

# Module 3 : Database

PURANJIT SINGH

03/29/2022

# Introduction

**Database** - any collection of related information. E.g. – amazon user database, banking services, auto industry etc.

**Database Management Systems (DBMS)** – a special software program that helps user create, maintain and secure a database.

- Makes it easy to manage large amounts of information
- Handles security
- Backups
- Importing/exporting data
- Concurrency
- Interacts with other software applications

# C. R. U. D. operations

There are generally four main operations that a database management systems perform, and they are:

- **Create** – Creation of a database
- **Read** – Reading of a database
- **Update** – Updating a database
- **Delete** – Deleting a database

**In a database** - Data is organized into one or more tables.

- Each table has columns and rows
- A unique key identifies each row
- Structured Query Language (SQL) is used for interacting with these database management systems

# Intuition & Imp. terms

PK		FK	
NUID	First name	Last name	Department Code
94849311	John	Wick	BSE
64895674	Bruce	Wayne	CSE
32158463	Peter	Parker	ECE

Student table

PK			
Location	Pantry Location	Opening time	Closing time
East	Dairy store	16:00:00	17:00:00
City	Health center	12:00:00	16:00:00

Pantry information table

PK	Primary Key
FK	Foreign Key
	Attributes

PK		FK	
Department Code	Department name	Location	
BSE	Biological Systems Engg.	East	
CSE	Computer Science Engg.	City	
ECE	Electrical & Computer Engg.	City	
STAT	Statistics	East	

Department table

# Problem description — Field Scouting Logs

Assume you are hired to maintain field scouting logs in a company, following tasks could be done -

- You need to maintain the record for people who went for data collection on different dates
- Maintain logs for which vehicles were taken on specific days
- Maintain record of drones with the company
- Personal details of employees
- ...



<https://www.potatopro.com/products/agroscout-crop-scouting-package-drone-software>





# Original dataset :

A	B	C	D	E	F	G	H	I	J	K	L	M
Flight No	Date	NU ID	Drone_id	Truck_id	First_name	Last_name	Phone_number	Drone_name	Date_purchase	Truck_name	Truck_plate	
1	5/1/2022	100	1	1	John	Wick	3526520474	DJI Matric 300 RTK	8/14/2021	Ford 150	UAV129	
2	5/8/2022	110	2	2	Peter	Parker	4319997713	DJI Phantom 4	4/5/2019	Ford 250	OIG896	
3	5/15/2022	120	2	3	Dwayne	Johnson	4024053487	DJI Phantom 4	4/5/2019	Ford 350	PII200	
4	5/22/2022	130	2	1	James	Bond	8872131333	DJI Phantom 4	4/5/2019	Ford 150	UAV129	
5	5/29/2022	100	1	2	John	Wick	3526520474	DJI Matric 300 RTK	8/14/2021	Ford 250	OIG896	
6	6/5/2022	130	2	3	James	Bond	8872131333	DJI Phantom 4	4/5/2019	Ford 350	PII200	
7	6/12/2022	120	1	1	Dwayne	Johnson	4024053487	DJI Matric 300 RTK	8/14/2021	Ford 150	UAV129	
8	6/19/2022	120	2	2	Dwayne	Johnson	4024053487	DJI Phantom 4	4/5/2019	Ford 250	OIG896	
9	6/26/2022	100	2	1	John	Wick	3526520474	DJI Phantom 4	4/5/2019	Ford 150	UAV129	
10	7/3/2022	110	2	3	Peter	Parker	4319997713	DJI Phantom 4	4/5/2019	Ford 350	PII200	
11	7/10/2022	100	1	1	John	Wick	3526520474	DJI Matric 300 RTK	8/14/2021	Ford 150	UAV129	
12	7/17/2022	110	1	2	Peter	Parker	4319997713	DJI Matric 300 RTK	8/14/2021	Ford 250	OIG896	
13	7/24/2022	120	2	3	Dwayne	Johnson	4024053487	DJI Phantom 4	4/5/2019	Ford 350	PII200	
14	7/31/2022	130	1	2	James	Bond	8872131333	DJI Matric 300 RTK	8/14/2021	Ford 250	OIG896	
15	8/7/2022	100	2	2	John	Wick	3526520474	DJI Phantom 4	4/5/2019	Ford 250	OIG896	
16	8/14/2022	130	2	1	James	Bond	8872131333	DJI Phantom 4	4/5/2019	Ford 150	UAV129	
17	8/21/2022	120	2	2	Dwayne	Johnson	4024053487	DJI Phantom 4	4/5/2019	Ford 250	OIG896	
18	8/28/2022	120	2	3	Dwayne	Johnson	4024053487	DJI Phantom 4	4/5/2019	Ford 350	PII200	
19	9/4/2022	100	1	1	John	Wick	3526520474	DJI Matric 300 RTK	8/14/2021	Ford 150	UAV129	
20	9/11/2022	110	2	1	Peter	Parker	4319997713	DJI Phantom 4	4/5/2019	Ford 150	UAV129	

