Lab 5

Objectives

In this lab you will use the Select statement to retrieve data from two tables.

Join

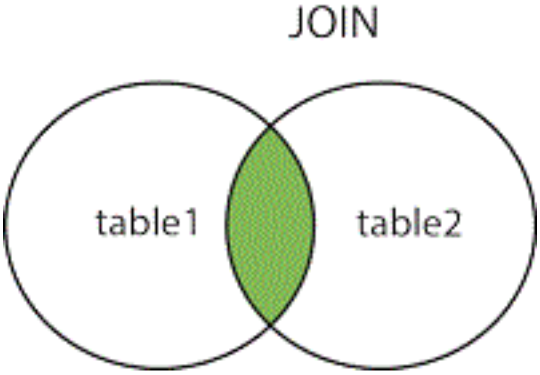
All the queries up until this point have been useful except for one major limitation - that is, we have been selecting from only one table at a time with the *SELECT* statement. It is time to introduce one of the most beneficial features of SQL and relational database systems - the **Join**. To put it simply, the **Join** makes relational database systems “relational”.

Joins allow you to link data from two or more tables together into a single query result - from one single SELECT statement.

A database is defined as a set of related data stored in tables of rows and columns. Most often the data we are looking for will be in more than one table. So, for example if we wanted to see the book details for a given book copy. We would want information from the tables: book and bookcopy.

The **JOIN** keyword selects all rows from both tables if there is a match between the *join* columns in both tables (i.e. Primary Key - Foreign Key link).

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



The JOIN statement

Recall the attributes for the book and bookcopy tables:

Book(ISBN, title, publisher, publishedDate, category, price)

BookCopy(copyId, ISBN, dateAcquired, dateDestroyed)

Notice how each of the tables have a common **ISBN** column. This column, will be used to JOIN the two tables. It is the primary key in the *Book* table and the foreign key in the *BookCopy* table.

If we wish to return the copyId and title for each bookcopy, we need values from both tables so we perform the statement as follows:

```
SELECT copyId, title
FROM book join bookcopy
ON book.isbn=bookcopy.isbn;
```

This particular **Join** is also known as an **Inner Join** or **Equijoin**. This is the most common type of **Join** that you will see or use.

Note: The *JOIN* keyword selects all rows from both tables as long as there is a match between the columns. If there are rows in the *Book* table that do not have matches in the *BookCopy* table, these Books (rows) will NOT be listed.

The following statements are NOT the same - run them both and see the different results. Count the number of records returned.

1. Without the **ON** clause:

```
SELECT copyId, title
FROM book JOIN bookcopy;
```

2. With the **ON** clause:

```
SELECT copyId, title
FROM book JOIN bookcopy
ON book.isbn=bookcopy.isbn;
```

Another alternative of writing the query is to use the **NATURAL JOIN**. In this case, the **join** columns must have the same name. The statement omits the **ON** clause.

```
SELECT copyId, title
FROM book NATURAL JOIN bookcopy;
```

Example One

We want to return the title and copyId of all books containing the word *JavaScript* in the title.
There are four things to do here:

1. Identify which tables have the data you are looking for, in this case it is *Book* and *BookCopy*. These need to be joined.
2. Identify which columns are the primary and foreign keys in these tables; in this case it is *ISBN* in the *Book* and *ISBN* in the *BookCopy* table. We use these in the *ON* clause.

3. Identify which columns we want the query to output; in this case it is *copyId* from the *BookCopy* table; and *title* from the *Book* table. We will use these in our *SELECT*.
4. Add any conditions necessary, in this example: *title like '%JavaScript%'*;

```
SELECT title, copyId
FROM book JOIN bookcopy
ON book.isbn=bookcopy.isbn
WHERE title LIKE '%JavaScript%';
```

Example Two

In this example, we wish to return the copyId, title, publisher, and acquired date for all book copies that are Computing books.

Again, there are four things to do here:

1. Identify which tables have the data you are looking for, in this case its *Book* and *BookCopy*. These need to be joined.

2. Identify which columns are the primary and foreign keys in these tables; in this case it is *ISBN* in the *Book* and *ISBN* in the *BookCopy* table. We use these in the *ON* clause.

3. Identify which columns we want the query to output; in this case it is *copyId* and *acquiredDate* from the *BookCopy* table;
and *title* and *publisher* from the *Book* table. We will use these in our *SELECT*.

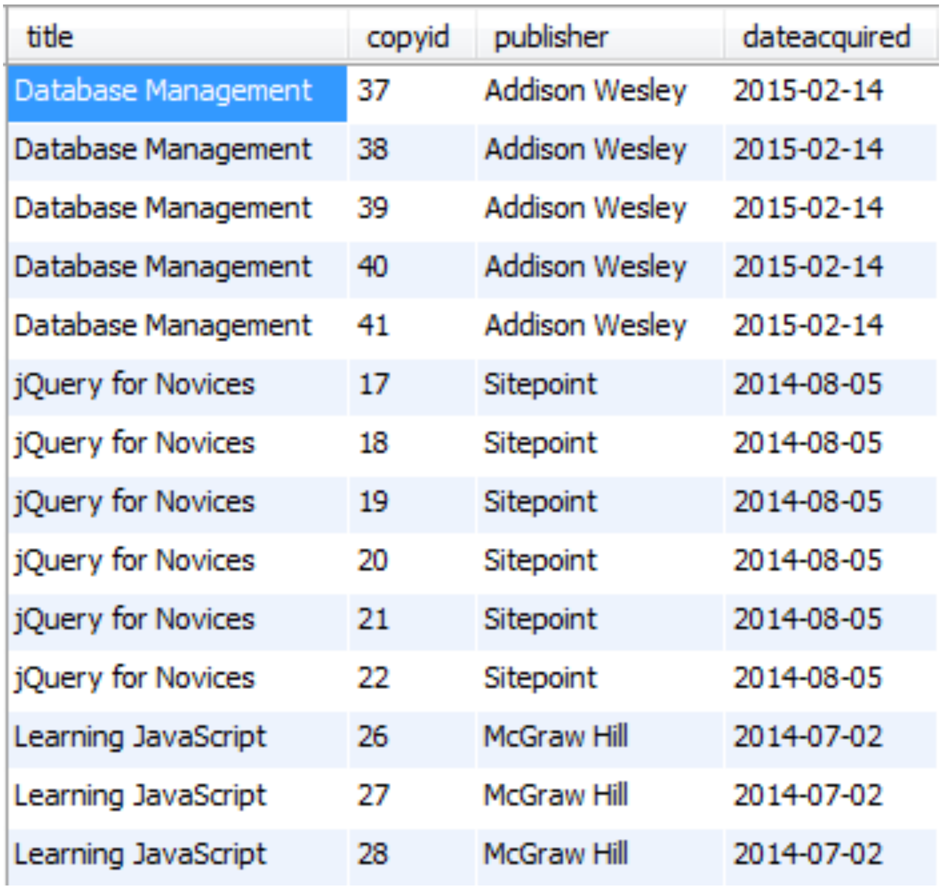
4. Add any conditions necessary, in this example: *category = 'Computing'*;

```
SELECT copyid, title, publisher, dateacquired
FROM book JOIN bookcopy
ON book.isbn = bookcopy.isbn
WHERE category = 'Computing';
```

