Submarine Data Dive

Navigating Data Challenges Beneath the Surface

Intro to Data MOOC Challenge

Presented by: Gemma Kingshott

My Mission

.– ––. . –. – ––.––

–.–– ––– ..– .–. / … .––. . –.–. .. .– .–.. .. … – / –.. .– – .– / … –.– .. .–.. .–.. … / .– .–. . / .–. . ––.– ..– .. .–. . –.. / ..–. ––– .–. / .– / … . .–. .. . … / ––– ..–. / – ––– .––. / … . –.–. .–. . – / – .– … –.– … –.–.–– / –.–– ––– ..– .–. / –– .. … … .. ––– –. ––..–– / .. ..–. / –.–– ––– ..– / –.–. …. ––– ––– … . / – ––– / .– –.–. –.–. . .––. – / .. – ––..–– / .. … / – ––– / –.– . . .––. / – …. . / –. .– – .. ––– –. / … .– ..–. . / –… –.–– / –.–. ––– –. ––.– ..– . .–. .. –. ––. / …– .– .–. .. ––– ..– … / –.. .– – .– –… .– … . / –– .– –. .– ––. . –– . –. – / – .– … –.– … .–.–.– –… ––– .– .–. –.. / – …. . / … ..– –… –– .– .–. .. –. . ––..–– / .– –. –.. / –.–. .– .––. – .– .. –. / .–. . ––. . –. – / .–– .. .–.. .–.. / –… . / – …. . .–. . / – ––– / ––. ..– .. –.. . / –.–– ––– ..– .–.–.–

––. .–. . ––. / –… .–. .. – .– .. –. ––..–– / –.. . ..–. . –. –.–. . / … . –.–. .–. . – .– .–. –.––

My Mission

Agent,

Your specialist Data skills are required for a series of top secret tasks! Your mission, if you choose to accept it, is to keep the nation safe by conquering various database management tasks.

Board the submarine, and Captain Regent will be there to guide you.

Greg Britain,
Defence Secretary

Task 1

Find enemy coordinates

The Code

```
CREATE TABLE coordinates_log (
id INT PRIMARY KEY AUTO_INCREMENT,
latitude DECIMAL(8, 5),
longitude DECIMAL(8, 5),
location_type VARCHAR(50),
signal_strength INT

INSERT INTO coordinates_log (latitude,
longitude, location_type,
signal_strength) VALUES
(34.05223, -118.24368, 'enemy', 85),
(37.77493, -122.41942, 'ally', 72),
(36.16994, -115.13983, 'neutral', 65),
(40.71278, -74.00597, 'enemy', 90);
```

The Solution

Use the WHERE clause

```
SELECT id, latitude, longitude, location_type
FROM coordinates_log
WHERE location_type = 'enemy';
```

	id	latitude	longitude	location_type
1	1	34.05223	-118.24368	enemy
2	4	40.71278	-74.00597	enemy

The Answer

We have two enemies in our midst! Located at:
(34.05223, -118.24368)
(40.71278, -74.00597)

Task 2

Check which torpedoes are armed and which need rearming before launch.

The Code

```
CREATE TABLE torpedo_inventory (
torpedo_id INT PRIMARY KEY
AUTO_INCREMENT,
status VARCHAR(50)

INSERT INTO torpedo_inventory (status)
VALUES
('armed'),
('disarmed'),
('armed'),
('disarmed');
```

The Solution

1. Torpedo Status

```
SELECT torpedo_id, status
FROM torpedo_inventory;
```

	torpedo_id	status
1	1	armed
2	2	disarmed
3	3	armed
4	4	disarmed

2. Disarmed Torpedoes

Use WHERE clause

```
SELECT torpedo_id, status
FROM torpedo_inventory
WHERE status = 'disarmed';
```

	torpedo_id	status
1	2	disarmed
2	4	disarmed

The Answer

Number 2 and 4 torpedoes need rearming

Task 3

Filter out any unencrypted messages and report them.

The Code

```
CREATE TABLE communications (
message_id INT PRIMARY KEY AUTO_INCREMENT,
message TEXT,
encryption_status VARCHAR(50)

INSERT INTO communications (message,
encryption_status)
VALUES
('Proceed to the next checkpoint.',
'encrypted'),
('Send backup immediately.', 'unencrypted'),
('Engaging enemy forces.', 'encrypted'),
('Hold current position.', 'unencrypted');
```

The Solution

Use the WHERE clause

```
SELECT message_id, encryption_status, message
FROM communications
WHERE encryption_status = 'unencrypted';
```

	message_id	encryption_status	message
1	2	unencrypted	Send backup immediately.
2	4	unencrypted	Hold current position.