# Data converted to a relational Database

Flights_record				
Flight_No	Date	NUID	Drone_id	Truck_id
1	5/1/2022	100	1	1
2	5/8/2022	110	2	2
3	5/15/2022	120	2	3
4	5/22/2022	130	2	1
5	5/29/2022	100	1	2
6	6/5/2022	130	2	3
7	6/12/2022	120	1	1
8	6/19/2022	120	2	2
9	6/26/2022	100	2	1
10	7/3/2022	110	2	3
11	7/10/2022	100	1	1
12	7/17/2022	110	1	2
13	7/24/2022	120	2	3
14	7/31/2022	130	1	2
15	8/7/2022	100	2	2
16	8/14/2022	130	2	1
17	8/21/2022	120	2	2
18	8/28/2022	120	2	3
19	9/4/2022	100	1	1
20	9/11/2022	110	2	1

NAMES TABLE			
NUID	First_name	Last_name	Phone_number
100	John	Wick	3526520474
110	Peter	Parker	4024053487
120	Dwayne	Johnson	4319997713
130	James	Bond	8872131333

DRONE TABLE		
Drone_id	Drone_name	Date purchased
1	DJI Phantom 4	1-May
2	Matrice 300 R	8-Aug

TRUCK TABLE		
Truck_id	Truck_name	Truck_plate
1	Ford150	UAV123
2	RAM 200	GHI877
3	Chevorlet	POI900

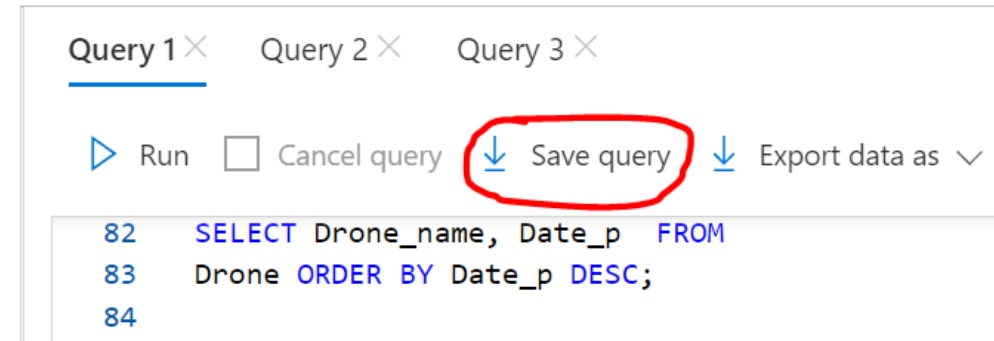
# Today's lab

At the end of today's lab, you will be able to:

1. Create a mini database with this dataset on Microsoft Azure;
2. Be able to run a few basic queries on the database with the SQL code provided.

