Introduction to SQL and Databases

Peter Smyth
Cathie Marsh Institute

26th March 2020

UK Data Service

Overview of Seminar

- What is a Database?
- Different types of databases
- When the desktop isn't enough
- Introducing SQL
- Our database interface
- Other ways of interfacing to a database

What is a Database?

You can call any collection of data a database.

But somethings generally aren't!

UK Data Service

What is a Database

- The data in the database has to be Structured in some way so as to facilitate its retrieval.
- In most circumstances you won't want all of the data.

Different types of databases

- All databases store data
- But not all databases store data in the same way
- Sometimes we don’t care how the data is stored
- But sometimes it can effect the
  - Efficiency of our queries
  - And how we write queries

Different types of Databases

- SQL or Relational Databases
  - Consists of tables which are related to each other in a defined way using one or more ‘key’ fields
  - Uses SQL queries to retrieve the data
- NoSQL SQL Databases
  - Data stored as Collections of Documents.
  - Documents can have complex structures.
  - Different documents in the same collection can have different structures
  - Generally do not use SQL queries to access the data

SQL Table = NoSQL Collection

SQL Table row = NoSQL Document

Tables

- Data in relational databases are arranged in tables.
- You can think of these very much like the format you might see in a spreadsheet.

- Rows and columns of data, with each column having its own column-name.
- Internally a database table is not quite the same as a spreadsheet, but as an image of what a table is like, it is perfect.

Tables

	A	B	C	D	E	F	G	H	I	J
1	agegpt	aprxsocgr	cofbt	econac1t	ethew1t	famcomp	health1	indus1t	marstat1t	numhrs
2	6	4	1	5	1	2	2	2	2	-9
3	4	3	1	1	1	5	1	6	1	4
4	4	4	1	1	1	3	1	11	1	3
5	2	2	1	1	1	3	2	7	1	3
6	5	2	1	1	1	3	1	4	4	3
7	6	3	1	1	1	2	2	2	2	3
8	4	3	1	1	1	5	2	11	3	2
9	1	-9	1	-9	1	3	2	-9	1	-9
10	7	2	1	5	1	2	1	2	2	-9
11	6	4	1	5	1	1	3	5	4	-9
12	3	4	1	1	1	1	3	2	1	3
13	1	-9	1	-9	1	2	1	-9	1	-9
14	2	2	1	1	1	1	1	2	1	3
15	2	2	1	4	1	1	2	-9	1	-9
16	2	4	1	1	1	5	2	4	1	2
17	2	2	1	4	1	1	2	6	1	-9

censusmicroteaching11ew

Why Relational Database?

- Although it is possible to have a database with only one table, in general a single database will be home to many tables, large and small.
- It is possible that these tables are totally independent of each other but in practice it is unlikely to be the case.

A paper sales receipt

No. **0001**

FREE YOUR LOGO

COMPANY NAME
1st Main Street
Sunny, FL 33436
123-456-7789

CUSTOMER ORDER NO.					DATE	
NAME						
ADDRESS						
CITY, STATE, ZIP						
SOLD BY	CASH	C.O.D.	CHARGE	ON ACCT.	MDSE. RTD	PAID OUT

