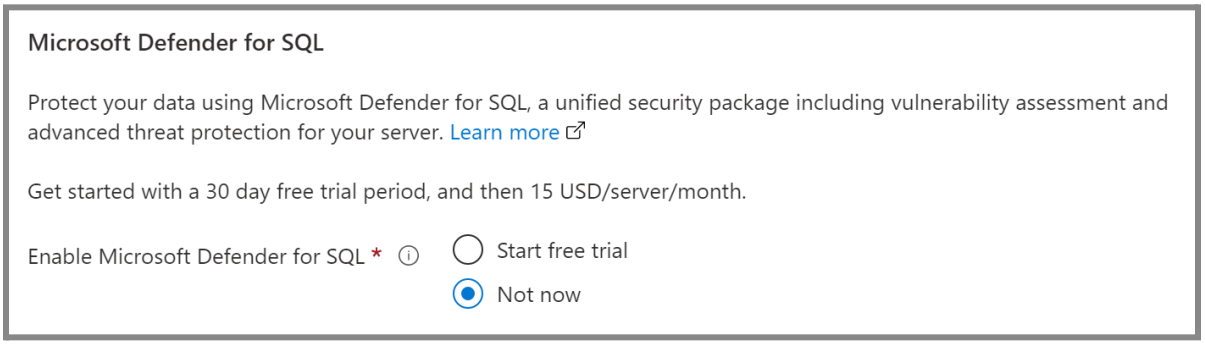
N. For “Connectivity method”, select “Public endpoint”.

O. Under “Firewall rules”, ensure that “Allow Azure services and resources to access this server” is set to “Yes”.



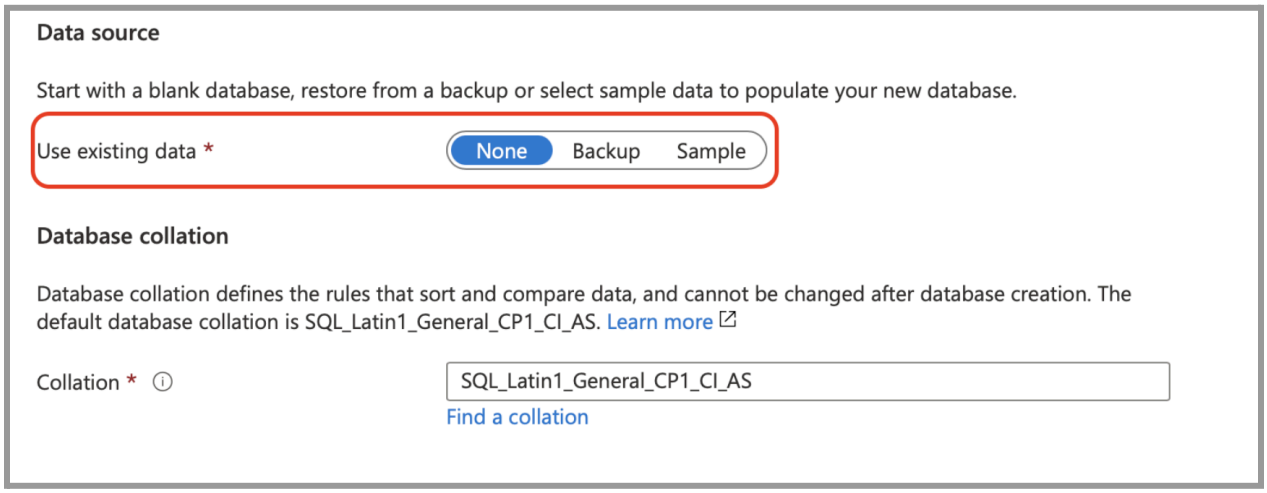
P. Click on “Next: Security” at the bottom of the page.