The Answer

2 Unencrypted messages should be reported.
Both messages could compromise our location and/or the location of allied forces

Task 4

See how much fuel each tank has, to make sure each one has sufficient fuel for the operations.

The Code

```
CREATE TABLE fuel_tanks (
tank_id INT PRIMARY KEY AUTO_INCREMENT,
fuel_amount DECIMAL(10, 2)

INSERT INTO fuel_tanks (fuel_amount)
VALUES
(2000.50),
(1500.75),
(1800.00),
(1700.25);
```


The Solution

Use the SELECT and ORDER BY clauses

```
SELECT tank_id, fuel_amount
FROM fuel_tanks
ORDER BY fuel_amount asc;
```

	tank_id	fuel_amount
1	2	1,500.75
2	4	1,700.25
3	3	1,800
4	1	2,000.5

The Answer

Unsure what is the minimum fuel level to complete operations. Yet, no fuel tanks are empty and the lowest level recorded is tank 2 with approximately 1500 units of fuel

Task 4b

Notify operatives to re-fuel

We need 1750 fuel in each tank to complete our operation. We want to see which fuel tanks need re-fuelling and by how much

```
SELECT tank_id, fuel_amount, (1750 - fuel_amount) AS fuel_required
FROM fuel_tanks
WHERE fuel_amount <= 1750
ORDER BY fuel_amount ASC;
```

	tank_id	fuel_amount	fuel_required
1	2	1,500.75	249.25
2	4	1,700.25	49.75

Fuel tanks 2 and 4 need refuelling by 249.25 and 49.75 respectively to complete the operation

Task 4c

Calculate total amount of fuel

Can I use the other fuel tanks to cover the fuel discrepancy for my operation?

If so, I need to check we have 7000 units of fuel

Use the SELECT SUM() query

```
SELECT SUM(fuel_amount) FROM fuel_tanks;
```

	SUM(fuel_amount)
1	7,001.5

I have a total number of 7001.5 units of fuel so the operation can commence

Task 5

Assess if there's any damaged equipment on the submarine that needs immediate repair.

The Code

```
CREATE TABLE equipment_status (
equipment_id INT PRIMARY KEY
AUTO_INCREMENT,
equipment_name VARCHAR(100),
equipment_condition VARCHAR(50)

INSERT INTO equipment_status
(equipment_name, equipment_condition)
VALUES
('Sonar System', 'good'),
('Navigation System', 'damaged'),
('Communication System', 'repairing'),
('Propulsion System', 'good'),
('Radar System', 'damaged');
```

The Solution

Use the SELECT and WHERE clauses

```
SELECT equipment_condition, equipment_id, equipment_name
FROM equipment_status
WHERE equipment_condition = 'Damaged';
```

	equipment_condition	equipment_id	equipment_name
1	damaged	2	Navigation System
2	damaged	5	Radar System

The Answer

Yes, the Navigation and Radar systems are damaged and need repairing

Task 5b

Assess compromised equipment

The Code

```
CREATE TABLE equipment_status (
equipment_id INT PRIMARY KEY
AUTO_INCREMENT,
equipment_name VARCHAR(100),
equipment_condition VARCHAR(50)

INSERT INTO equipment_status
(equipment_name, equipment_condition)
VALUES
('Sonar System', 'good'),
('Navigation System', 'damaged'),
('Communication System', 'repairing'),
('Propulsion System', 'good'),
('Radar System', 'damaged');
```

The Solution

Add the WHERE and IN statement to include extra conditions

```
SELECT equipment_condition, equipment_id, equipment_name
FROM equipment_status
WHERE equipment_condition in ('Damaged', 'Repairing')
ORDER BY equipment_condition asc;
```

	equipment_condition	equipment_id	equipment_name
1	damaged	2	Navigation System
2	damaged	5	Radar System
3	repairing	3	Communication System