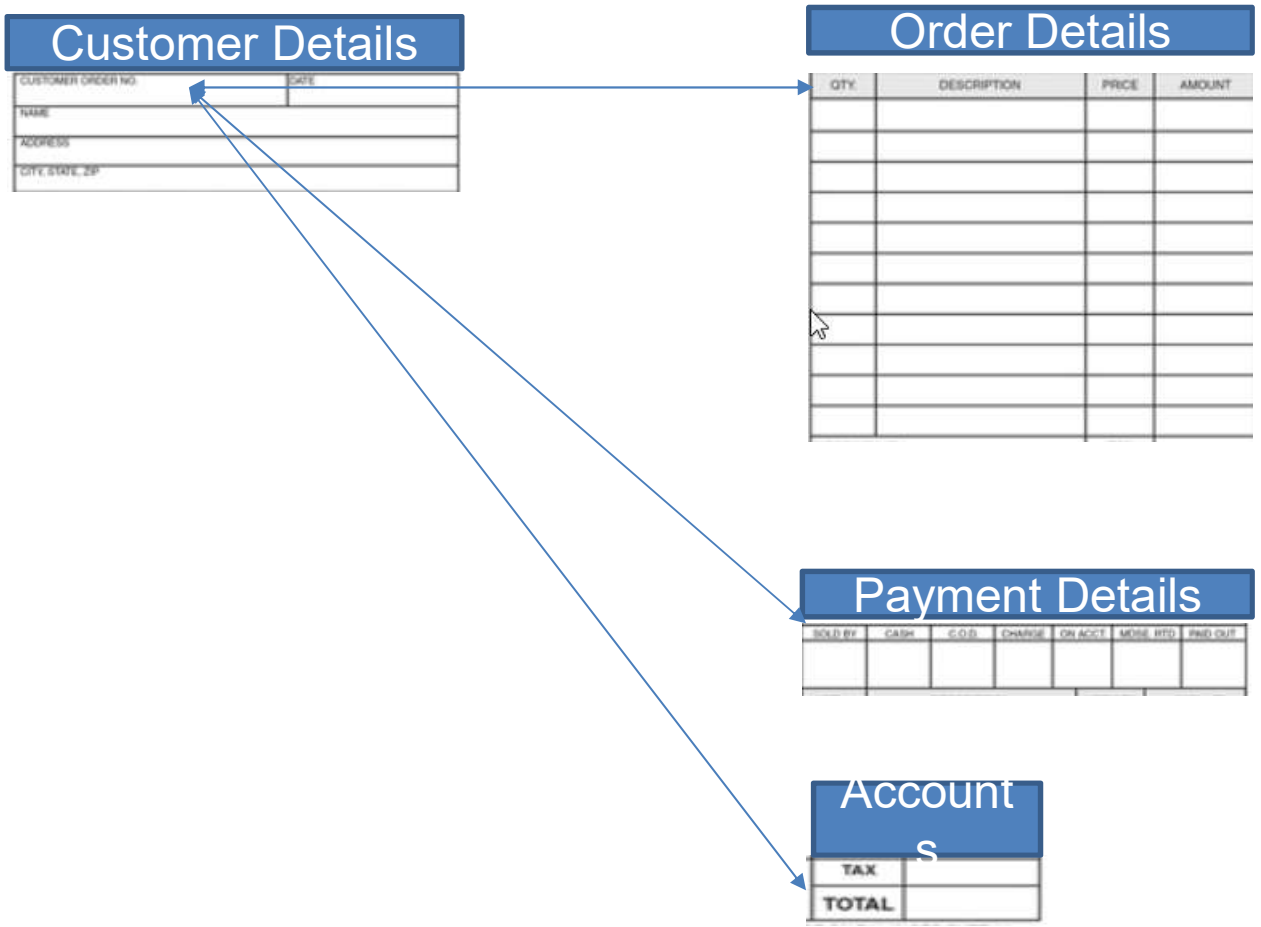
QTY	DESCRIPTION	PRICE	AMOUNT
RECEIVED BY:		TAX	
		TOTAL	

TERMS: NET 30 DAYS. 1.5% SERVICE CHARGE ON BALANCES OVER 30

Thank You

UK Data Service

Broken down to tables



Using multiple tables

Advantages

- Smaller amounts of data to manipulate
- Security of the data

Disadvantages

- Slower to update (i.e. add new sales receipts)
- More complex queries if data needed from multiple tables

When the Desktop isn't enough

- Databases can be very large – GBs, Millions of records
- Too big to store on your desktop
- May not be yours anyway
- You may only be interested in part of it
  - Some tables
  - Parts of some tables