Q. Ensure “Microsoft Defender for SQL” is set to “Not now”.



R. Click on “Next: Additional settings” at the bottom of the page.

S. Under “Data source”, ensure that “Use existing data” is set to “None”.



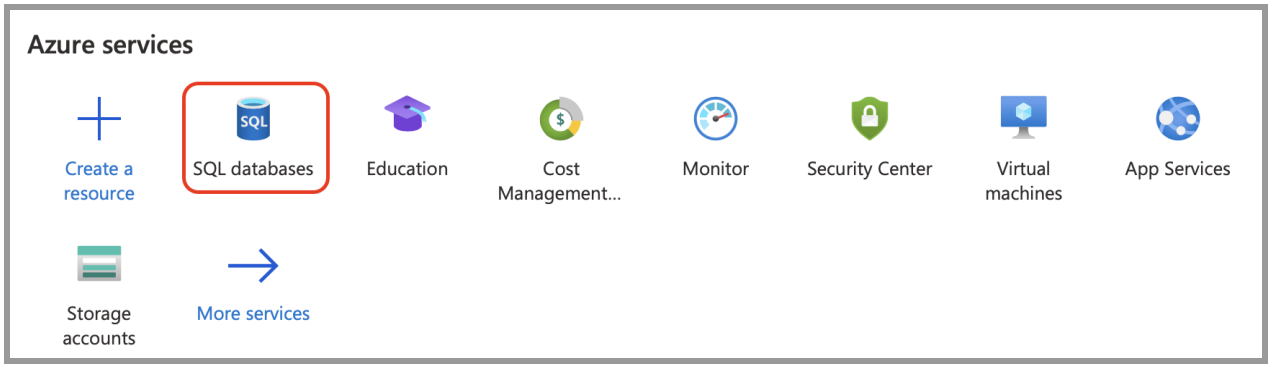
T. Click on “Review + create” at the bottom of the page.

U. Click on “Create” at the bottom of the page.

V. Wait a few minutes! If you encounter any issues, please contact the course staff!

Step 4: Set the Server Firewall

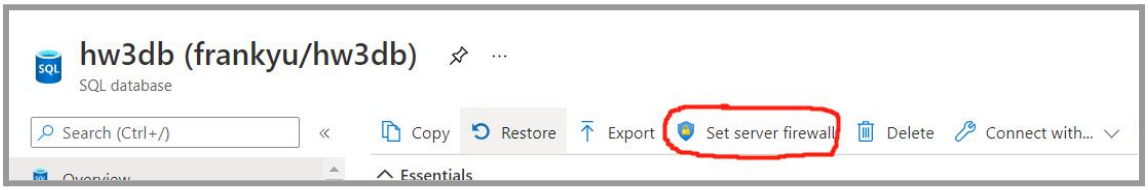
A. Once the deployment is complete, navigate to Azure portal. Under “Azure services”, click on “SQL database”.



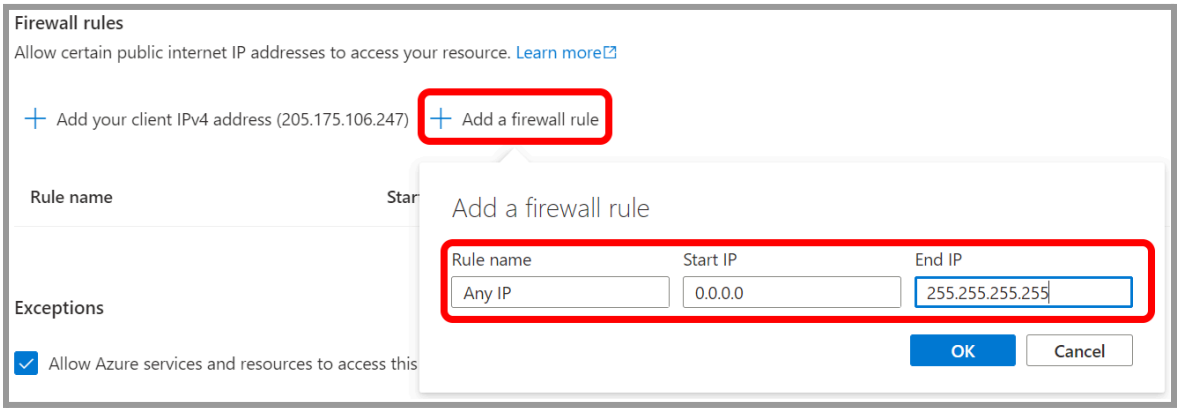
B. Click on the database that you just created.