The Answer

3 pieces of equipment are compromised until fixed

Task 5c

See the order of Priority of equipment that needs to be fixed

1. Use ALTER TABLE to add a new “Priority” column

```
ALTER TABLE equipment_status ADD COLUMN Priority VARCHAR(20);
```

2. Use CASE in an UPDATE query to add conditional values to the new column

```
UPDATE equipment_status
SET Priority = CASE
WHEN equipment_condition IN (‘damaged’, 'repairing') and equipment_name = 'Communication System' THEN 'Critical'
WHEN equipment_condition = 'damaged' AND equipment_name = 'Navigation System' THEN 'Critical'
WHEN equipment_condition = 'damaged' THEN 'High'
WHEN equipment_condition = 'repairing' THEN 'Medium'
ELSE NULL
END;
```

3. SELECT the table and order by the Priority

```
SELECT *
FROM equipment_status
WHERE Priority IN ('Critical', 'High', 'Medium', 'Low')
ORDER BY Priority ASC;
```

	equipment_id	equipment_name	equipment_condition	Priority
1	2	Navigation System	damaged	Critical
2	3	Communication System	repairing	Critical
3	5	Radar System	damaged	High

The 3 compromised pieces of equipment are now in priority order. Damaged equipment results in a higher priority until the problem is identified.

Task 5d

Find crew that can fix equipment

The Code

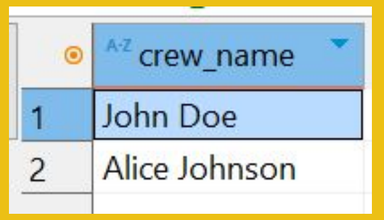
```
CREATE TABLE equipment_status (
equipment_id INT PRIMARY KEY AUTO_INCREMENT,
equipment_name VARCHAR(100),
equipment_condition VARCHAR(50)

INSERT INTO equipment_status
(equipment_name, equipment_condition)
VALUES
('Sonar System', 'good'),
('Navigation System', 'damaged'),
('Communication System', 'repairing'),
('Propulsion System', 'good'),
('Radar System', 'damaged');
```

The Solution

Add the WHERE and AND statement to include extra conditions from the crew_members table

```
SELECT crew_name
FROM crew_members
WHERE active = 1 AND duty_status = 'on duty';
```



	crew_name
1	John Doe
2	Alice Johnson

The Answer

John or Alice are available to fix the equipment

Task 6

Identify any unauthorized access attempts in the submarine's control room.

The Code

```
CREATE TABLE access_logs (
access_id INT PRIMARY KEY AUTO_INCREMENT,
user_id INT,
location VARCHAR(100),
access_time DATETIME,
authorized TINYINT(1)

INSERT INTO access_logs (user_id, location,
access_time, authorized) VALUES
(1, 'control room', '2024-11-01 09:15:00', 1),
(2, 'engine room', '2024-11-01 12:30:00', 1),
(3, 'control room', '2024-11-02 08:45:00', 0),
(4, 'control room', '2024-11-03 10:00:00', 0);
```

The Solution

Use the WHERE clause

```
SELECT user_id, access_time
FROM access_logs
WHERE location = 'control room' AND authorized = 0;
```

	user_id	access_time
1	3	2024-11-02 08:45:00
2	4	2024-11-03 10:00:00

The Answer

There are 2 unauthorised access attempts to the control room

Task 7

Find the crew members who have been on duty the most in the last 30 days.

The Code

```
CREATE TABLE crew_duty_log (
duty_id INT PRIMARY KEY AUTO_INCREMENT,
crew_id INT,
duty_status VARCHAR(50), -- 'on duty', 'off duty'
duty_start DATETIME,
FOREIGN KEY (crew_id) REFERENCES crew_members(crew_id)

INSERT INTO crew_duty_log (crew_id, duty_status, duty_start) VALUES
(1, 'on duty', '2025-03-05 08:00:00'),
(1, 'on duty', '2025-03-10 08:30:00'),
(1, 'on duty', '2025-03-15 09:00:00'),
(1, 'on duty', '2025-03-30 12:00:00'),
and so on... ;
```

The Solution

Use COUNT(*) to create a new column to count the number of shifts and use GROUP BY to group the results by the crew ID.

```
SELECT crew_id, COUNT(*) AS shift_count
FROM crew_duty_log
WHERE duty_status = 'on duty' AND duty_start >= NOW() - INTERVAL 1 MONTH
GROUP BY crew_id
ORDER BY shift_count DESC
LIMIT 3;
```

	crew_id	shift_count
1	3	6
2	2	5
3	1	4

The Answer

The crew member with ID number 3 completed the most shifts in the last 30 days, totalling 6 shifts.