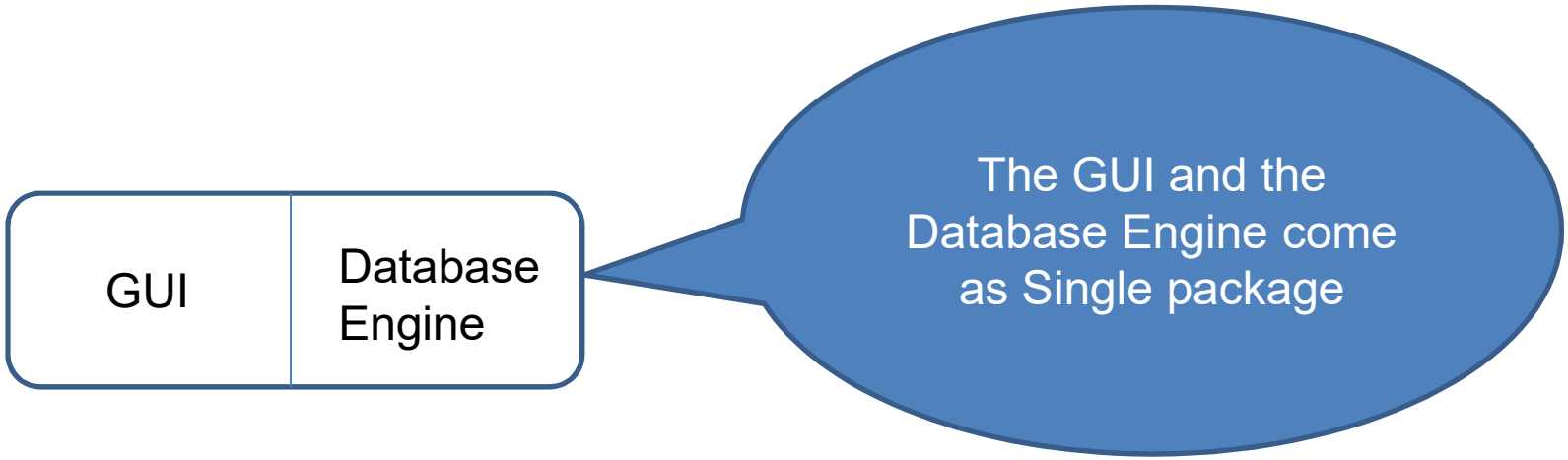
- Fortunately Database systems make it easy to share.

Database environments

- A Database ‘System’ is not a single program (like Excel)
- There are two main parts:
- The Database Engine
  - This defines the database type
  - It manages the storage of the data
  - It interprets and acts on SQL requests
  - It returns the data from SQL queries
- The User interface
  - How we communicate with the Database engine
  - Can take on many forms
  - All Database systems come with their preferred user interface
    - Usually some kind of GUI (Graphical User interface)

