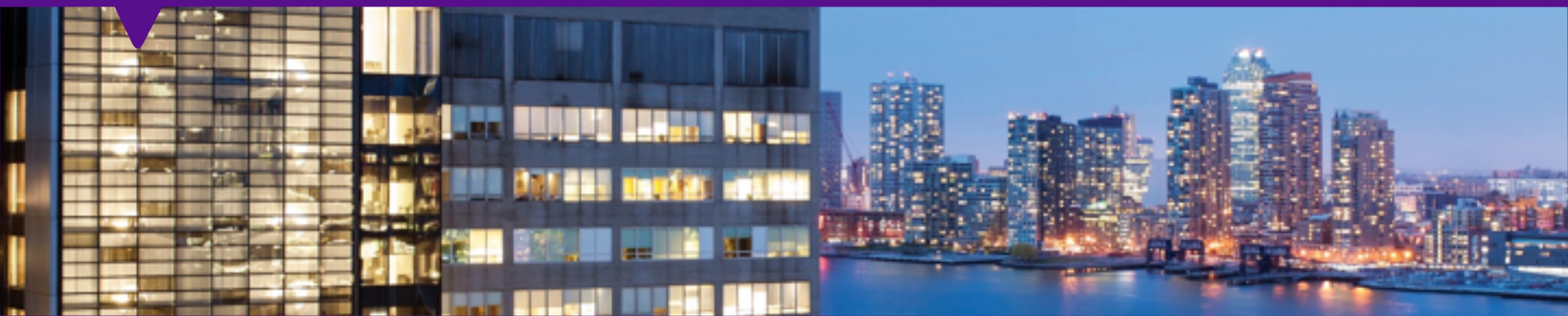
NYU Langone Health

NYU School of Medicine

INTRODUCTION TO SQL

Dr Simon Jones

Professor of Population Health



Agenda

- Introduction to SQL
  - SELECT
  - CREATE
- Query optimization
  - A Query optimization Checklist
  - The Tools of the Trade

NYU Langone Health

GENERAL POINTS

NYU Langone Health

What is a Database?

- A Database System is a Computerised Record Keeping System
- Rather Like an Electronic Filing Cabinet
- The Data can be Added to, Deleted, Modified etc…
- The Data Contained is of the Same Type
- Would Not Have a Database Containing Patient Records and the Sales Records of a Pet Shop, For Example

History of SQL

- Selected History

1970 Codd published the paper, "A Relational Model of Data for Large Shared Data Banks"

1979 Relational Software, Inc. (now Oracle) first commercially available SQL

1986 First ANSI standard

1988 Sybase partners with Microsoft

1992 Major revisions to ANSI standard

1993 Microsoft buys SQL Server from Sybase

Some key dialects

SQL Server (Microsoft)
MySQL (Sun/Oracle)
PL/SQL (Oracle)
PL/PSM (PostgresSQL)

SQL

NYU Langone Health

Key SQL Commands

SELECT - extracts data from a database

UPDATE - updates data in a database

DELETE - deletes data from a database

INSERT INTO - inserts new data into a database

CREATE DATABASE - creates a new database

ALTER DATABASE - modifies a database

CREATE TABLE - creates a new table

ALTER TABLE - modifies a table

DROP TABLE - deletes a table

CREATE INDEX - creates an index (search key)

DROP INDEX - deletes an index

SELECT Statement Syntax

```
SELECT [ ALL | DISTINCT ]
    [TOP ( expression ) [PERCENT]
    < select_list >
    [ INTO new_table ]
    [ FROM { <table_source> } [ ,...n ] ]
    [ WHERE <search_condition> ]
    [ <GROUP BY> ]
    [ <ORDER BY> ]
    [ HAVING < search_condition > ]
```