Task 7b

Find the names of the crew members who have been on duty the most in the last 30 days.

The Code

```
CREATE TABLE crew_duty_log (
duty_id INT PRIMARY KEY AUTO_INCREMENT,
crew_id INT,
duty_status VARCHAR(50), -- 'on duty', 'off
duty'
duty_start DATETIME,
FOREIGN KEY (crew_id) REFERENCES
crew_members(crew_id)

INSERT INTO crew_duty_log (crew_id,
duty_status, duty_start) VALUES
(1, 'on duty', '2025-03-05 08:00:00'),
(1, 'on duty', '2025-03-10 08:30:00'),
(1, 'on duty', '2025-03-15 09:00:00'),
(1, 'on duty', '2025-03-30 12:00:00'),
and so on... ;
```

The Solution

Use cdl and cm to reference the 'crew duty log' and 'crew members' tables in the SELECT query. Then use JOIN to combine the crew_id shift column. The ON clause defines the condition for the JOIN.

```
SELECT
    cdl.crew_id,
    cm.crew_name,
    COUNT(*) AS shift_count
FROM crew_duty_log cdl
JOIN crew_members cm ON cdl.crew_id = cm.crew_id
WHERE cdl.duty_status = 'on duty' AND duty_start >= NOW()
- INTERVAL 1 MONTH
GROUP BY cdl.c
ORDER BY shift
LIMIT 3;
```

	crew_id	crew_name	shift_count
1	3	Alice Johnson	6
2	2	Jane Smith	5
3	1	John Doe	4

The Answer

Alice Johnson was the crew member that completed the most shifts in the last 30 days.

Task 8

Find equipment that has been marked as "damaged" more than 3 times in the past month.

The Code

```
CREATE TABLE equipment_log (
log_id INT PRIMARY KEY AUTO_INCREMENT,
equipment_id INT,
equipment_condition VARCHAR(50),
change_time DATETIME,
FOREIGN KEY (equipment_id) REFERENCES
equipment_status(equipment_id)

INSERT INTO equipment_log (equipment_id,
equipment_condition, change_time) VALUES
-- Equipment 1 (Sonar System) - Mostly Good
(1, 'good', '2025-03-21 11:00:00'),
(1, 'good', '2025-03-29 12:00:00'),
and so on...
```

The Solution

Use COUNT(*) AS and WHERE to create a new column to count the times a piece of equipment has been damaged based on the equipment condition and time. Use the HAVING clause to filter the group to only include counts of > 3.

```
select equipment_id,
COUNT(*) AS damage_count
from equipment_log
where equipment_condition = 'Damaged' AND change_time >=
NOW() - INTERVAL 1 month
group by equipment_id
having damage_count > 3
order by damage_count DESC;
```

Notes: Make sure the HAVING clause is after GROUP BY and before ORDER BY.

	equipment_id	damage_count
1	5	4

The Answer

Equipment ID no. 5 has been damaged 4 times in the past month

Task 8b

Find the name of the piece of equipment that's been damaged the most

The Code

```
CREATE TABLE crew_duty_log (
duty_id INT PRIMARY KEY AUTO_INCREMENT,
crew_id INT,
duty_status VARCHAR(50), -- 'on duty', 'off
duty'
duty_start DATETIME,
FOREIGN KEY (crew_id) REFERENCES
crew_members(crew_id)

INSERT INTO crew_duty_log (crew_id,
duty_status, duty_start) VALUES
(1, 'on duty', '2025-03-05 08:00:00'),
(1, 'on duty', '2025-03-10 08:30:00'),
(1, 'on duty', '2025-03-15 09:00:00'),
(1, 'on duty', '2025-03-30 12:00:00'),
and so on... ;
```

The Solution

Use el and es to reference the 'equpipment_log' and 'equipment_status' tables in the SELECT query. Then use JOIN to combine the crew_id shift column.

```
SELECT
el.equipment_id,
es.equipment_name,
COUNT(*) AS damage_count
FROM equipment_log el
JOIN equipment_status es ON el.equipment_id = es.equipment_id
WHERE el.equipment_condition = 'Damaged'AND el.change_time >=
NOW() - INTERVAL 1 MONTH
GROUP BY el.equipment_id, es.equipment_name
HAVING damage_count > 3
ORDER BY damage_count DESC;
```

	equipment_id	equipment_name	damage_count
1	5	Radar System	4

The Answer

The Radar system has been the most damaged in the last month

Troubleshooting Learnings

Removing Duplicates

Helpful link:
https://phoenixnap.com/kb/mysql-remove-duplicate-rows#:~:text=After%20confirming%20that%20a%20database,clause%2C%20and%20the%20DISTINCT%20keyword.