Database environments

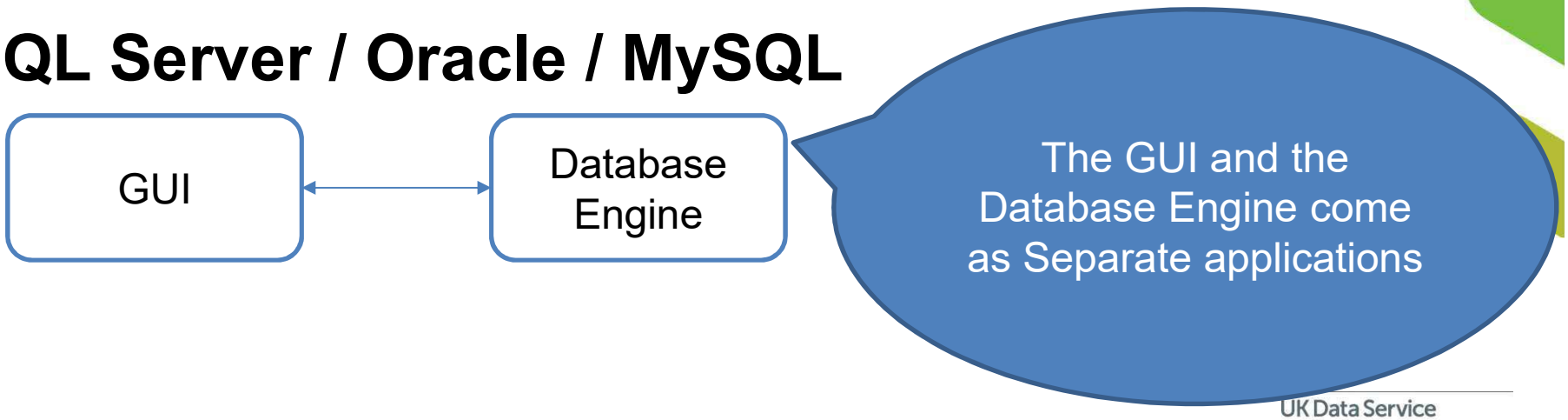
MS Access



Database environments

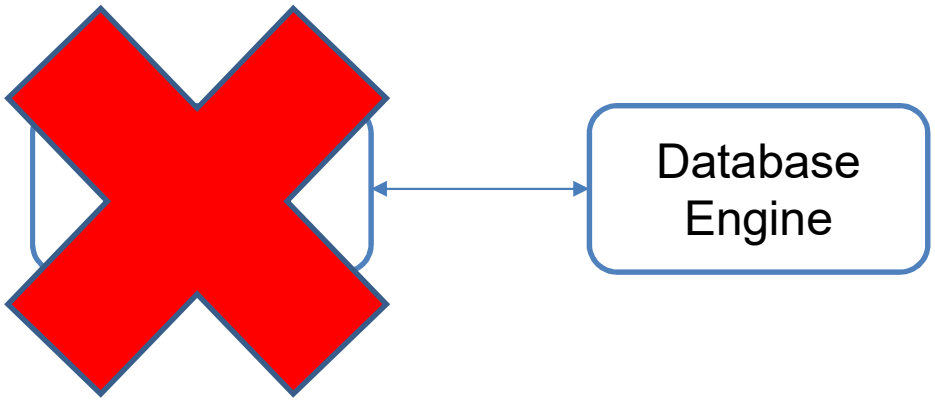
MS Access

SQL Server / Oracle / MySQL



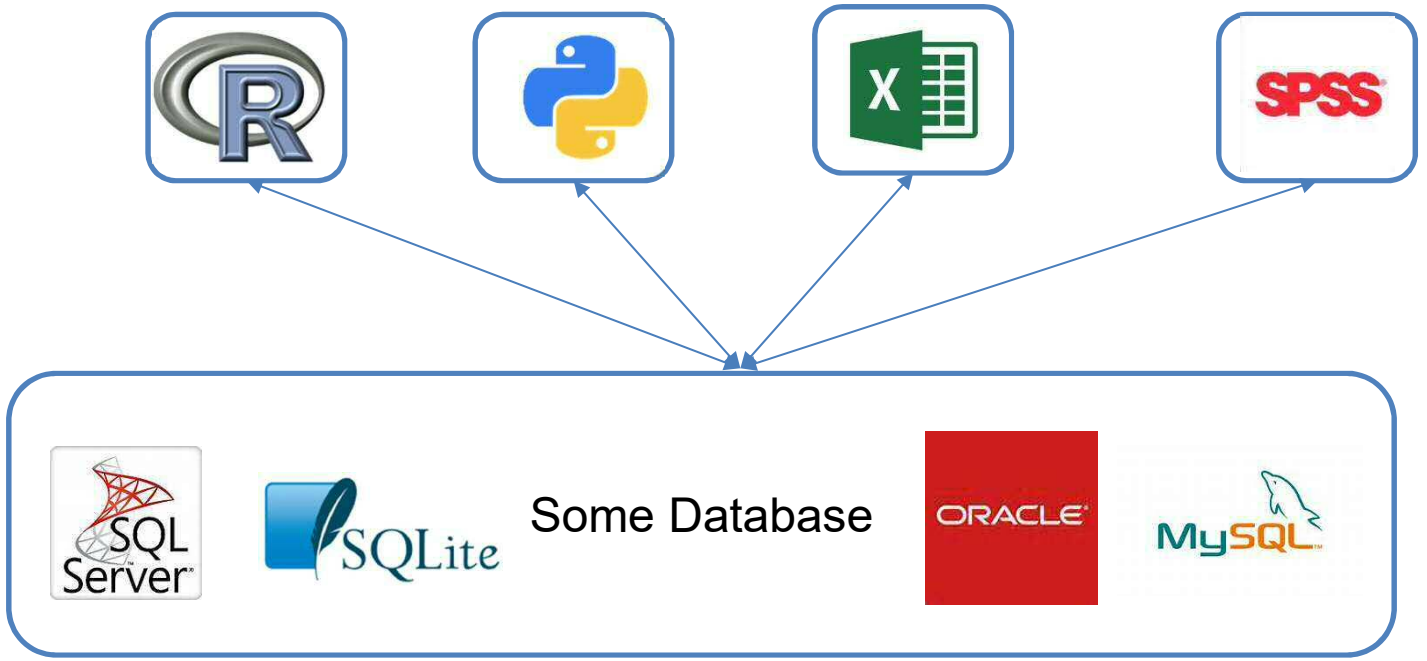
Database environments

You can install the database engine, without installing the GUI