Join Exercises

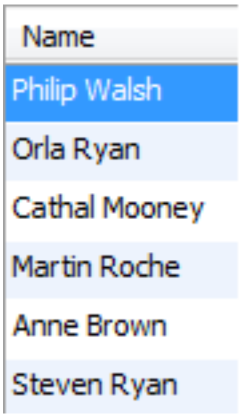
1. Retrieve the title, copyId, publisher, and acquired date for all book copies that were acquired since June 1 2014.

   Sort the results in alphabetical order of title.



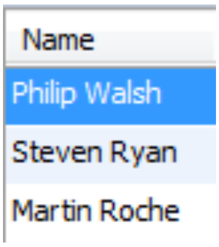
title	copyid	publisher	dateacquired
Database Management	37	Addison Wesley	2015-02-14
Database Management	38	Addison Wesley	2015-02-14
Database Management	39	Addison Wesley	2015-02-14
Database Management	40	Addison Wesley	2015-02-14
Database Management	41	Addison Wesley	2015-02-14
jQuery for Novices	17	Sitepoint	2014-08-05
jQuery for Novices	18	Sitepoint	2014-08-05
jQuery for Novices	19	Sitepoint	2014-08-05
jQuery for Novices	20	Sitepoint	2014-08-05
jQuery for Novices	21	Sitepoint	2014-08-05
jQuery for Novices	22	Sitepoint	2014-08-05
Learning JavaScript	26	McGraw Hill	2014-07-02
Learning JavaScript	27	McGraw Hill	2014-07-02
Learning JavaScript	28	McGraw Hill	2014-07-02

2. List the students (combined fname and lname) who have borrowed a book(s). Return the student name only once even if he/she has more than one loan.



Name
Philip Walsh
Orla Ryan
Cathal Mooney
Martin Roche
Anne Brown
Steven Ryan

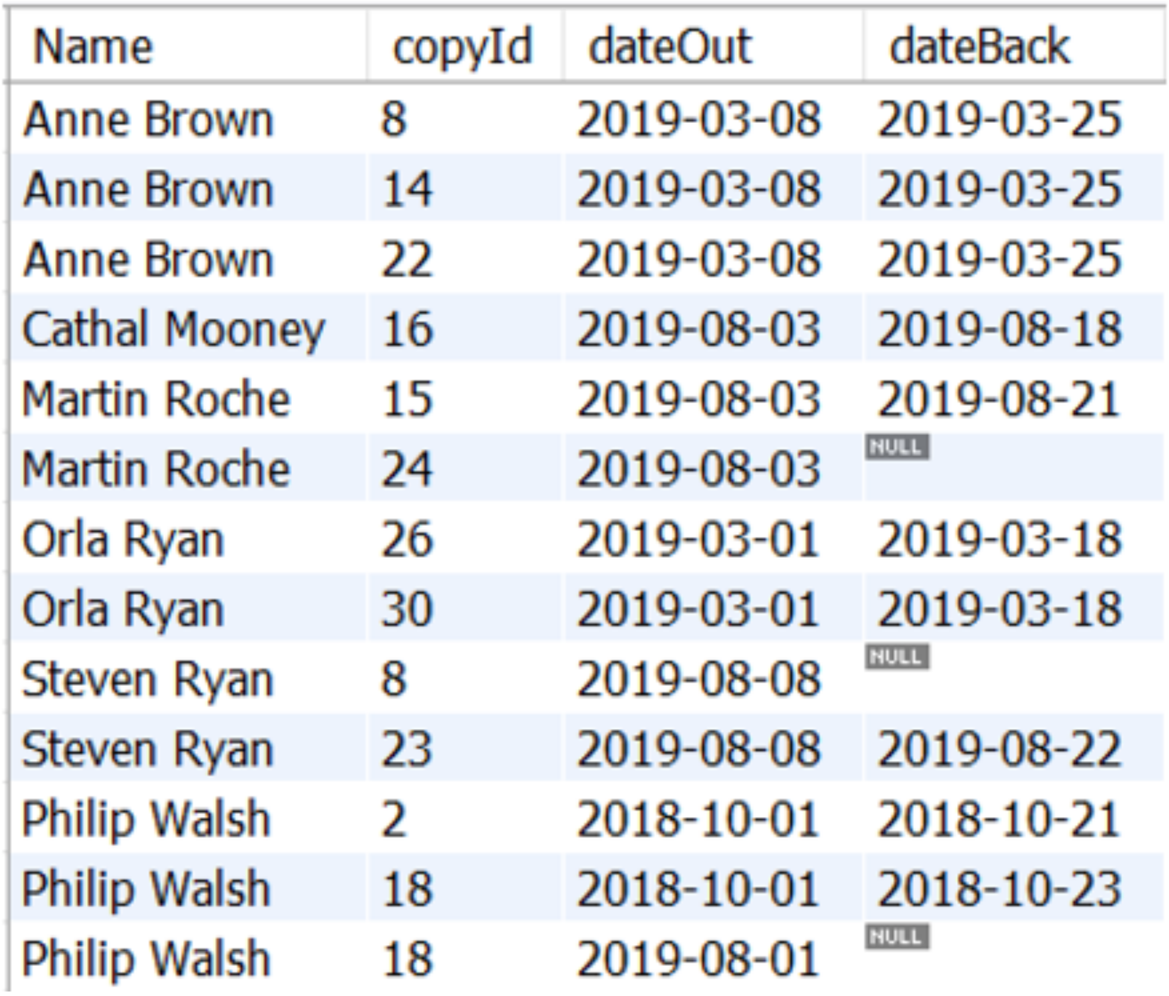
3. List the students (combined fname and lname) who have borrowed a book(s) and has it still on loan (dateBack is empty). Return the student name only once even if he/she has more than one loan.