1. Count the total number of rows

```
SELECT COUNT(id) AS
totalnocoordinates
FROM coordinates_log;
```

	totalnocoordinates
1	16

2. Selected the data and created a new column called “number” that adds 1 for each duplicated row

```
SELECT *, ROW_NUMBER() OVER
(PARTITION BY latitude, longitude,
location_type ORDER BY id)
AS number
FROM coordinates_log;
```

	id	latitude	longitude	location_type	signal_strength	number
1	1	34.05223	-118.24368	enemy	85	1
2	5	34.05223	-118.24368	enemy	85	2
3	9	34.05223	-118.24368	enemy	85	3
4	13	34.05223	-118.24368	enemy	85	4
5	3	36.16994	-115.13983	neutral	65	1
6	7	36.16994	-115.13983	neutral	65	2
7	11	36.16994	-115.13983	neutral	65	3
8	15	36.16994	-115.13983	neutral	65	4
9	2	37.77493	-122.41942	ally	72	1
10	6	37.77493	-122.41942	ally	72	2
11	10	37.77493	-122.41942	ally	72	3
12	14	37.77493	-122.41942	ally	72	4
13	4	40.71278	-74.00597	enemy	90	1
14	8	40.71278	-74.00597	enemy	90	2

3. Filter to show first instances and enemy coordinates

```
SELECT * FROM
(SELECT *, ROW_NUMBER() OVER
(PARTITION BY latitude, longitude, location_type
ORDER BY id)
AS number
FROM coordinates_log) t
WHERE number = 1 and location_type = 'enemy';
```

	id	latitude	longitude	location_type	signal_strength	number
1	1	34.05223	-118.24368	enemy	85	1
2	4	40.71278	-74.00597	enemy	90	1

Troubleshooting Learnings

Other quick wins

1. Confirm my database was available

```
show databases;
```

2. Show all tables in database

```
show full tables;
```

3. Dropping duplicate tables

```
drop table if exists coordinates_log;
```

4. Check host name using mySQL

```
select @@hostname;
```

Troubleshooting Learnings

Creating a Common Table Expression (CTE)

A CTE is used to define a temporary result set (e.g. a column) that can be referenced within a single query. It can be referenced multiple times within the query.

I didn't realise this was temporary when I was doing it so it wasn't possible to quote it in a new select query.

I found it easier to read and understand at the time of exploring how to get the result I wanted.

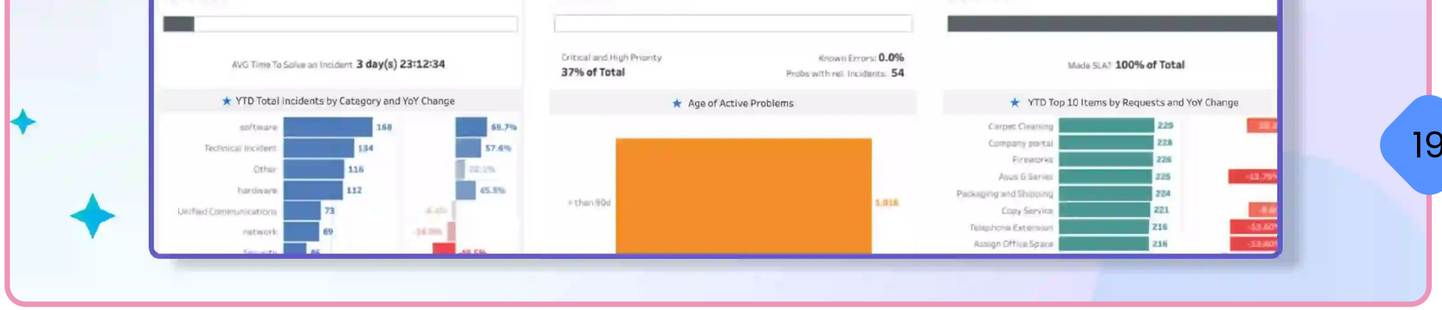
This was the CTE I wrote:

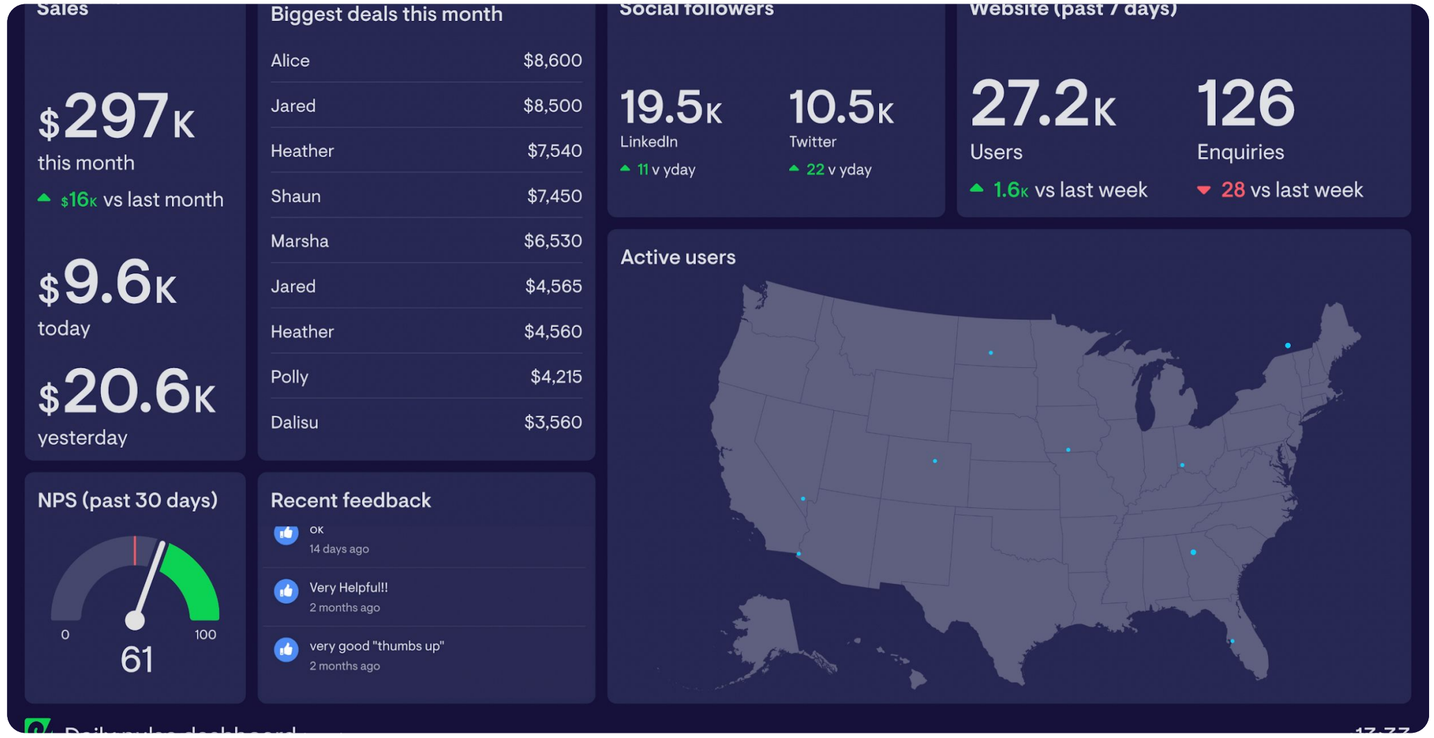
```
with cte_Priority AS
(select -- Cannot select columns here!
CASE
WHEN equipment_condition = 'damaged' and equipment_name = 'Navigation System' THEN 'Critical'
WHEN equipment_condition = 'damaged' THEN 'High'
WHEN equipment_condition = 'repairing' THEN 'Medium'
ELSE NULL
END AS "Priority"
FROM equipment_status)
SELECT *
FROM cte_Priority
WHERE Priority IN ('Critical', 'High', 'Medium', 'Low')
ORDER BY Priority ASC;
```

	Priority
1	Critical
2	High
3	Medium

Next steps

- Shift database online (AWS
- Link geo data to present allied and enemy coordinates to put them on the map and identify their source
- Use the online database to create an online dashboard (Tableau, powerBI, Geckoboard etc.) of all the results presented today so Greg can get live results to make efficient decisions





Thank you!

Any questions?

Intro to Data MOOC Challenge

Presented by: Gemma Kingshott