You don’t need a GUI, anything that can connect to the database engine will do

Database environments



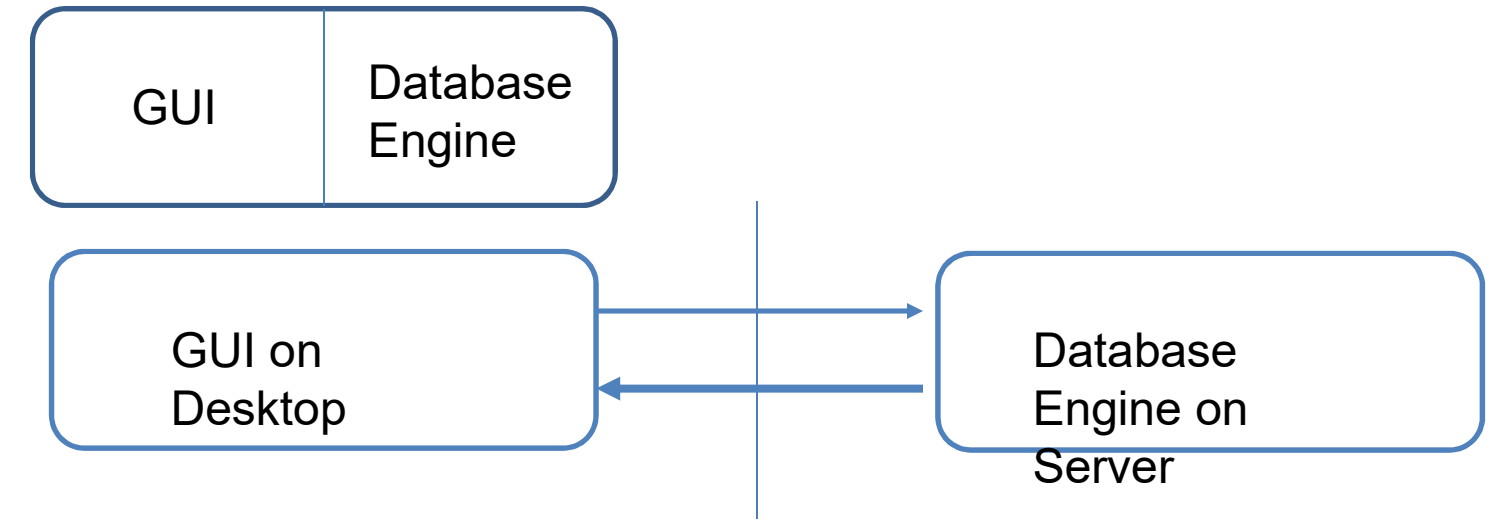
Standalone GUI's

- MySQL Workbench
- DB Browser for SQLite
- Dbeaver
- Sophisticated Text editors (VS Code, Atom etc)

