

First Agenda

Item

Embedded SQL

→ Take your
Front end & connect
to db database

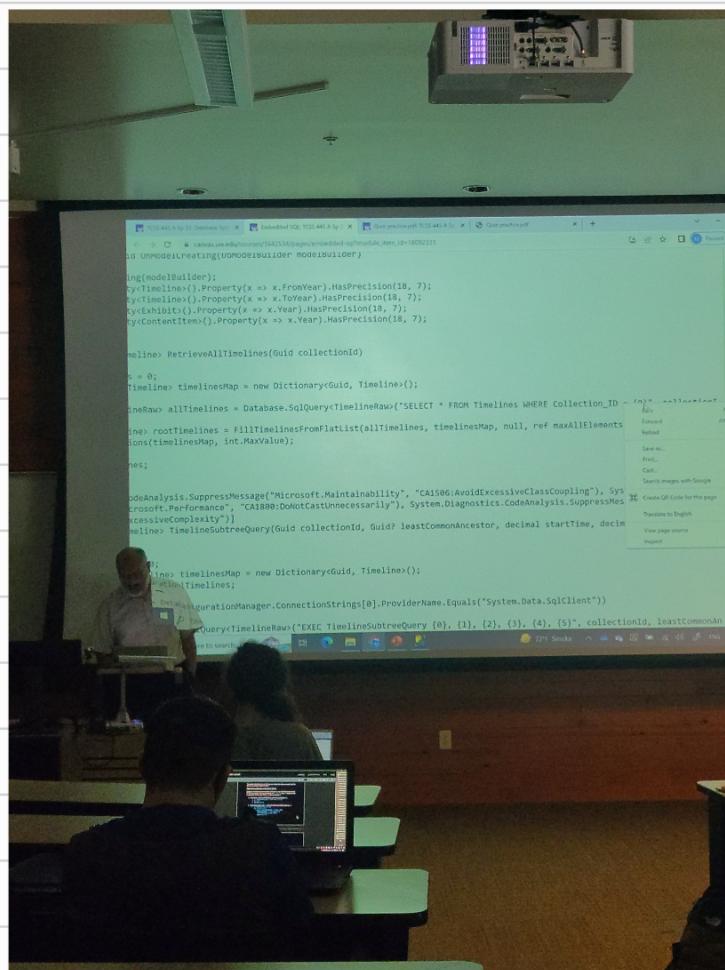
This will look similar
in my language's

Steps to connect

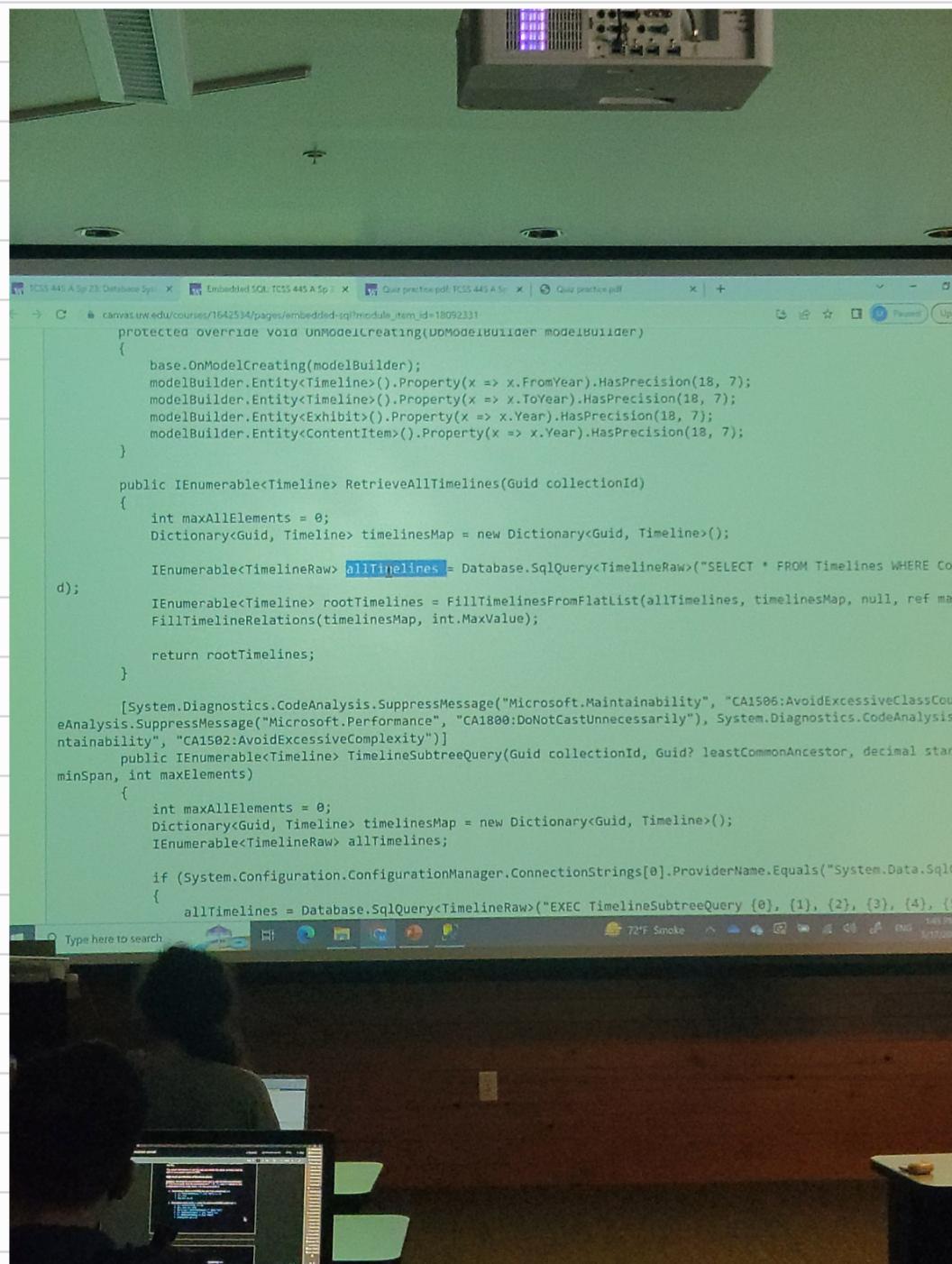
1. Get connection string

Ex (IP Address, Local host
& Name of specific database

2. Run query, since
the date range
appropriate may
exist already



Query
Result stored in
Variable



2 Main lines

- establish conn
- Request & Expr

New slide
week 8

Conn'g .SQL

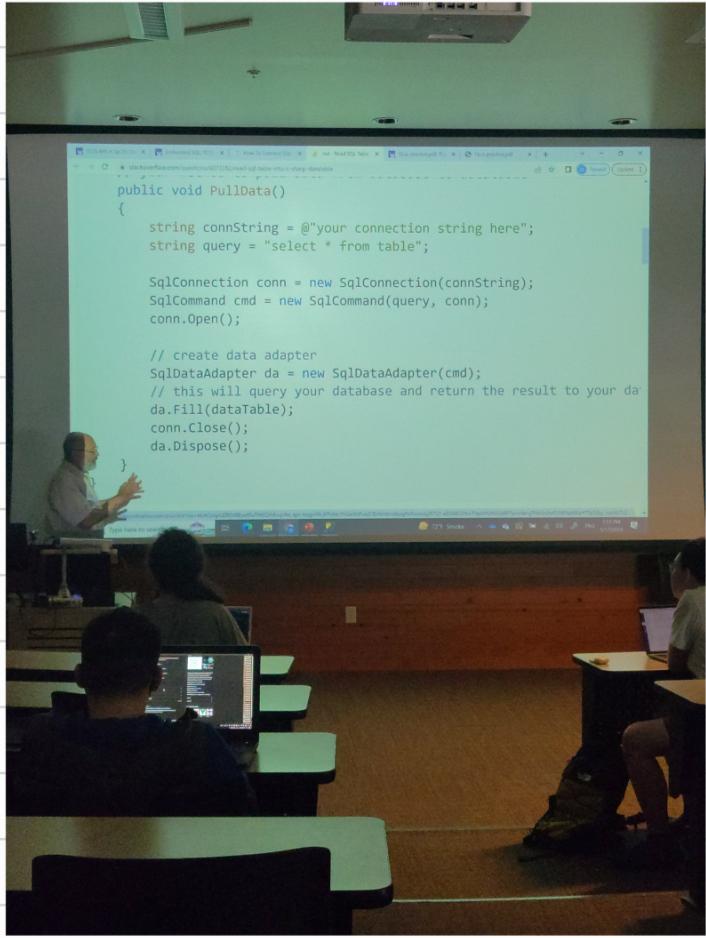
To ASP.NET

New slide

Conn'g to SQL

into L# database

From Exchange, flow, mi



Mendcy Canvas
Very Important
Part

Pretarance & how
to handle a -

To build an index
for the data

When searching, thing
about how this does
might be accessed
and its $O(n)$

Scheman Example

Here is a schema about battleships and the battles they fought in:

`Ships (name, Year Launched, country, numGuns, gunsize, displacement)`

`Battles (ship, battlename, result)`

A typical ships tuple would be: ('New Jersey', 1943, 'USA', 9, 16, 46000)

Which means that the battleship New Jersey was launched in 1943; it belonged to the USA, carried guns of size 16-inch (bore, or inside diameter of the barrel), and weighted 46,000 tons.