## Database Assignment 1, due 5:00 pm, April 11:

1. Download and submit the mini database you create on Azure with the flight dataset;
2. Turn in your .sql file with the five queries assigned on page 8 of this document)



.sql file will be downloaded when you click on this button on Azure SQL editor



# Database Assignment *(Due – April 11)*

Write SQL queries to create a database then extract the following from the defined tables in your query editor –

1. Insert the name of a new person into the *Names* table besides the names already provided in the slides and return/display all the names on the table. *(20 points)*
2. Return/display only the NUID and Drone\_id in the *Flights\_record* table. *(20 points)*
3. Return/display when ‘DJI Phantom 4’ was purchased. *(25 points)*
4. Update the name of ‘DJI Matrice 300 RTK’ to ‘DJI 300’ in the *Drone\_table*. *(25 points)*
5. Bonus: Return/display NUID, Drone\_id, Truck\_id for all the dates in July month *(30 bonus points)*
6. Comments: include extensive comments for your code to explain your thinking process. *(10 points)*

## **Due Date:**

5 PM, Monday, April 11th in Canvas

# Microsoft Azure Services

Azure is a cloud computing platform and an online portal that allows you to access and manage cloud services and resources provided by Microsoft.

...

**Azure SQL Database**

...



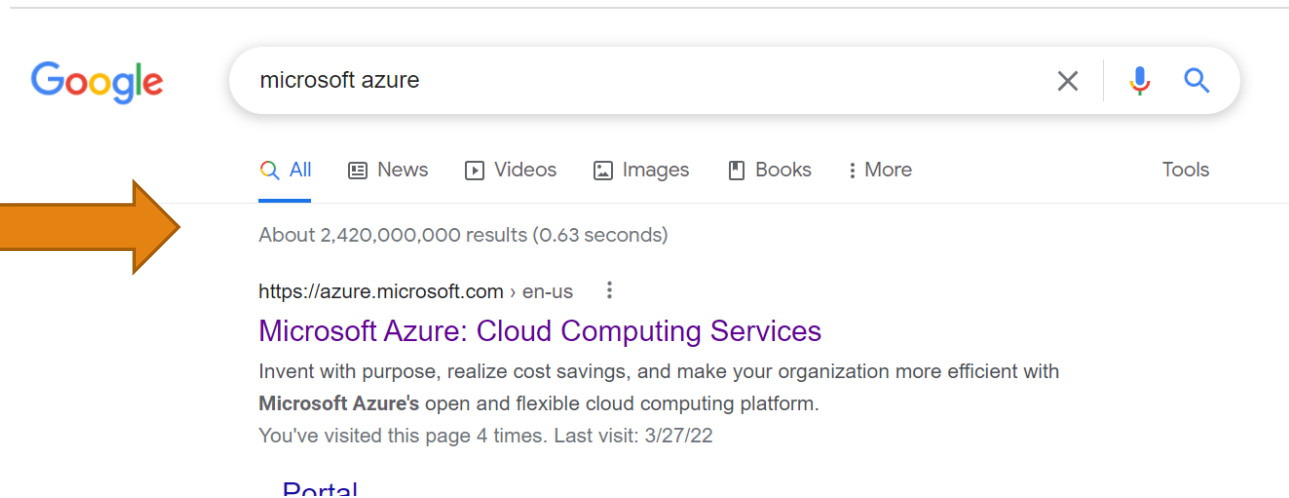
# Open browser and follow the steps

1. Search Microsoft azure



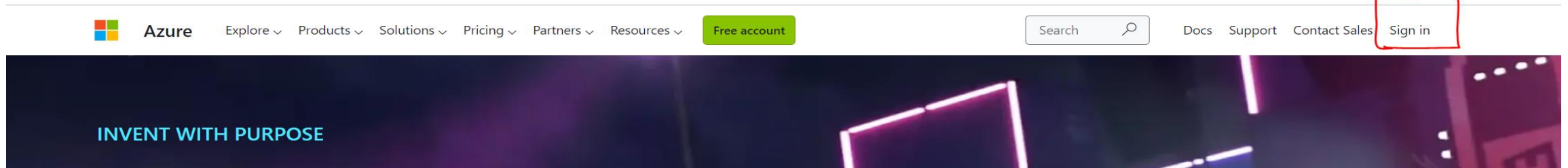
2. You can do the following google search or use this link to open Microsoft Azure in your browser