Online exercise from
http://www.w3schools.com/sql/sql_select.asp

SQL Aggregate Functions

AVG() - Returns the average value

COUNT() - Returns the number of rows

FIRST() - Returns the first value

LAST() - Returns the last value

MAX() - Returns the largest value

MIN() - Returns the smallest value

SUM() - Returns the sum

SQL Scalar Functions

UPPER() - Converts a field to upper case

LOWER() - Converts a field to lower case

SUBSTRING() - Extract characters from a text field

LEN() - Returns the length of a text field

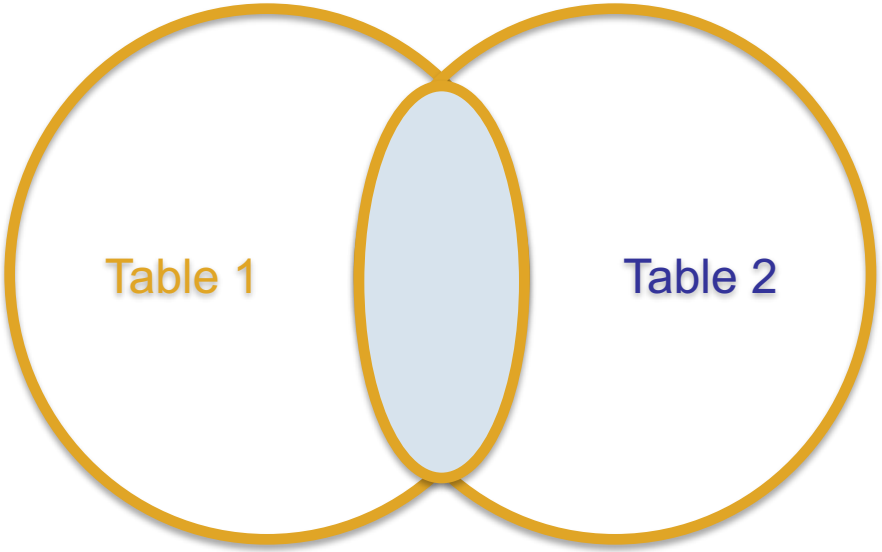
ROUND() - Rounds a numeric field to the number of decimals specified

GETDATE() - Returns the current system date and time

CONVERT() - Formats how a field is to be displayed

Inner Joins

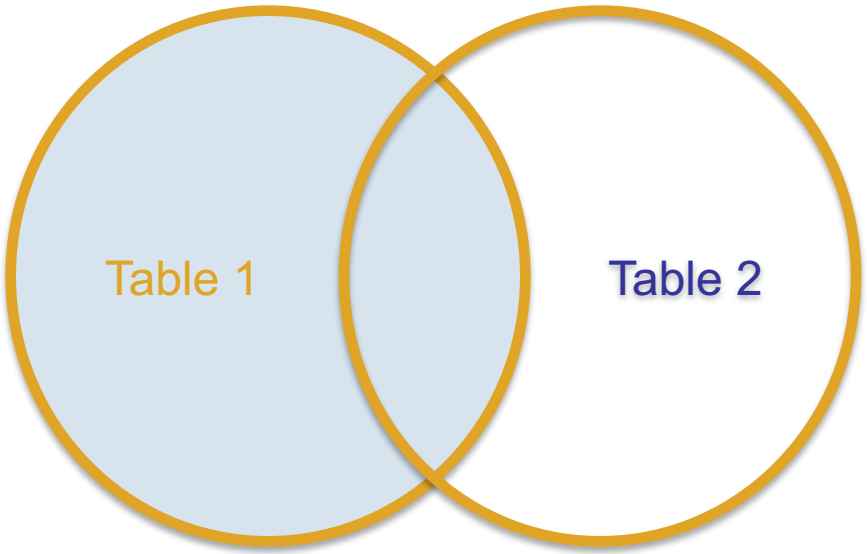
```
SELECT column_name(s)
FROM table1
INNER JOIN table2
    ON table1.column_name=table2.column_name;
```