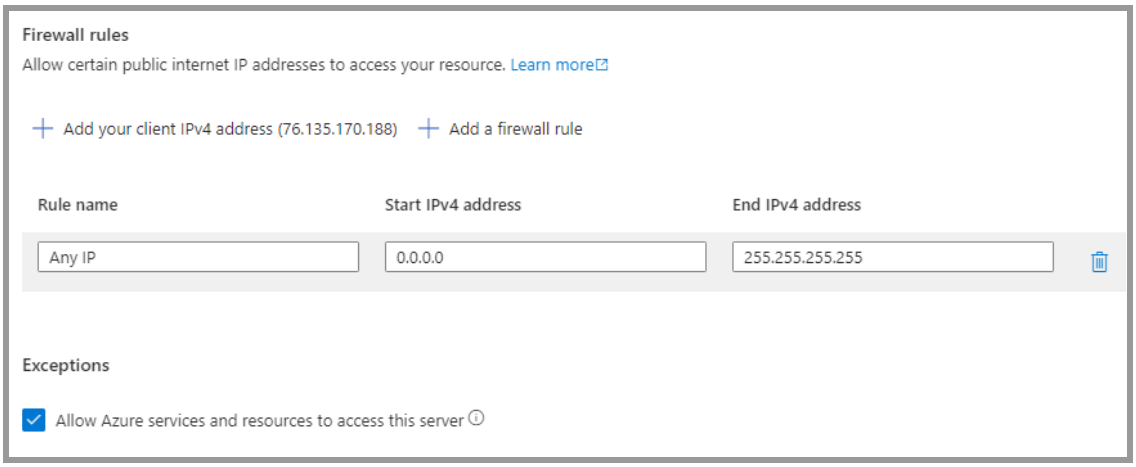
The above screenshot shows an example of a database; yours will be named differently. If you have any questions, please contact the course staff.

C. At the top of the page, click on “Set server firewall”.



D. Add a new firewall rule that allows connections from any client. For “Start IP”, type in “0.0.0.0”, for “End IP”, type in “255.255.255.255”. Remember to save your settings!





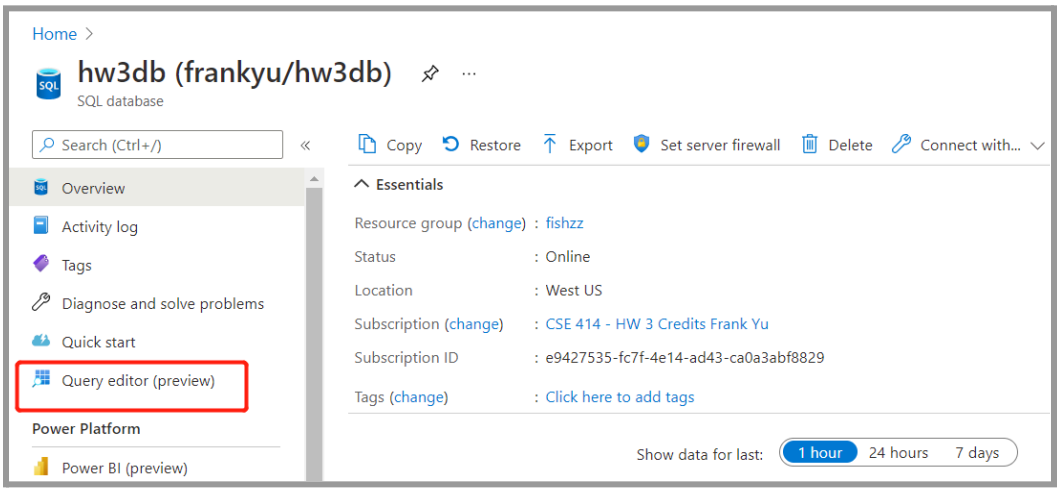
E. You have successfully set up your first SQLServer database on Azure!