Link - [https://azure.microsoft.com/en-us/?exp=175071&adobe\\_mc\\_sdid=SDID%3D3AF4519F95387A4C-0B212557B10650A9%7CMCORGID%3DEA76ADE95776D2EC7F000101%40AdobeOrg%7CTS%3D1648423877&adobe\\_mc\\_ref=https%3A%2F%2Fwww.google.com%2F](https://azure.microsoft.com/en-us/?exp=175071&adobe_mc_sdid=SDID%3D3AF4519F95387A4C-0B212557B10650A9%7CMCORGID%3DEA76ADE95776D2EC7F000101%40AdobeOrg%7CTS%3D1648423877&adobe_mc_ref=https%3A%2F%2Fwww.google.com%2F)

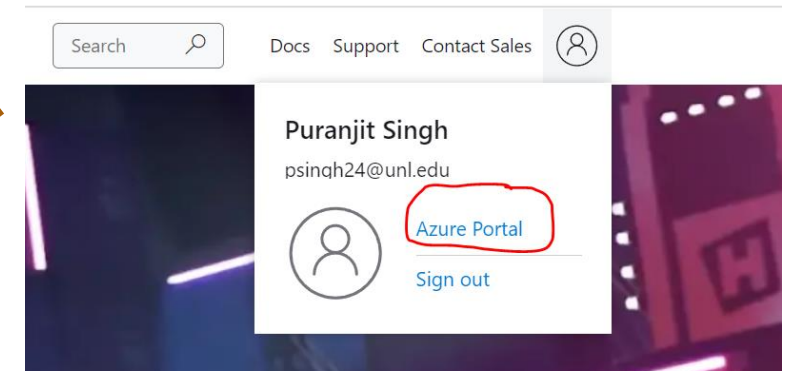


# Continued:

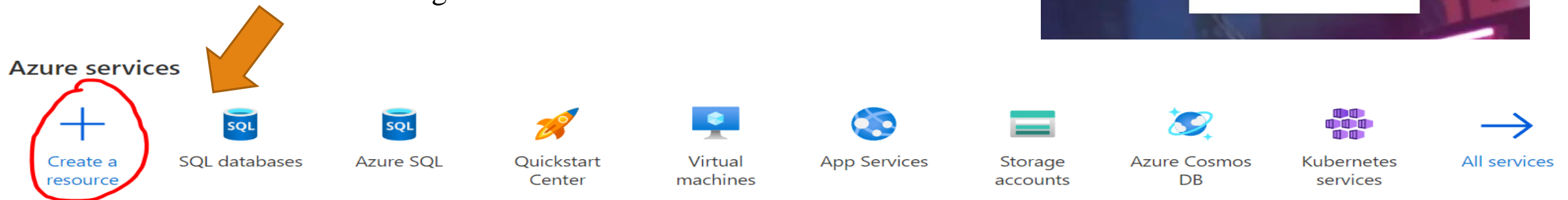
3. Sign in with your UNL user id into the Microsoft azure platform (it could ask to enter the details using Duo push).



4. After logging in : Select Azure portal



5. Azure services page would open up :  
◦ Select the following – Create a resource



# Continued:

## 6. Select SQL Database under Popular Azure services section

## 7. Section - Project details

- Subscription : Azure for students (by default)
- Resource group : Click on Create New
  - Enter name of your choice (E.g. – MYSM492892)
  - Click Ok

A resource group is a container that holds related resources for an Azure solution.