NYU Langone Health

Left Joins

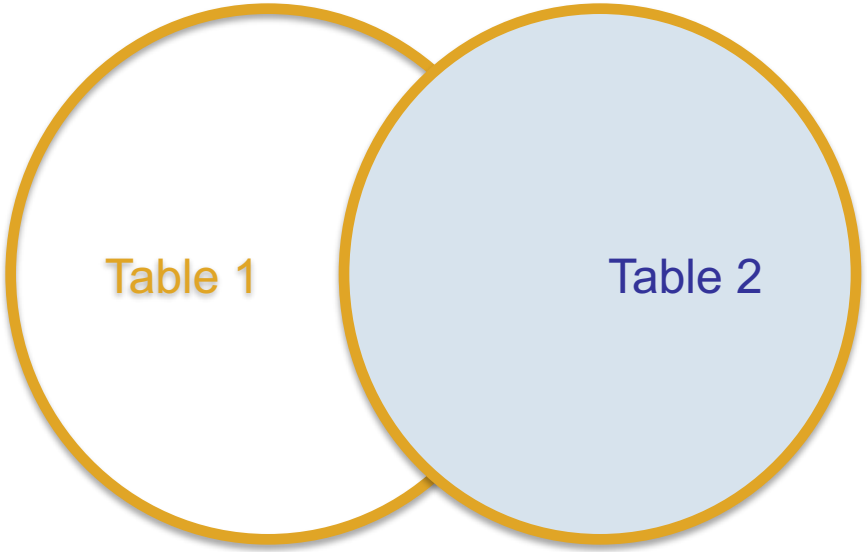
```
SELECT column_name(s)
FROM table1
LEFT OUTER JOIN table2
   ON table1.column_name=table2.column_name;
```



NYU Langone Health

Right Joins

```
SELECT column_name(s)
FROM table1
RIGHT OUTER JOIN table2
   ON table1.column_name=table2.column_name;
```



NYU Langone Health

CREATE TABLE

The CREATE TABLE statement is used to create a table in a database.

```
CREATE TABLE table_name
(
    column_name1 data_type(size) [NOT] NULL [UNIQUE] PRIMARY
KEY,
    column_name2 data_type(size) [NOT] NULL [UNIQUE],
    column_name3 data_type(size) [NOT] NULL [UNIQUE],
    ....
);
```

CREATE TABLE Example

```
CREATE TABLE Persons
(
    PersonID int NOT NULL PRIMARY KEY,
    LastName varchar(50) NULL,
    FirstName varchar(50) NULL,
    Address varchar(255) NULL,
    City varchar(30) NULL,
    Postcode varchar(13) NULL
);
```

NYU Langone Health

MySQL Data Types - Numeric

https://dev.mysql.com/doc/refman/5.7/en/data-types.html

Type	Storage (Bytes)	Minimum Value (Signed/Unsigned)	Maximum Value (Signed/Unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INT	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615

NYU Langone Health

MySQL Data Types -Strings

https://dev.mysql.com/doc/refman/5.7/en/string-types.html

CHAR vs vARCHAR

alue	CHAR(4)	Storage Required	VARCHAR(4)	Storage Required
''	' '	4 bytes	''	1 byte
'ab'	'ab '	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	5 bytes
'abcdefgh'	'abcd'	4 bytes	'abcd'	5 bytes

NYU Langone Health

MySQL Data Types - Dates

https://dev.mysql.com/doc/refman/5.7/en/date-and-time-types.html

Data Type	“Zero” Value
DATE	'0000-00-00'
TIME	'00:00:00'
DATETIME	'0000-00-00 00:00:00'
TIMESTAMP	'0000-00-00 00:00:00'
YEAR	0000

NYU Langone Health

QUERY OPTIMIZATION

NYU Langone Health

Don’t use SELECT *

```
select * from da_source
```

- Returns unnecessary data
- Causes application maintenance problems
- Can force a clustered index scan
- DB Engine must traverse the entire table
- Returns every column of every joined table