Step 5: Try out the built-in Query editor

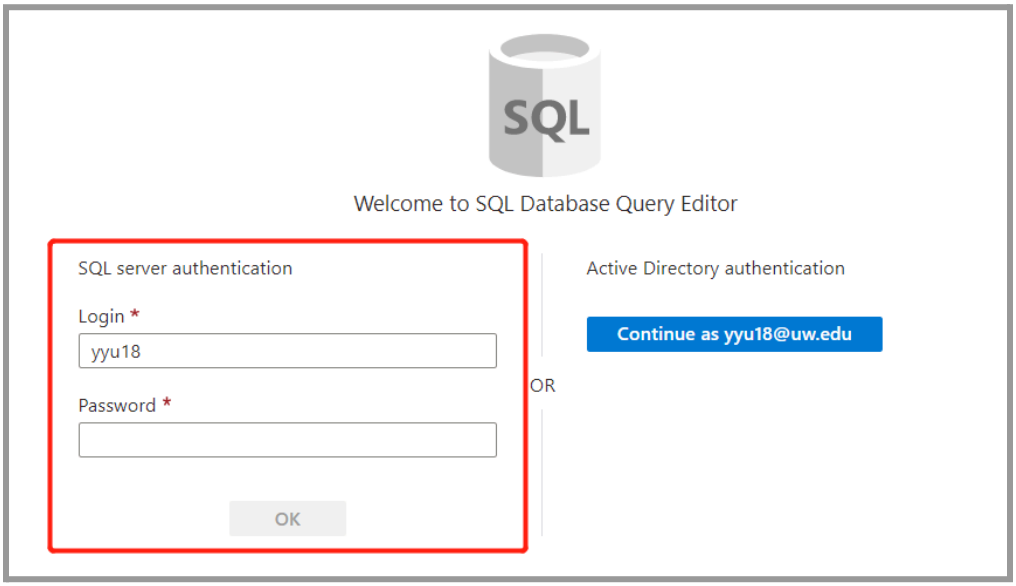
The simplest way to play with the database is using the built-in Query editor. To launch this:

A. Navigate to the SQL database you just created.

B. Click on “Query editor (preview).



C. Log in using the admin login and password you created above (Step 3H).



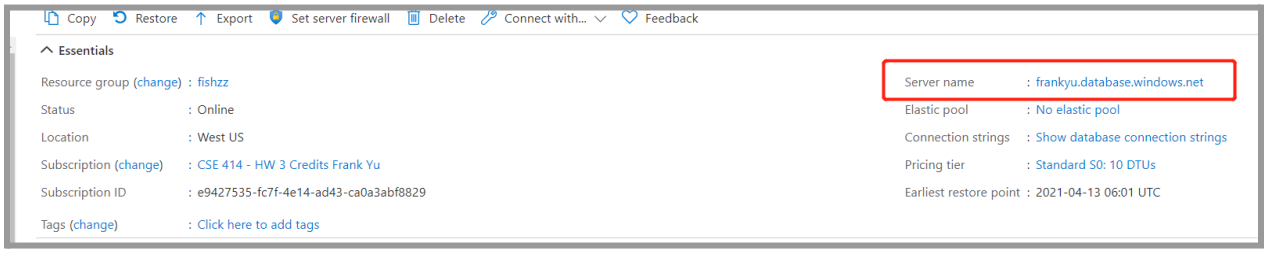
The above screenshot shows an example of a login; your information will be different. If you have any questions, please contact the course staff.