Name
Philip Walsh
Steven Ryan
Martin Roche

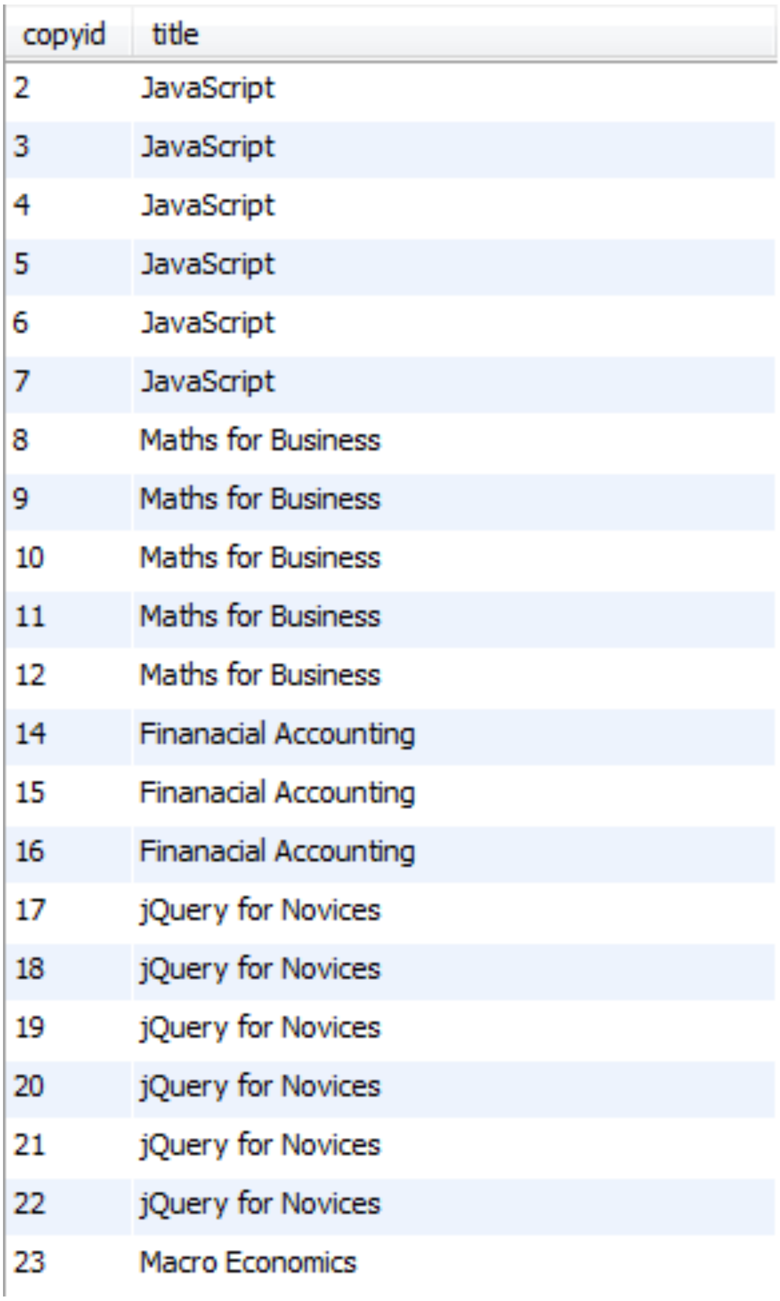
4. For each loan record, retrieve the student by Name (combined fname and lname) who made the loan, and the copyId, dateOut and dateBack for each book loaned.

   Sort the results in alphabetical order of Name (fname sorted within lname).