Name \*

MYSM492892 ✓

OK Cancel

## 8. Section – Database details

- Select : Create new
- Make sure – the values under these sections are not clear

### Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name \*

Enter database name

Server \* ⓘ

Select a server

Create new

Popular Azure services [See more in All services](#)



Virtual machine  
[Create](#) | [Learn more](#)



Kubernetes Service  
[Create](#) | [Docs](#) | [MS Learn](#)



Azure Cosmos DB  
[Create](#) | [Docs](#) | [MS Learn](#)



Function App  
[Create](#) | [Docs](#)



SQL Database  
[Create](#) | [Docs](#) | [MS Learn](#)



Storage account  
[Create](#) | [Docs](#) | [MS Learn](#)



DevOps Starter  
[Create](#) | [Docs](#) | [MS Learn](#)



Web App  
[Create](#) | [Docs](#) | [MS Learn](#)

# Continued:

## 9. A pop-up page –

- Title : Create SQL Database Server will open-up
- Put in the details as shown in the image
- Create password on your own and remember it for further use
- Click OK at the bottom of the page



[Home](#) > [Create a resource](#) > [Create SQL Database](#) >

## Create SQL Database Server

Microsoft

### Server details

Enter required settings for this server, including providing a name and location. This server will be created in the same subscription and resource group as your database.

Server name \*  ✓  
Location \*  ✓

### Authentication

Select your preferred authentication methods for accessing this server. Create a server admin login and password to access your server with SQL authentication, select only Azure AD authentication [Learn more](#) using an existing Azure AD user, group, or application as Azure AD admin [Learn more](#), or select both SQL and Azure AD authentication.

Authentication method  
☒ Use SQL authentication  
☐ Use only Azure Active Directory (Azure AD) authentication  
☐ Use both SQL and Azure AD authentication

Server admin login \*  ✓

Password \*  ✓

Confirm password \*  ✓

## 10. Click on configure database

- A pop-up page will open



Want to use SQL elastic pool? \* ⓘ

☐ Yes ☒ No

Compute + storage \* ⓘ

### General Purpose

Gen5, 2 vCores, 32 GB storage

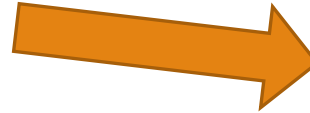
[Configure database](#)



# Continued :

11. Select the following option from the available ones

- Basic (for less demanding workloads)



12. Click on : Apply

- (Bottom-left of the page)

13. Select : Review + create

- Deployment is in progress (window will pop-up)



**Review + create**

Next : Networking >

14. After completion : Go to resource

[Home](#) > [Create a resource](#) > [Create SQL Database](#) >

**Configure** ...

 Feedback

## Service and compute tier

Select from the available tiers based on the needs of your workload. The vCore model provides a wide range of configuration controls and offers Hyperscale and Serverless to automatically scale your database based on your workload needs. Alternately, the DTU model provides set price/performance packages to choose from for easy configuration. [Learn more](#)

Service tier

Basic (For less demanding workloads) ▾

DTUs [What is a DTU?](#)

**5 (Basic)**

Data max size (GB)

### vCore-based purchasing model

General Purpose (Scalable compute and storage options)

Hyperscale (On-demand scalable storage)

Business Critical (High transaction rate and high resiliency)

### DTU-based purchasing model

Basic (For less demanding workloads)

Standard (For workloads with typical performance requirements)

Premium (For IO-intensive workloads)

# Continued :

## 15. Select : Query editor (preview)

demo-mysm (demo-mysm/demo-mysm) SQL database

Search (Ctrl+/) << Copy Restore Export Set server firewall Delete Connect with... Feedback

Overview

- Activity log
- Tags
- Diagnose and solve problems
- Quick start
- Query editor (preview)**