NYU Langone Health

Improving JOIN Performance

- JOIN columns should:
  - Share the same datatype
  - Be numeric datatypes, e.g. int
    - Consider using a surrogate key for non-numeric types
  - Be indexed
    - Unique indexes are best
  - Have mostly unique values
    - This helps to prevent table scans

Script #2

NYU Langone Health

Minimise the Number of JOINs

```
SELECT      *
FROM        some_table          a
INNER JOIN  another_table       b ON a.some_table_id         = b.some_table_id
LEFT JOIN   yet_another_table   c ON a.another_table_id      = c.another_table_id
INNER JOIN  and_another_one     d ON c.and_another_table_id = d.and_another_table_id
LEFT JOIN   and_theres_more     e ON d.and_theres_more_id    = e.and_theres_more_id
LEFT JOIN   and_so_it_goes_on   f ON c.and_so_it_goes_on_id  = f.and_so_it_goes_on_id
INNER JOIN  and_on              g ON e.and_on_id             = g.and_on_id
INNER JOIN  and_on_and_on       h ON c.and_on_and_on_id      = h.and_on_and_on_id
```

NYU Langone Health

Minimise the Number of JOINs

- Only JOIN to tables you actually need
- More JOINs ⟶ more I/O and more CPU
- Try to JOIN to a maximum of 4 tables
- Reduce JOIN count by denormalising
- OPTION (FORCE ORDER) forces JOIN order
  - It's best to leave JOIN order to the query optimizer

NYU Langone Health

Temp Table Recommendations

- Always delete a temporary table after use
- Use table variables instead for small datasets
  - Use fewer resources than temp tables
  - More likely to persist in memory
  - Fewer compilations
  - Not logged
- Create indexes on large temp tables
  - Doesn’t always work, but worth a try

Script #5

NYU Langone Health

Avoid Expensive Operators

- E.g. LIKE
  - Values enclosed in wildcards almost always cause a table scan
    - e.g. WHERE LastName LIKE '%Jones%'
- Negative operations are difficult to resolve efficiently
  - E.g. <> or NOT LIKE
    - Try to rewrite these another way if you can
- If only checking for existence, use EXISTS
  - If there is a table scan, at least it will exit at first occurrence!

Functions in the WHERE clause

- The optimizer cannot select an index for columns inside functions in the WHERE clause
  - Hence can't use these columns in plan optimization!
- Columns in functions are treated as expressions
- Place column outside the function if possible
  - Push the function to the literal expression

```
-- Don't do this!
SELECT OrderID FROM NorthWind.dbo.Orders WHERE DATEADD(day, 15, OrderDate) = '07/23/2008'

-- Do this instead :)
SELECT OrderID FROM NorthWind.dbo.Orders WHERE OrderDate = DATEADD(day, -15, '07/23/2008')
```

NYU Langone Health

Understand How Indexes Work

- Two Types of Index
  - Clustered
    - Records in the table are physically stored, hence
    - One per table
    - Contains all table data on the leaf level
    - Leaf level is made up of data pages
    - A good analogy is a dictionary
  - Non-Clustered
    - Maximum of 249 per table
    - Only contains data for the indexed/included columns
    - Leaf level is made up of index pages
    - Each index row contains a row locator
    - A good analogy is the index of a reference book