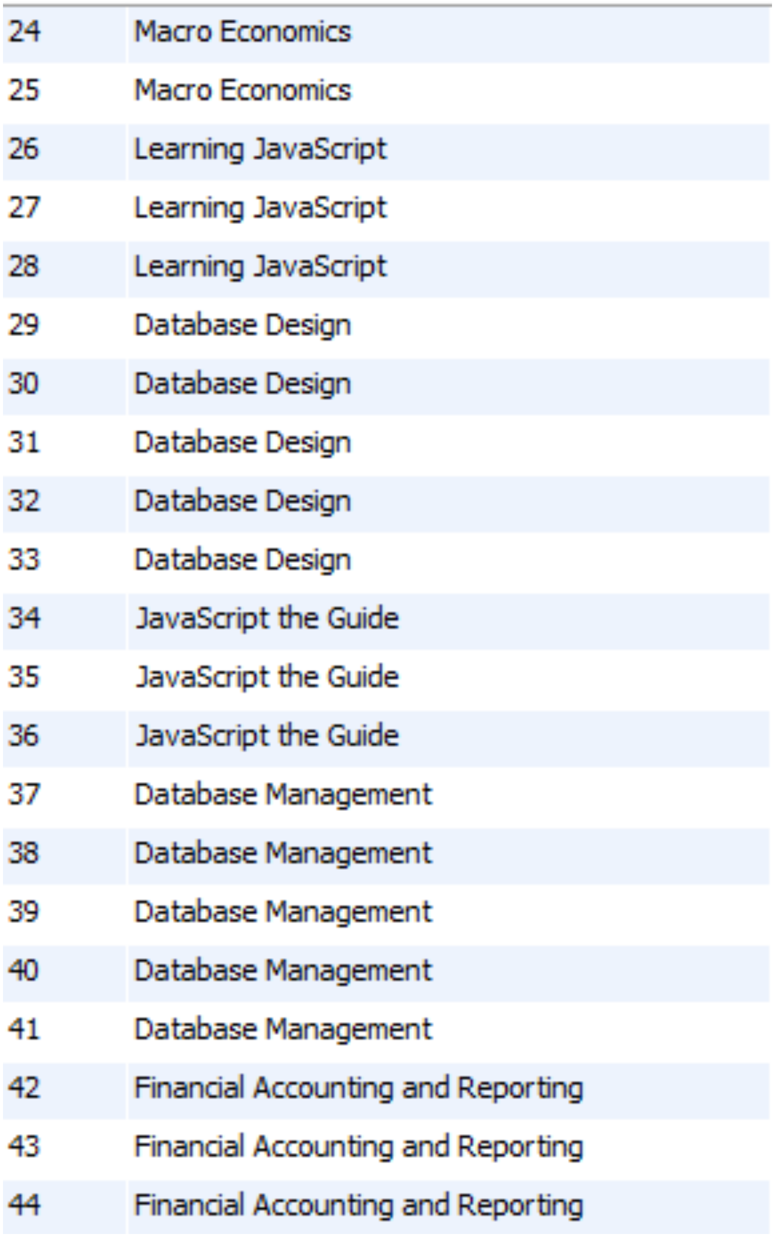


Name	copyId	dateOut	dateBack
Anne Brown	8	2019-03-08	2019-03-25
Anne Brown	14	2019-03-08	2019-03-25
Anne Brown	22	2019-03-08	2019-03-25
Cathal Mooney	16	2019-08-03	2019-08-18
Martin Roche	15	2019-08-03	2019-08-21
Martin Roche	24	2019-08-03	NULL
Orla Ryan	26	2019-03-01	2019-03-18
Orla Ryan	30	2019-03-01	2019-03-18
Steven Ryan	8	2019-08-08	NULL
Steven Ryan	23	2019-08-08	2019-08-22
Philip Walsh	2	2018-10-01	2018-10-21
Philip Walsh	18	2018-10-01	2018-10-23
Philip Walsh	18	2019-08-01	NULL

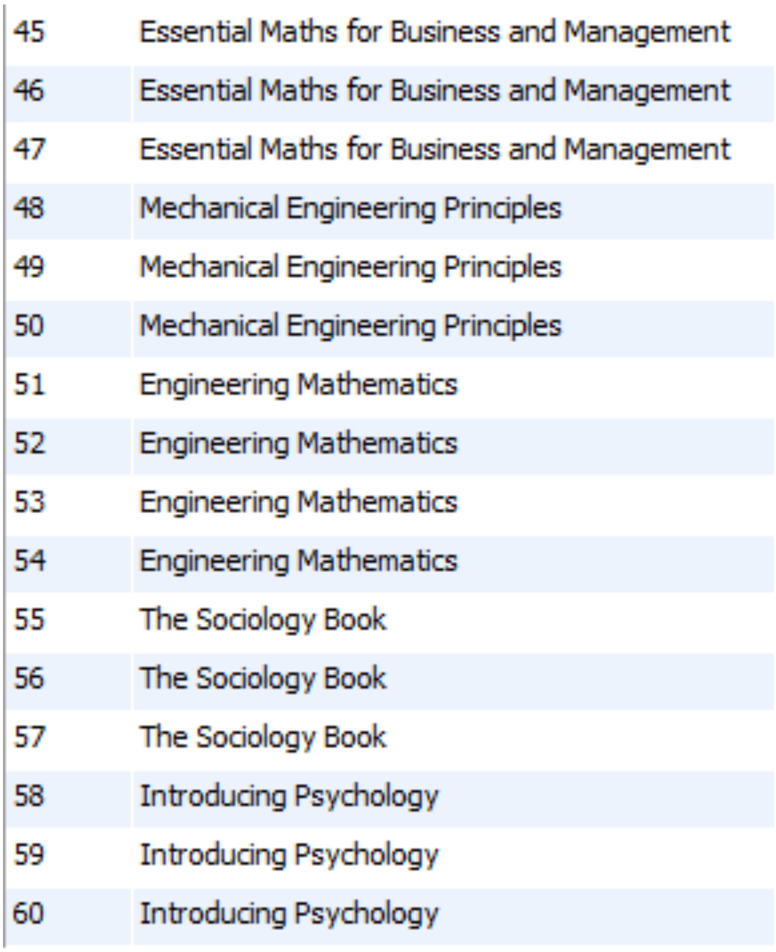
5. List the copyId and title for all books that are in service now (datedestroyed is empty).



copyid	title
2	JavaScript
3	JavaScript
4	JavaScript
5	JavaScript
6	JavaScript
7	JavaScript
8	Maths for Business
9	Maths for Business
10	Maths for Business
11	Maths for Business
12	Maths for Business
14	Finanacial Accounting
15	Finanacial Accounting
16	Finanacial Accounting
17	jQuery for Novices
18	jQuery for Novices
19	jQuery for Novices
20	jQuery for Novices
21	jQuery for Novices
22	jQuery for Novices
23	Macro Economics