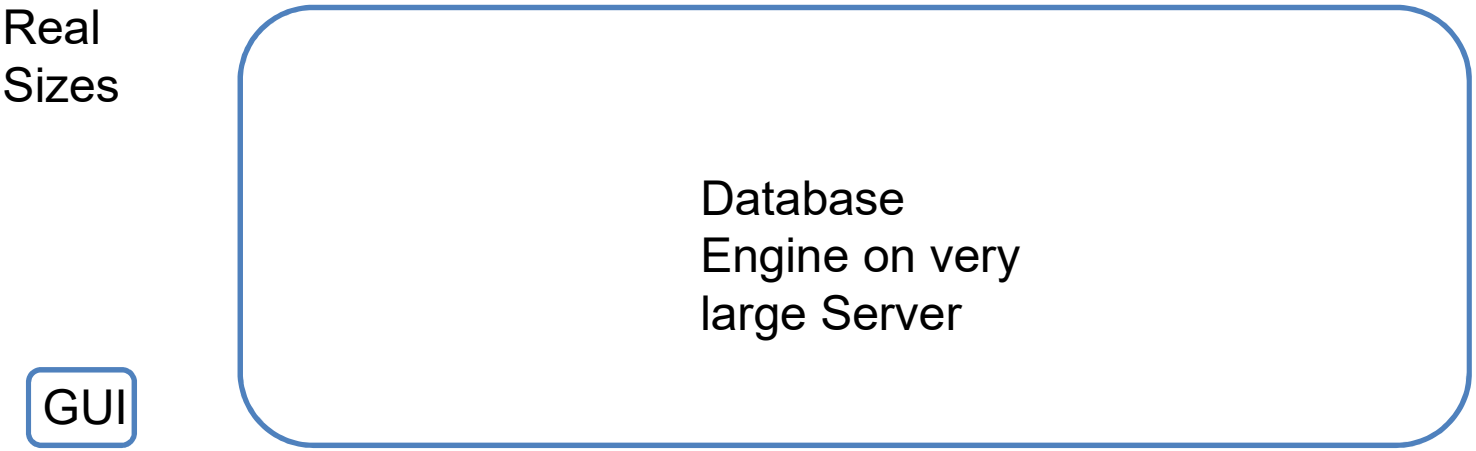
When the desktop isn’t enough

- The Database engine and the User access programs are separate entities
- They don’t have to be on the same machine
- The GUI could be on your desktop/laptop
- The database Engine could be on a far larger machine, it may have 100’s of GB of memory

When the desktop isn't enough



Real Sizes



Introducing SQL

- There are two basic types of SQL commands
- DDL statements and DML statements
- DDL – Data Definition Language statements are used to Create databases and Tables and generally manage the environment
- DML – Data Manipulation Language statement are used to deal with the actual data in the tables.

Introduction to SQL

- Only really interested in DML statements in this Webinar
- And really only one of these.
- DML statements include;
  - Create (Insert)
  - Read (Select)
  - Update and
  - Delete
- Collectively referred to as `CRUD` statements

Introduction to SQL

- We are going to assume that our data is already in table so all we want to do is to Read (or Select) that data in different ways
- So all of our focus from now is on the SELECT statement
- In reality much of your time spent on SQL is in coding SELECT queries.
- SELECT Statements are generally referred to as SQL queries.

The SELECT statement

What can we do in a SELECT query?

1. Select specific columns from a table (or tables)
2. Select specific rows from a table (or tables)
3. Add new columns to the **output** of a query
   - Based on existing columns
   - Based on a variety of builtin function
4. Create aggregations from the rows in a table
   - By grouping values in one or more columns
5. Produce sorted output based on one or more columns

The Select Query

- Need to know;
- Nothing in a SELECT statement can affect the underlying data in the tables
- The output of a SELECT is always a table
  - Even if there is only one row and one column
- There are various ways of saving the output from a SELECT query
  - As another Table or View
  - Write it to an output (typically csv) file