Power Platform

- Power BI
- Power Apps
- Power Automate

Settings

- Compute + storage
- Connection strings
- Properties
- Locks

Data management

- Replicas
- Sync to other databases

Essentials

Resource group (move) : [MYSM492892](#)

Status : Online

Location : Central US

Subscription (move) : [Azure for Students](#)

Subscription ID : 0de50d01-2e4f-443b-80cd-234f2d750260

Tags (edit) : [Click here to add tags](#)

Show data for last: 1 hour 24 hours 7 days

Compute utilization

DTU percentage (Max) demo-mysm/demo-mysm 3%

## 16. SQL server authentication windows will open –

- Enter your registered username and password here that you have set
- Select : Allowlist IP \_\_\_\_\_ on server demo-mysm to continue

Welcome to SQL Database Query Editor

SQL server authentication

Login \* root-mysm

Password \* .....

Cannot open server 'demo-mysm' requested by the login. Client with IP address '129.93.161.221' is not allowed to access the server. To enable access, use the Azure Portal or run sp\_set\_firewall\_rule on the master database to create a firewall rule for this IP address or address range. It may take up to five minutes for this change to take effect.

[Allowlist IP 129.93.161.221 on server demo-mysm](#)

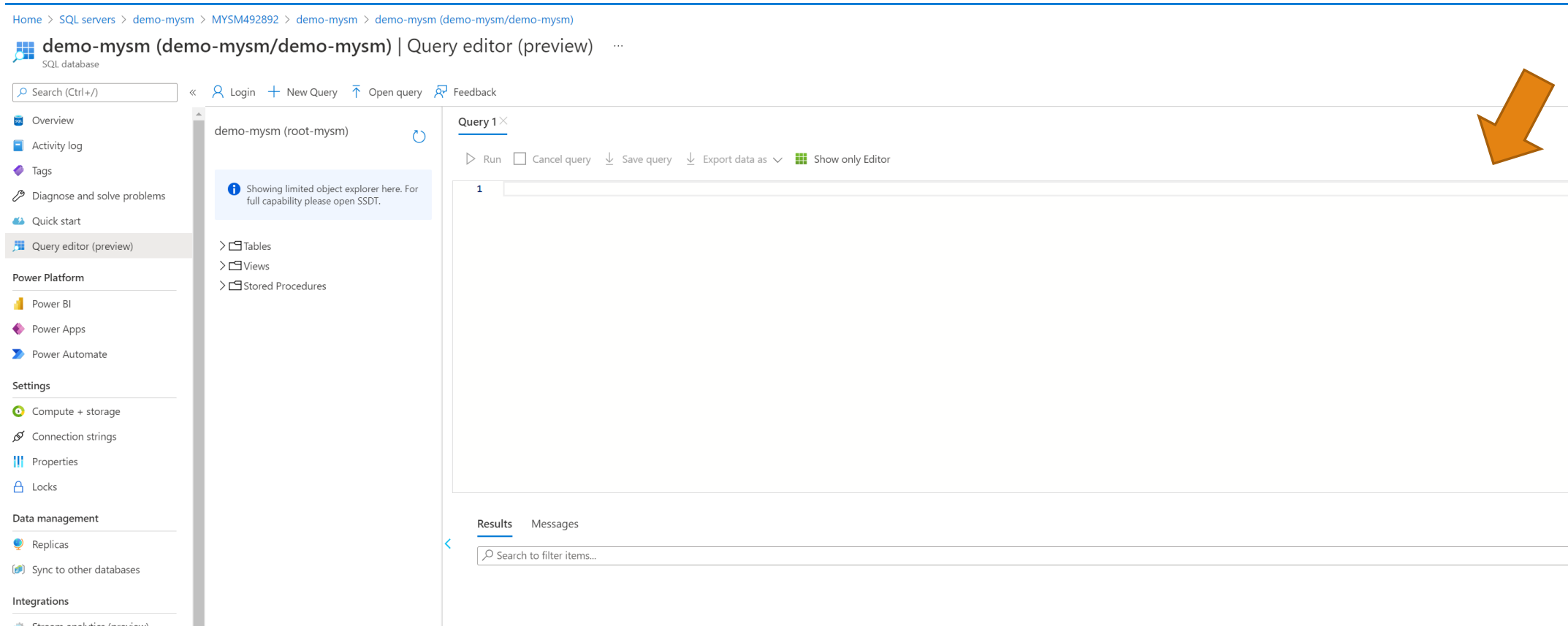
OK

Active Directory authentication

Continue as psingh24@unl.edu

# Continued :

17. SQL editor will open-up where you can run SQL queries to create your first database.



The screenshot displays the Azure SQL Editor interface. The top navigation bar shows the path: Home > SQL servers > demo-mysm > MYSM492892 > demo-mysm > demo-mysm (demo-mysm/demo-mysm). Below this, the title bar reads "demo-mysm (demo-mysm/demo-mysm) | Query editor (preview)". The interface is divided into three main sections: a left sidebar, a central object explorer, and a right query editor.

The left sidebar contains a search bar and a list of navigation items: Overview, Activity log, Tags, Diagnose and solve problems, Quick start, Query editor (preview) (highlighted), Power Platform (Power BI, Power Apps, Power Automate), Settings (Compute + storage, Connection strings, Properties, Locks), Data management (Replicas, Sync to other databases), and Integrations (Stream analytics preview).

The central object explorer shows the "demo-mysm (root-mysm)" database with a message: "Showing limited object explorer here. For full capability please open SSDT." Below this, there are expandable sections for Tables, Views, and Stored Procedures.