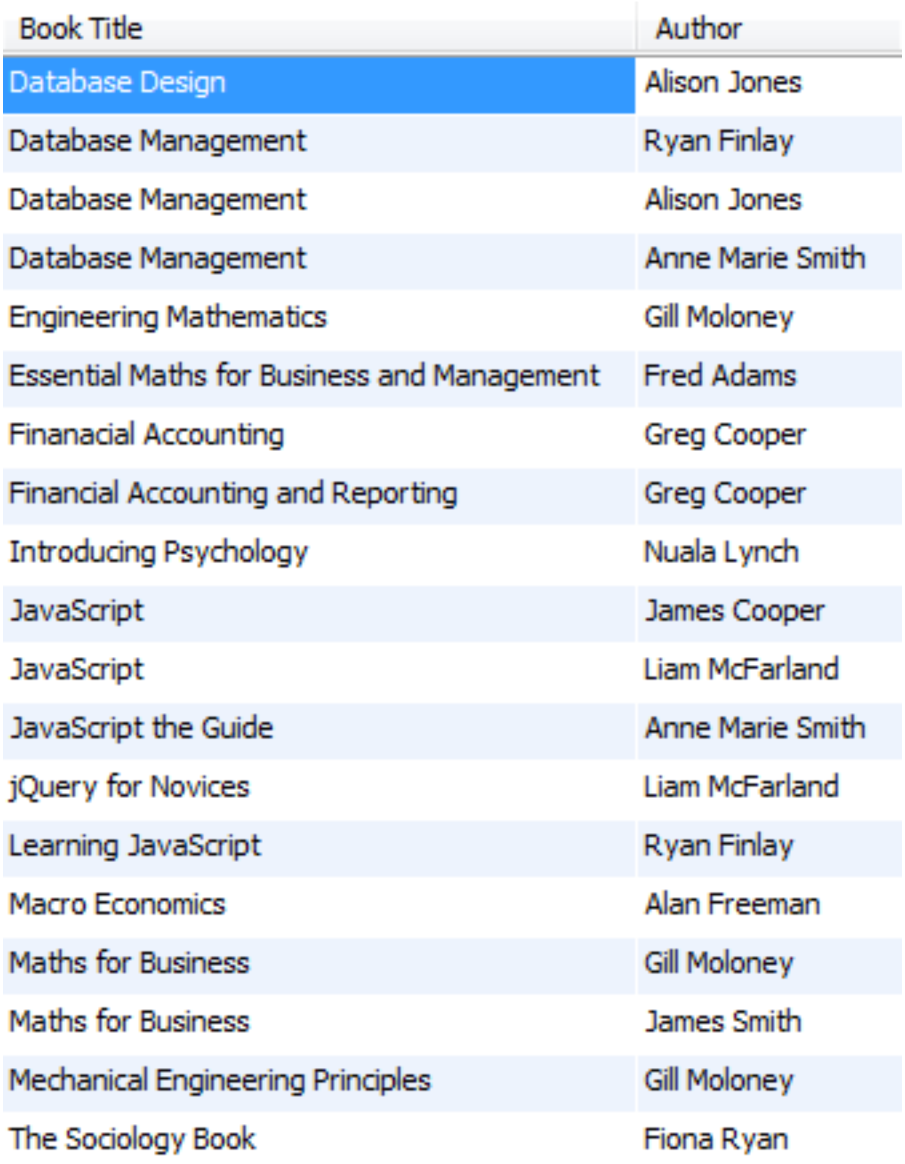


24	Macro Economics
25	Macro Economics
26	Learning JavaScript
27	Learning JavaScript
28	Learning JavaScript
29	Database Design
30	Database Design
31	Database Design
32	Database Design
33	Database Design
34	JavaScript the Guide
35	JavaScript the Guide
36	JavaScript the Guide
37	Database Management
38	Database Management
39	Database Management
40	Database Management
41	Database Management
42	Financial Accounting and Reporting
43	Financial Accounting and Reporting
44	Financial Accounting and Reporting



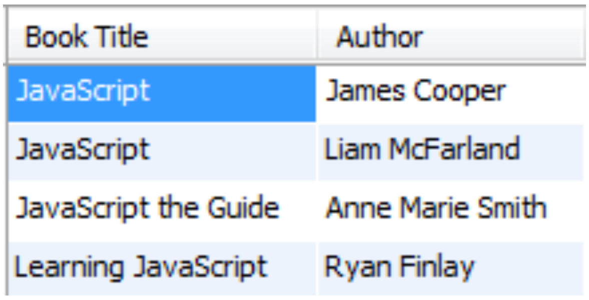
45	Essential Maths for Business and Management
46	Essential Maths for Business and Management
47	Essential Maths for Business and Management
48	Mechanical Engineering Principles
49	Mechanical Engineering Principles
50	Mechanical Engineering Principles
51	Engineering Mathematics
52	Engineering Mathematics
53	Engineering Mathematics
54	Engineering Mathematics
55	The Sociology Book
56	The Sociology Book
57	The Sociology Book
58	Introducing Psychology
59	Introducing Psychology
60	Introducing Psychology

6. List the titles and authors by name for all books. Label the title *Book Title* and label the author name *Author*. Sort in alphabetical order of title.



Book Title	Author
Database Design	Alison Jones
Database Management	Ryan Finlay
Database Management	Alison Jones
Database Management	Anne Marie Smith
Engineering Mathematics	Gill Moloney
Essential Maths for Business and Management	Fred Adams
Finanacial Accounting	Greg Cooper
Financial Accounting and Reporting	Greg Cooper
Introducing Psychology	Nuala Lynch
JavaScript	James Cooper
JavaScript	Liam McFarland
JavaScript the Guide	Anne Marie Smith
jQuery for Novices	Liam McFarland
Learning JavaScript	Ryan Finlay
Macro Economics	Alan Freeman
Maths for Business	Gill Moloney
Maths for Business	James Smith
Mechanical Engineering Principles	Gill Moloney
The Sociology Book	Fiona Ryan

7. List the titles and authors by name for all books that have JavaScript in the title. Label the title *Book Title* and label the author name *Author*. Sort in alphabetical order of title.



Book Title	Author
JavaScript	James Cooper
JavaScript	Liam McFarland
JavaScript the Guide	Anne Marie Smith
Learning JavaScript	Ryan Finlay

Example Three

In this example, we wish to return the number of book copies per Book title and output the count (result) with the label *Number of Books*.

There are three things to do here:

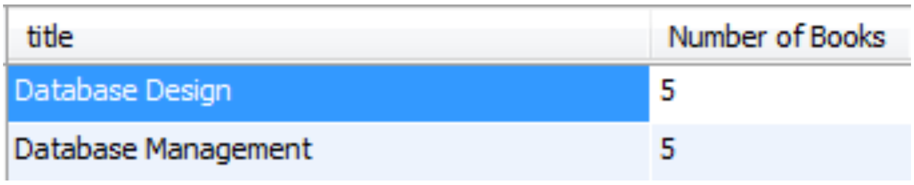
1. Identify which tables have the data you are looking for, in this case its *Book* and *BookCopy*. These need to be joined.
2. Identify which columns are the primary and foreign keys in these tables; in this case it is *ISBN* in the *Book* and *ISBN* in the *BookCopy* table. We use these in the *ON* clause.
3. Identify which columns we want the query to output; in this case it is *title* from the *Book* table and *count(copyId)* from the *ookCopy* table. We will use these in our *SELECT*.