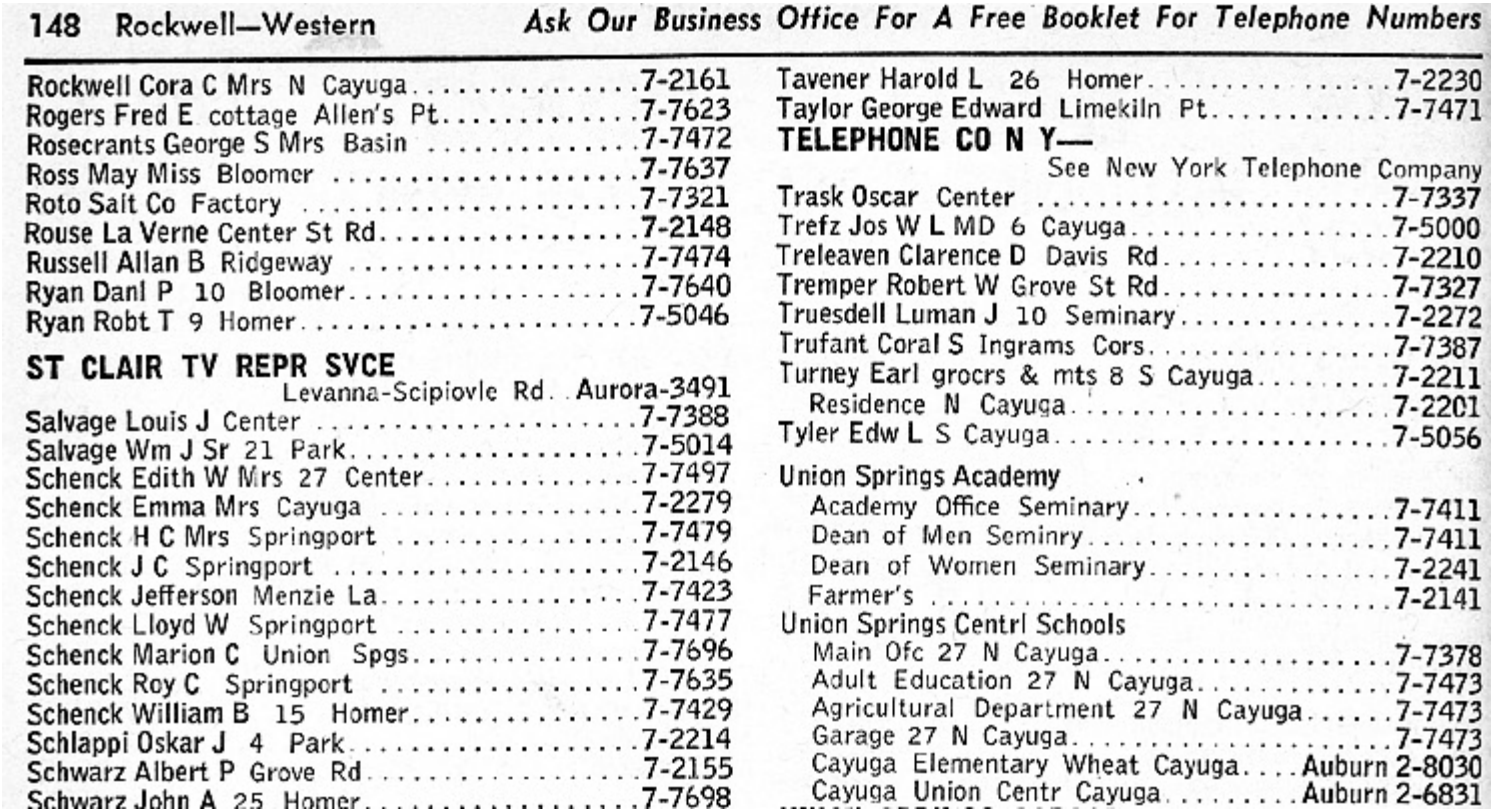
NYU Langone Health

Understand How Indexes Work

- Keep indexes in mind when designing queries:
  - Know the column order of the index you use
    - Use the head of the index!
    - E.g. A phonebook has a clustered index on LastName, FirstName
    - Easy to find WHERE LastName = ‘Bloggs’ AND FirstName = ‘Fred’
    - Not useful for finding WHERE FirstName = ‘Fred’
  - Use covering indexes where possible
    - Avoid bookmark lookups
  - The query optimizer may still decide not to use an index

Understand How Indexes Work

Clustered Indexes have been around for a while!