The right query editor is titled "Query 1" and includes a toolbar with buttons for Run, Cancel query, Save query, Export data as, and Show only Editor. The query text area is currently empty, with a line number "1" visible. Below the query editor, there are tabs for Results and Messages, and a search bar labeled "Search to filter items...".

A large orange arrow points to the "Query editor (preview)" tab in the left sidebar.

# Guidelines for completion of the assignment:

1. Firstly, we will define the schema for all the tables that are to be included in the database.
2. Than we will alter the Flights record table and add the foreign keys to it (in order to relate the tables with others) – **Imp.** -*This step is to be done before insertion of any data into the table*
3. Insert the values into the table in a sequence – Names table, Drone table, Truck table, Flight record table.
4. Once all the data is inserted – Run queries to fetch the information asked in the class assignment

# Hands on session

## 1. -- Creating tables in the DB

- CREATE TABLE *Table\_name*(*Defining attributes*);

## 2. -- Defining the foreign keys

- ALTER TABLE *Table\_name* ADD FOREIGN KEY(*Attribute*) REFERENCES *Table\_name*(*Attribute*)

## 3. – Insert values into the table

- INSERT INTO *Table\_name* VALUES(*insert values in correct order*)

## 4. Drop/Deleting a table from DB

- DROP TABLE *Table\_name*;

## 5. Running queries

- SELECT keyword, functions etc.
- Functions in SQL

## 6. Assignment related doubts

# Running queries

- **SELECT** keyword –
  - `SELECT * FROM (Table_name);` --**Display all the entries in a table**
  - `SELECT (Col_name) FROM (Table_name);` --**Display only specific columns**
  - `SELECT * FROM (Table_name) ORDER BY (Col_name);` --**Display all the entries in a table in a specific order**
  - `SELECT * FROM (Table_name) LIMIT n;` -- **Display all the attributes in a table with a Limit**
  - `SELECT DISTINCT (Col_name) FROM (Table_name);` --**Display all the distinct entries of an attribute in a table**
- Functions in SQL -
  - `SELECT COUNT(Col_name) FROM (Table_name);`
  - `SELECT (Col_name) AS (_____) FROM (Table_name);`
  - `SELECT * FROM (Table_name) WHERE (Col_name) (define condition);`
  - `UPDATE (Table_name) SET (Col_name) = (New_value) WHERE (Col_name) = (Old_value)`
  - `SELECT (Col_name) FROM (Table_name) WHERE (Col_name) > CONDITION`