A typical tuple for Battle is: ('Hood', 'North Atlantic', 'sunk')

That is, H.M.S.Hood was sunk in the battle of the North Atlantic. The other possible results are 'damaged'.

Write the SQL as well as relational algebra for the following questions:

1. Which battleships launched before 1930 had 16-inch guns? List their names, their country, number of guns they carried.



Some Query
examples

From Ships

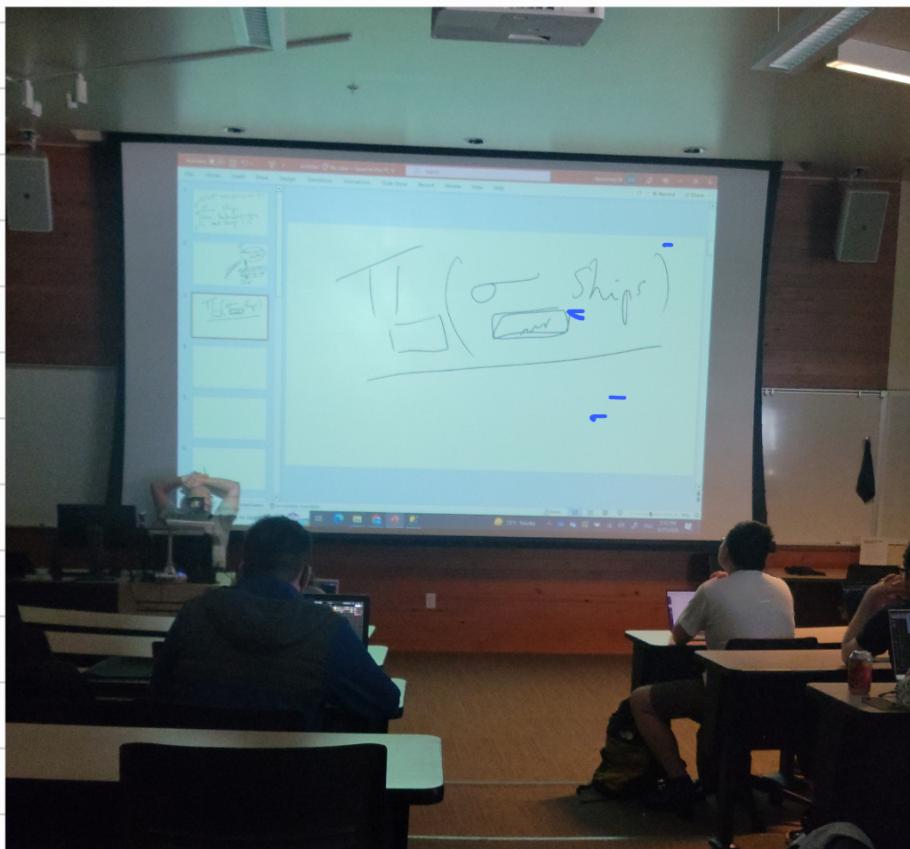
Where YearLaunch < 1930

And GunSize = 16

Select

Name, Country, Numbers

Relational Algebra





Second Quiz

which means that the battleship New Jersey was launched in 1943; it belonged to the USA, carried 9 guns of size 16-inch (bore, or inside diameter of the barrel), and weighted 46,000 tons.

A typical tuple for Battle is: ('Hood', 'North Atlantic', 'sunk')

That is, H.M.S.Hood was sunk in the battle of the North Atlantic. The other possible results are 'ok' and 'damaged'.

Write the SQL as well as relational algebra for the following questions:

1. Which battleships launched before 1930 had 16-inch guns? List their names, their country, and the number of guns they carried.
2. List for each country, the average number of guns carried by their battleships.
3. List all the pairs of countries that fought each other in battles. List each pair only once, and list them with the country that comes first in alphabetical order first.
4. List all the triplets of countries that fought each other in a battle and the battle name on the form of (Country 1, Country 2, BattleName). List each triplet only once, and list them with the country that had more ships in this battle first.