148 Rockwell—Western *Ask Our Business Office For A Free Booklet For Telephone Numbers*

Rockwell Cora C Mrs N Cayuga 7-2161
Rogers Fred E cottage Allen's Pt 7-7623
Rosecrants George S Mrs Basin 7-7472
Ross May Miss Bloomer 7-7637
Roto Salt Co Factory 7-7321
Rouse La Verne Center St Rd 7-2148
Russell Allan B Ridgeway 7-7474
Ryan Danl P 10 Bloomer 7-7640
Ryan Robt T 9 Homer 7-5046

ST CLAIR TV REPR SVCE
Levanna-Scipiovle Rd .. Aurora-3491

Salvage Louis J Center 7-7388
Salvage Wm J Sr 21 Park 7-5014
Schenck Edith W Mrs 27 Center 7-7497
Schenck Emma Mrs Cayuga 7-2279
Schenck H C Mrs Springport 7-7479
Schenck J C Springport 7-2146
Schenck Jefferson Menzie La 7-7423
Schenck Lloyd W Springport 7-7477
Schenck Marion C Union Spgs 7-7696
Schenck Roy C Springport 7-7635
Schenck William B 15 Homer 7-7429
Schlappi Oskar J 4 Park 7-2214
Schwarz Albert P Grove Rd 7-2155
Schwarz John A 25 Homer 7-7698

Tavener Harold L 26 Homer 7-2230
Taylor George Edward Limekiln Pt 7-7471

TELEPHONE CO N Y—
See New York Telephone Company

Trask Oscar Center 7-7337
Trefz Jos W L MD 6 Cayuga 7-5000
Treleaven Clarence D Davis Rd 7-2210
Tremper Robert W Grove St Rd 7-7327
Truesdell Luman J 10 Seminary 7-2272
Trufant Coral S Ingrams Cors 7-7387
Turney Earl grocrs & mts 8 S Cayuga 7-2211
Residence N Cayuga 7-2201
Tyler Edw L S Cayuga 7-5056

Union Springs Academy
Academy Office Seminary 7-7411
Dean of Men Seminry 7-7411
Dean of Women Seminary 7-2241
Farmer's 7-2141
Union Springs Centrl Schools
Main Ofc 27 N Cayuga 7-7378
Adult Education 27 N Cayuga 7-7473
Agricultural Department 27 N Cayuga 7-7473
Garage 27 N Cayuga 7-7473
Cayuga Elementary Wheat Cayuga .... Auburn 2-8030
Cayuga Union Centr Cayuga ..... Auburn 2-6831

NYU Langone Health

Understand How Indexes Work

And so have non-clustered indexes

NYU Langone Health

GROUP BYs and ORDER BYs

- Avoid unnecessary GROUP BYs and ORDER BYs
- Only order or group data if necessary
  - Why waste resources doing it otherwise?

- Top Tip: Use the clustered index on columns used in GROUP BYs or ORDER BYs to improve performance