Let us put it all together:

```
SELECT title, count(copyId) AS "Number of Books"
FROM book JOIN bookcopy
ON book.isbn = bookcopy.isbn
GROUP BY title;
```

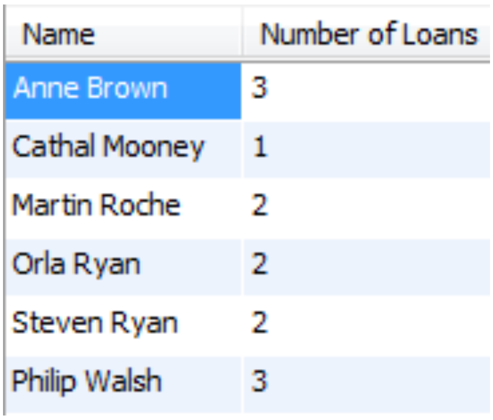
More Join Exercises

1. Return the book title and corresponding number of book copies for all books that have Database in the book title. Output the count with the label Number of Books.



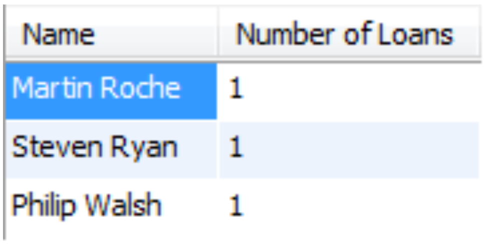
title	Number of Books
Database Design	5
Database Management	5

2. Return the number of loans per student (identified by name). Output the count with the label Number of Loans and identify each student by name (combined fname and lname). Note: Use the combined fname and lname in the Group By clause. Sort in alphabetical order by Last Name and then First Name.



Name	Number of Loans
Anne Brown	3
Cathal Mooney	1
Martin Roche	2
Orla Ryan	2
Steven Ryan	2
Philip Walsh	3

3. Return the number of loans that a student has at present (where dateBack is empty). Output the count with the label Number of Loans and again identify each student by name.



Name	Number of Loans
Martin Roche	1
Steven Ryan	1
Philip Walsh	1

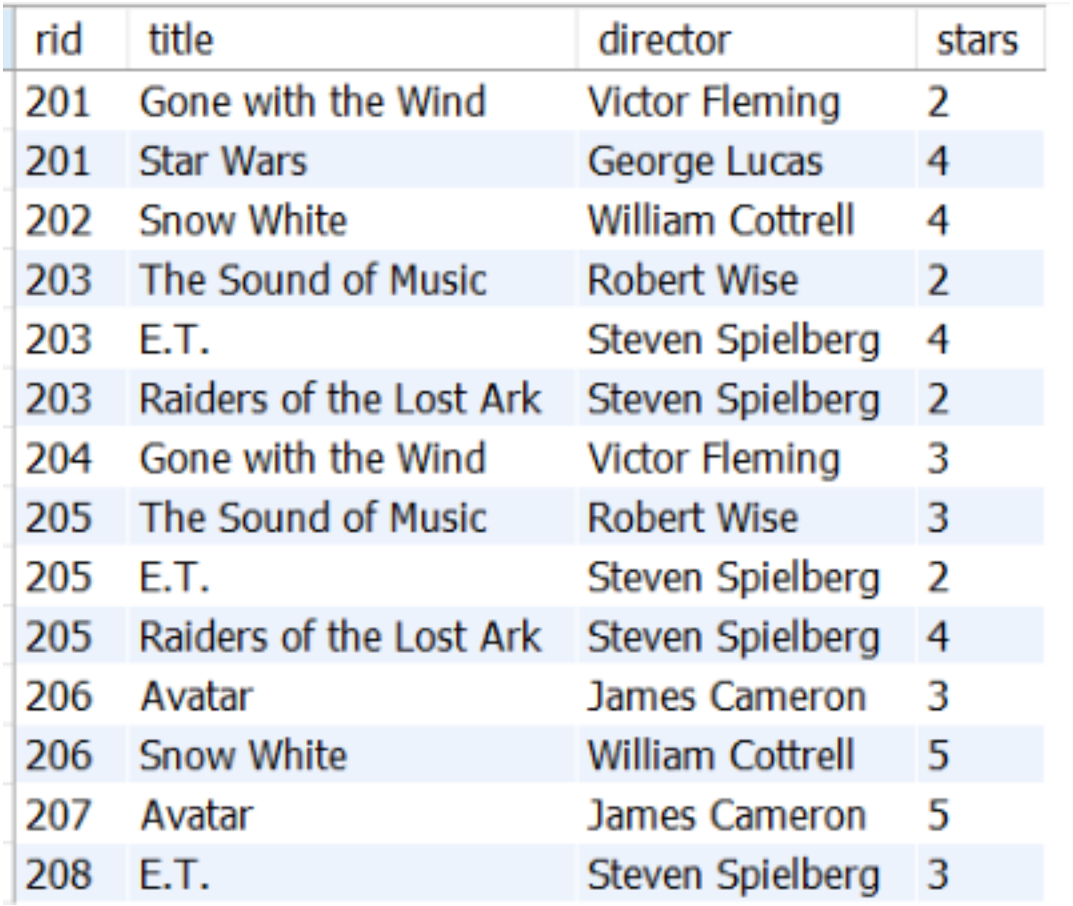
Movies Database Exercise

For these exercises, load the *movies* database and remember to enter the command:

```
USE MOVIES;
```

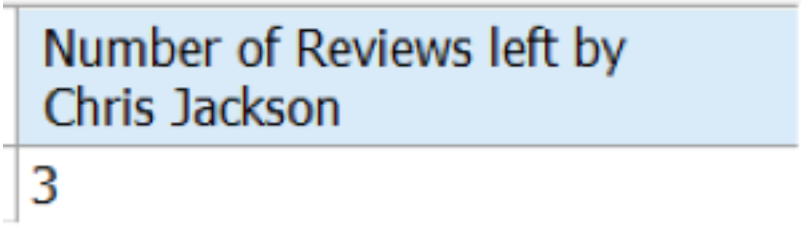
To do:

1. For each rating, return the reviewer id, film title, director, and number of stars.



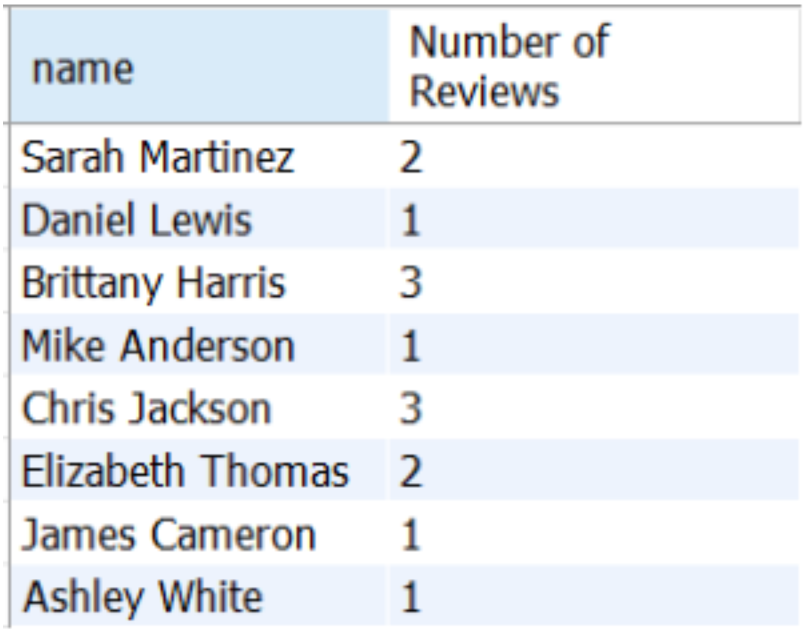
rid	title	director	stars
201	Gone with the Wind	Victor Fleming	2
201	Star Wars	George Lucas	4
202	Snow White	William Cottrell	4
203	The Sound of Music	Robert Wise	2
203	E.T.	Steven Spielberg	4
203	Raiders of the Lost Ark	Steven Spielberg	2
204	Gone with the Wind	Victor Fleming	3
205	The Sound of Music	Robert Wise	3
205	E.T.	Steven Spielberg	2
205	Raiders of the Lost Ark	Steven Spielberg	4
206	Avatar	James Cameron	3
206	Snow White	William Cottrell	5
207	Avatar	James Cameron	5
208	E.T.	Steven Spielberg	3

2. Return the number of reviews (in the ratings table) left by Chris Jackson. Label the output "Number of Chris's Reviews.



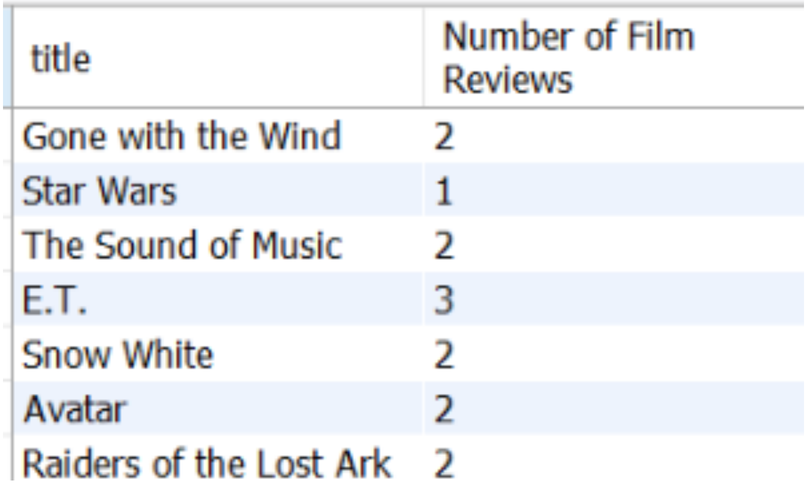
Number of Reviews left by Chris Jackson
3

3. Return the number of reviews left by each reviewer (identified by name). Label the number of reviews (Number of Reviews).



name	Number of Reviews
Sarah Martinez	2
Daniel Lewis	1
Brittany Harris	3
Mike Anderson	1
Chris Jackson	3
Elizabeth Thomas	2
James Cameron	1
Ashley White	1

4. Return the number of reviews left for each film (identified by title). Label the number of reviews (Number of Film Reviews).



title	Number of Film Reviews
Gone with the Wind	2
Star Wars	1
The Sound of Music	2
E.T.	3
Snow White	2
Avatar	2
Raiders of the Lost Ark	2

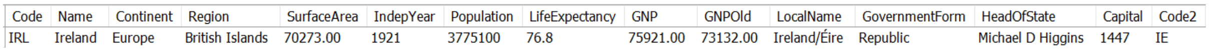
Geography Database Exercise

- Download and open the **Geography** script (as an SQL file).
- The *Geography* database has 3 tables: *country*, *city*, and *countrylanguage*.
- The tables are populated and you will use this data in your exercise. The schema is stored in **theworld.sql**.
- Open and Execute the SQL script in MySQL Workbench.

```
USE theworld; /*Select the theworld database*/
SHOW TABLES; /*List all the tables*/
```

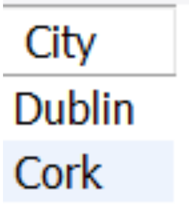
To do:

1. Describe each table.
2. Return all the data from the Country table for Ireland.



Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExpectancy	GNP	GNPOld	LocalName	GovernmentForm	HeadOfState	Capital	Code2
IRL	Ireland	Europe	British Islands	70273.00	1921	3775100	76.8	75921.00	73132.00	Ireland/Éire	Republic	Michael D Higgins	1447	IE