Step 5.1: Use VSCode! (Optional)

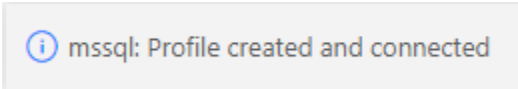
Besides using the built-in query editor, you can also connect VSCode to your database. Visual Studio Code, also known as VSCode, is a free source code editor made by Microsoft. It runs on Windows, Linux, and macOS.

Read the instructions here to set up your VSCode! More specifically, follow the steps in:

- "Install the mssql extension in VSCode"
- "Create or open a SQL file"
- *(most importantly)* "Connect to SQL Server"
    - Your server name can be found in the dashboard for your SQL database.



    - Ensure you do not skip the "Database name". Enter the database you created.
    - For Authentication Type, select "SQL Login"
    - After creating the profile, you should see



- Press Ctrl+Shift+E to execute SQL statements in the active SQL file (or click the green arrow in the top right corner)
- *Note that if you are logged into attu you will not be able to connect to your Azure database*. Close the remote connection (File -> Close Remote Connection) and then try connecting to again.

Step 6: Ingest the data!

A. Declare where the flights data can be found by declaring an external data source. In your SQLServer query editor, run the following:

```
CREATE EXTERNAL DATA SOURCE flightdata_blob
WITH (TYPE = BLOB_STORAGE,
LOCATION = 'https://introdatamanagement.blob.core.windows.net/flightdata'
);
```

B. Execute your create table statements from hw2
   a. Don't copy the SQLite-specific statements, such as `PRAGMA foreign_keys=ON` or '.mode' statements.
   b. **If your tables haven't been created before proceeding with the following steps, you will have a bad time**

C. Import the flight data into your newly-created tables and set up indexes on them.

```
bulk insert Carriers from 'carriers.csv'
with (ROWTERMINATOR = '0x0a',
DATA_SOURCE = 'flightdata_blob', FORMAT='CSV', CODEPAGE = 65001, --UTF-8
encoding
FIRSTROW=1,TABLOCK);

bulk insert Months from 'months.csv'
with (ROWTERMINATOR = '0x0a',
DATA_SOURCE = 'flightdata_blob', FORMAT='CSV', CODEPAGE = 65001, --UTF-8
encoding
FIRSTROW=1,TABLOCK);

bulk insert Weekdays from 'weekdays.csv'
with (ROWTERMINATOR = '0x0a',
DATA_SOURCE = 'flightdata_blob', FORMAT='CSV', CODEPAGE = 65001, --UTF-8
encoding
FIRSTROW=1,TABLOCK);

-- Import for the large Flights table.
-- This last import might take a little under 10 minutes on the
-- provided server settings
bulk insert Flights from 'flights-small.csv'
with (ROWTERMINATOR = '0x0a',
DATA_SOURCE = 'flightdata_blob', FORMAT='CSV', CODEPAGE = 65001, --UTF-8
encoding
FIRSTROW=1,TABLOCK);
```

```
-- After you run the code above successfully, you can move on to creating the
indexes.

-- Indexes, which we'll discuss later this quarter, will make your
-- homework queries run much faster (optional, but STRONGLY recommended).
-- In aggregate, these three statements will take about 10 minutes
create index Flights_idx1 on Flights(origin_city,dest_city,actual_time);
create index Flights_idx2 on Flights(actual_time);
create index Flights_idx3 on Flights(dest_city,origin_city,actual_time);
```

D. In order to verify that your imports were successful, do some SELECT COUNT(*) statements. If everything is set up correctly, you should have:
    a. 1594 rows for Carriers.
    b. 12 rows for Months.
    c. 8 rows for Weekdays.
    d. 1148675 rows for Flights.

E. Congratulations! You're done with the setup and can proceed back to the homework.