2) Select Country, Avg(numGuns)
as ~

1.) From Ships

3) Group by Country

Third Query
Select S1, S2
 dist. no
 S1. Country
 S2. Country

From Battles B1, Battles B2
Ships S1, Ships S2

Where b1.name, b1 = b2.name, b2
DIN

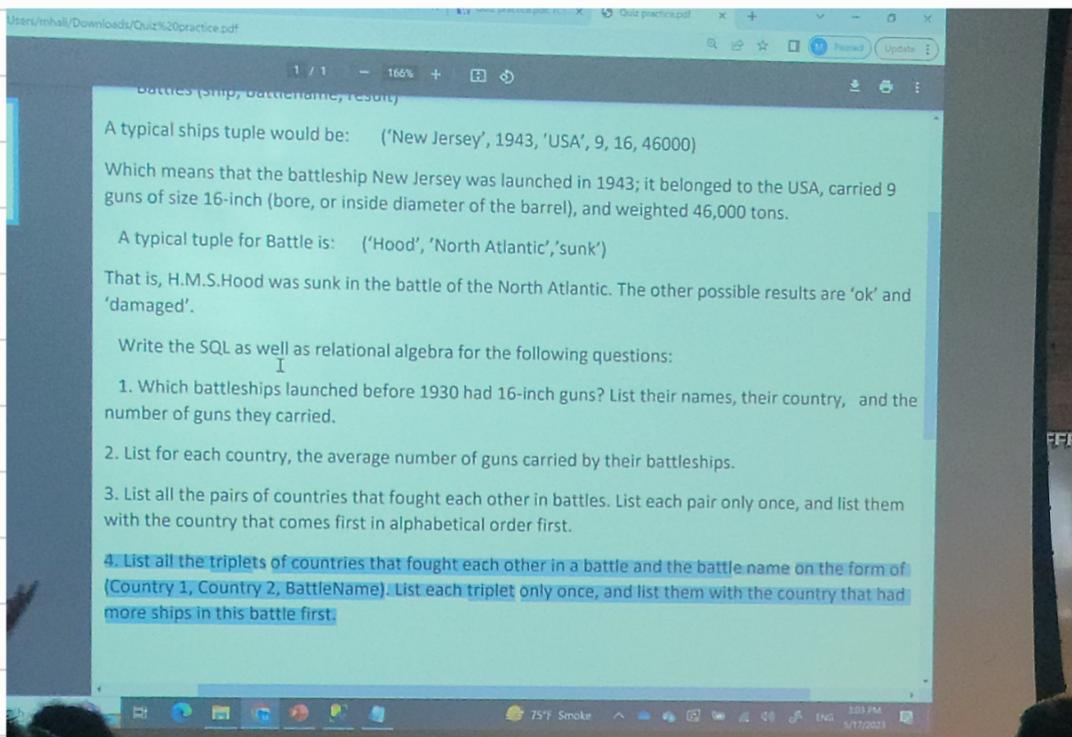
B1.name < > B2.name

and

B1.Name = S1.Name

and B2.Name = S2.Name

4th Query



Select distinct

S1.Country

S2.Country

B1.BattleName

From Bagges. B1

Bagges. B2

Ships. S1

Ships. S2

Where B1. BoatName = B2. BoatName
and B1. Name (→ B2. Name)

and B1. Name = S1. Name

and B2. Name = S2. Name

S1. Country S2. Country

JA

Join
(

Select S. Country,

B. BoatName

comes (the) as Ship

From Ships, Bagges.B

Machine S. Name B. Name

W S. Name = B' Name

Groupby S. Country, B. BattleName

?B

ON A.BattleName = B. BattleName

and A.Country <> B.Country

and A.ShipCount



?B.ShipCount

```
File Edit Format View Help
SELECT distinct A.Country, B.Country, A.BattleName
(
    SELECT S.Country,B.BattleName, count(*) AS ShipCount
    FROM Ships S, Battles B
    Where S.Name = B.Name
    GROUP BY S.Country,B.BattleName
) A
JOIN
(
    SELECT S.Country,B.BattleName, count(*) AS ShipCount
    FROM Ships S, Battles B
    Where S.Name = B.Name
    GROUP BY S.Country,B.BattleName
) B
ON A.Country<>B.Country and A.BattleName = B.BattleName
AND A.ShipCount>B.ShipCount
```