# Datatypes and syntax for SQL

-- We need to define variable datatype whenever we create attributes in a table

INT -- Whole Numbers

DECIMAL(M, N) -- Decimal Numbers - Exact value

VARCHAR(1) -- String of text of length 1, this can be variable

DATETIME -- 'YYYY-MM-DD HH:MM:SS' used for recoding exact time

-- Any query that you type in SQL needs to end with a semi-colon (;) in order to be valid

-- 1. Creation of a table

```
CREATE TABLE (Table_name1)(  
    (Attribute_name1) (Data_type1) PRIMARY KEY,  
    (Attribute_name2) (Data_type2),  
    (Attribute_name3) (Data_type3)
```



-- Actual Query to be run in editor

```
CREATE TABLE Student(  
    NUID INT PRIMARY KEY,  
    First_name VARCHAR(12),  
    Last_name VARCHAR(12),  
    Drone_id INT  
);
```

-- 2. Inserting/Deleting an attribute in a table

```
ALTER TABLE (Table_name) ADD (Attribute_name) (Data_type);
```



```
ALTER TABLE Student ADD Age INT;  
ALTER TABLE Student DROP COLUMN Age;
```

# Continued:

## Actual Query

-- 3. DELETING A TABLE

DROP TABLE (Table\_name)



DROP TABLE Student

-- 4. INSERTING DATA INTO A TABLE (enter elements in the correct order)

INSERT INTO (Table\_name) VALUES()



INSERT INTO Student  
VALUES(9489312, 'Puranjit', 'Singh', 2);

--5. DISPLAY THE TABLE

SELECT \* FROM (Table\_name)



SELECT Attribute\_name(multiple) FROM (Table\_name)

SELECT \* FROM Student;  
SELECT NUID, Last\_name FROM Student;

-- 6. UPDATE VALUES IN A TABLE

a. UPDATE (Table\_name) SET (Attribute\_name) = '\_\_\_' or

b. UPDATE (Table\_name) SET (Attribute\_name) = '\_\_\_'

WHERE (Attribute\_name) = '\_\_\_\_'



-- This will update all the attributes

UPDATE Truck\_table SET Truck\_name = 'Ford'

-- This will update selected attributes

UPDATE Truck\_table SET Truck\_name = 'Ford'

WHERE Truck\_name = 'Ford 150';

# Continued:

## -- 7. DELETING ROWS IN A TABLE

a. DELETE FROM (Table\_name); --delete all entries  
b. DELETE FROM (Table\_name)  
WHERE (Attribute\_name) = '\_\_\_\_'



## Actual Query

a. DELETE FROM Drone;  
b. DELETE FROM Drone  
WHERE Drone\_id = 2;

---

## -- 8. FILTERING IN A DATABASE

a. SELECT (Attribute\_name)  
FROM (Table\_name) WHERE (Attribute\_name) <= 3;  
b. SELECT (Attribute\_name) FROM  
(Table\_name)  
ORDER BY (Attribute\_name) ASC/DESC  
-- MULTIPLE CONDITIONS  
c. SELECT (Attribute\_name1), (Attribute\_name2) FROM  
FROM (Table\_name) WHERE NUID <= 3 AND Drone\_id= 1;



a. SELECT First\_name, Drone\_id  
FROM Student WHERE NUID <= 3;  
b. SELECT Drone\_name, Date\_p FROM  
Drone ORDER BY Date\_p ASC/DESC  
c. SELECT First\_name, Drone\_id  
FROM Student  
WHERE NUID <= 3 AND Drone\_id = 1;  
d. SELECT \* FROM Student  
WHERE First\_name IN ('Jack', 'Nancy', 'Paul');