Useful resources

- http://www.w3schools.com/sql/

- http://www.sqlservercentral.com/

NYU Langone Health

EXERCISE

NYU Langone Health

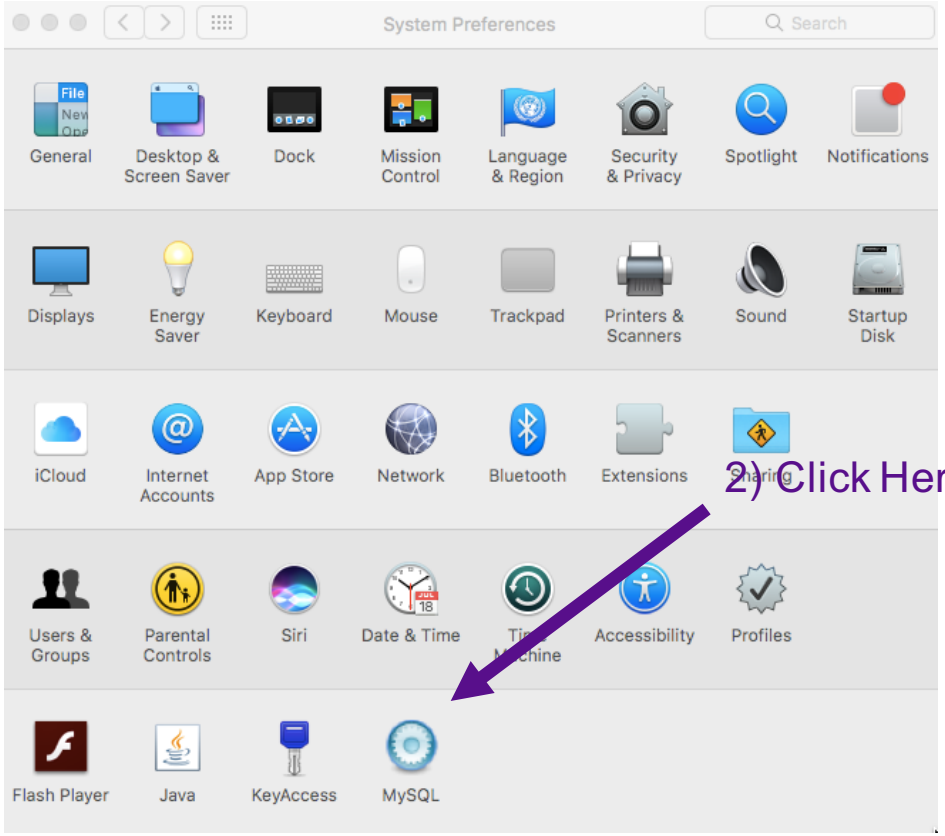
Exercise for next two lectures

- Question: **Dose the NHS's performance management regime for family practitioners (QOF) reduce Unplanned hospitalisation for chronic ambulatory care sensitive conditions?**
- Exercise designed to illustrate:
  - Loading data into MySQL
  - Managing and combing data in SQL
  - Pulling SQL data into R in order to undertake statistical analysis
- Data
  - Family Practioner (GP) level QOF points http://digital.nhs.uk/catalogue/PUB22266
    - QOF 2015-16: Prevalence, achievements and exceptions at regional and national level v2 [289.43KB]
  - Deprivation Scores for each CC https://www.gov.uk/government/statistics/english-indices-of-deprivation-2015
  - **Unplanned hospitalisation for chronic ambulatory care sensitive conditions**
  - https://data.england.nhs.uk/dataset/ccgois-2-6-unplanned-hospitalisation-for-chronic-ambulatory-care-sensitive-conditions
  - Original data in Excel but will provide extract data files to save time