Select Statement Demo

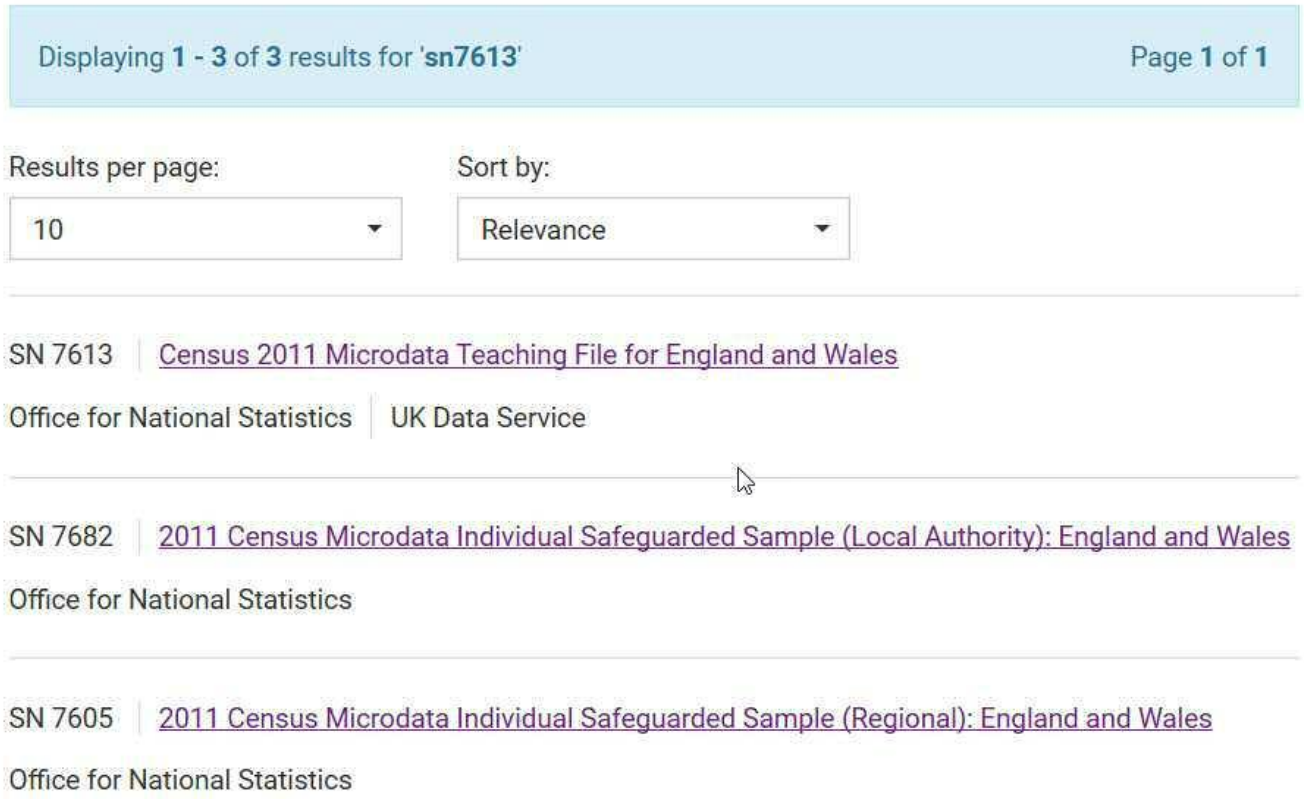
The rest of this Webinar is devoted to demonstrations.

We will look at a GUI (DB Browser for SQLite)

We will create and run some SELECT queries

The dataset being used

- For the demonstrations we are going to use a dataset from the UKDS



UK Data Service

The Demos

- Using DB Browser for SQLite
- Using MS Query from Excel

Access from Python Code

```
import sqlite3
import pandas as pd
```

```
con = sqlite3.connect('./SN7613/SN7613.db')
```

```
sql = "SELECT * FROM Occupations"

cur = con.cursor()
cur.execute(sql)
rows = cur.fetchall()
for row in rows:
    print(row)
```

```
(1, 'Managers, Directors and Senior Officials')
(2, 'Professional Occupations')
(3, 'Associate Professional and Technical Occupations')
(4, 'Administrative and Secretarial Occupations')
(5, 'Skilled Trades Occupations')
(6, 'Caring, Leisure and Other Service Occupations')
(7, 'Sales and Customer Service Occupations')
(8, 'Process, Plant and Machine Operatives')
(9, 'Elementary Occupations')
(-9, 'No code required (under 16/never worked/student/pupil living away)')
```

Questions

Peter Smyth

Peter.smyth@manchester.ac.uk

ukdataservice.ac.uk/help/

Subscribe to the UK Data Service news list at https://www.jiscmail.ac.uk/cgi-bin/webadmin?A0=UKDATASERVICE

Follow us on Twitter https://twitter.com/UKDataService or Facebook https://www.facebook.com/UKDataService