NYU Langone Health

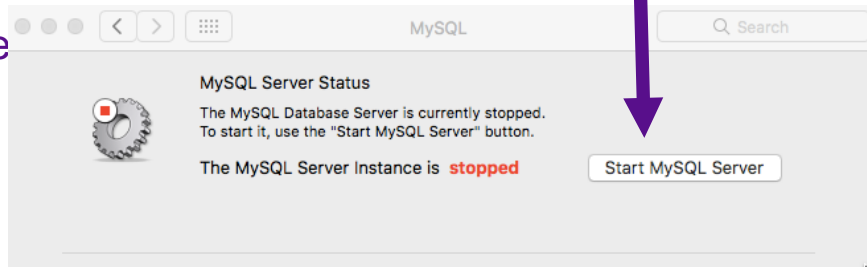
Start MySQL Server – on a Mac

- 1) Go into **System Preferences**



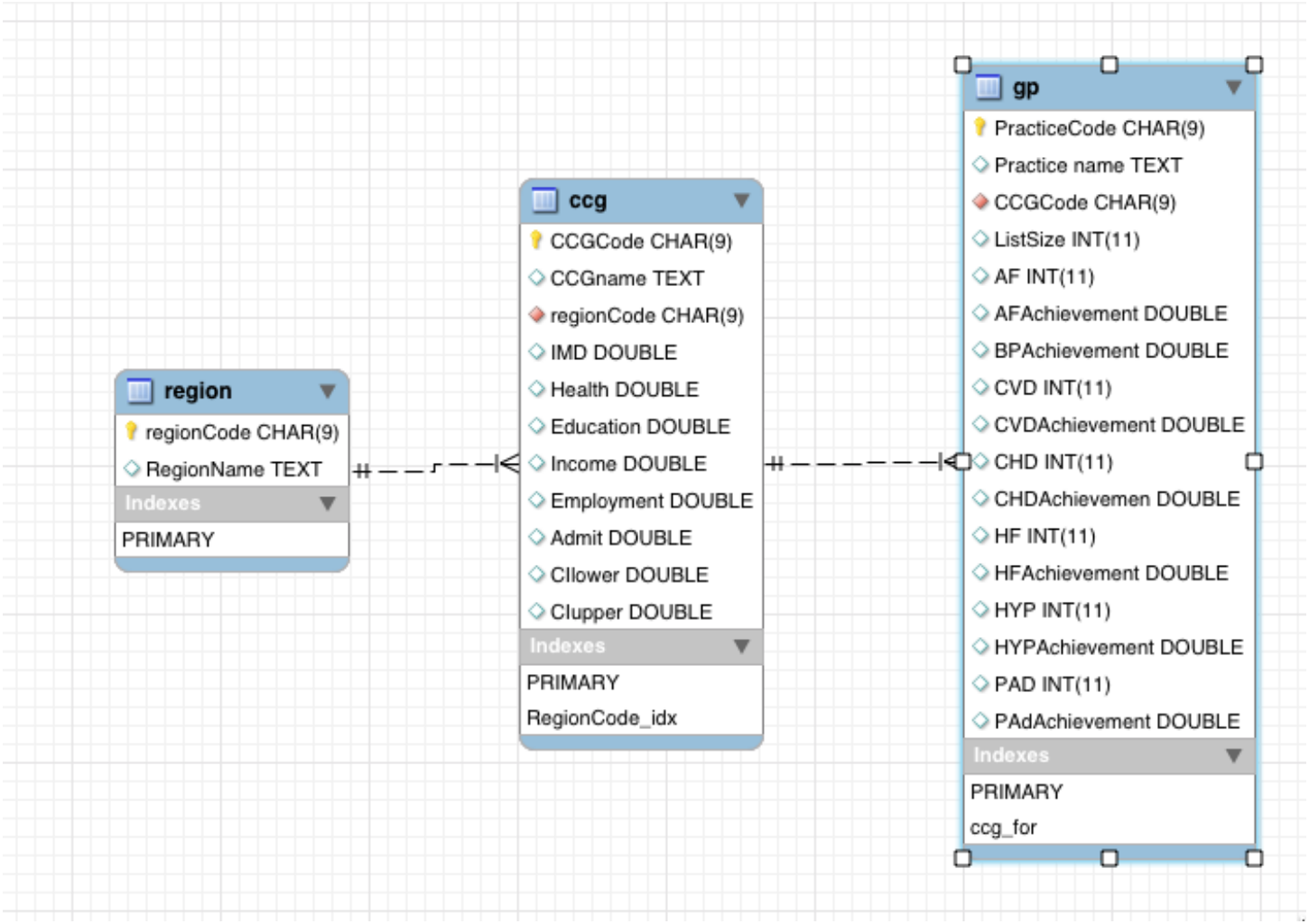
2) Click Here

3) Click Here



NYU Langone Health

Database ERR



NYU Langone Health

Open MySQL WorkBench

- Follow guidance at https://dev.mysql.com/doc/workbench/en/wb-admin-export-import-table.html to use the WorkBech to import data

THANK YOU

NYU Langone Health