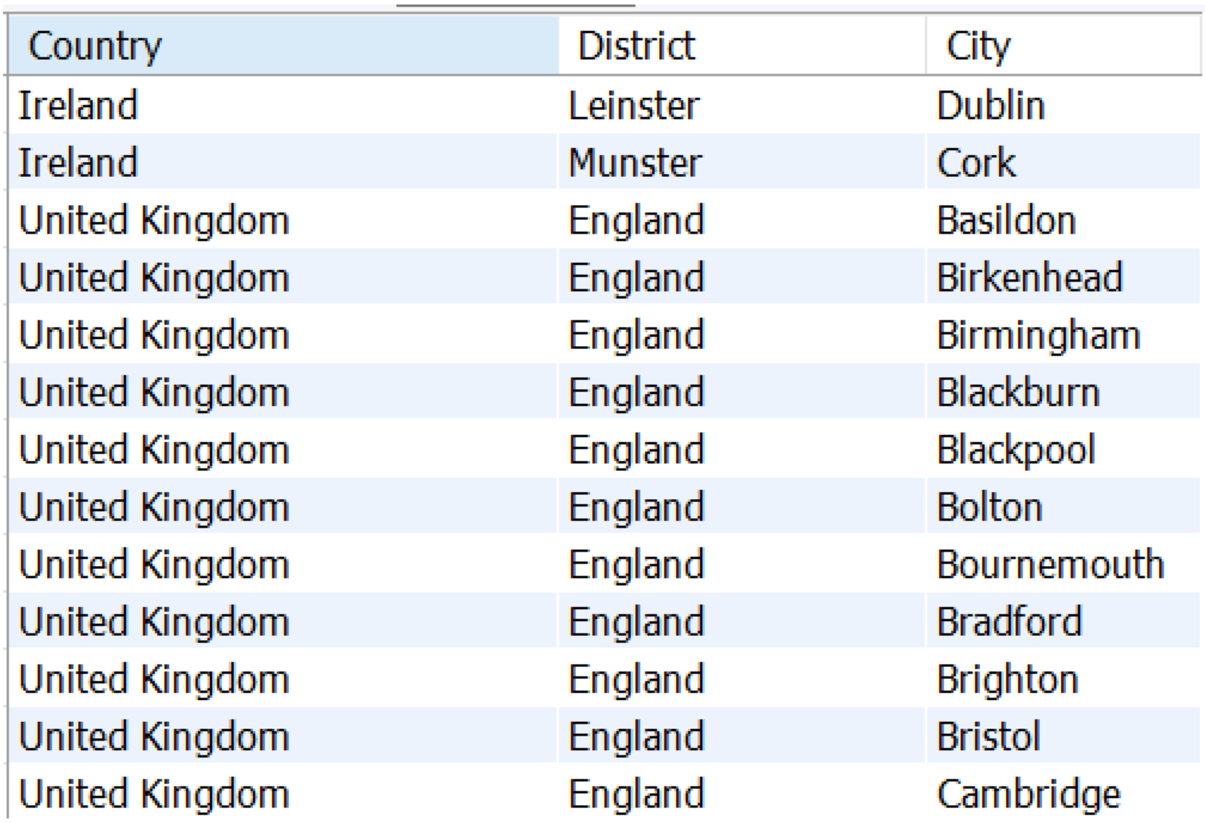
3. Return the names of all cities for Ireland. Sort in alphbetical order.



City
Dublin
Cork

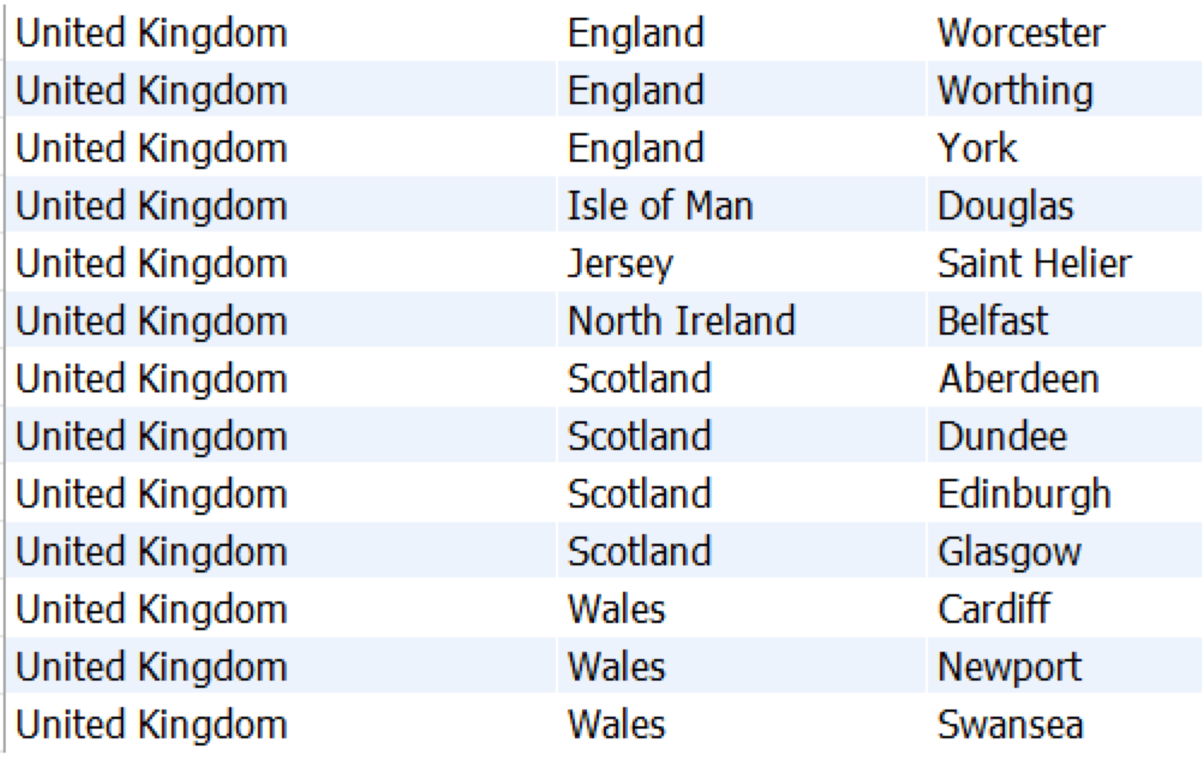
4. Return the country (by name), district, and city (by name) for Ireland and United Kingdom. Sort in alphbetical order of country, then district and then city.

Note: If columns returned from 2 different tables have the same name then you need to preface them by the table name.



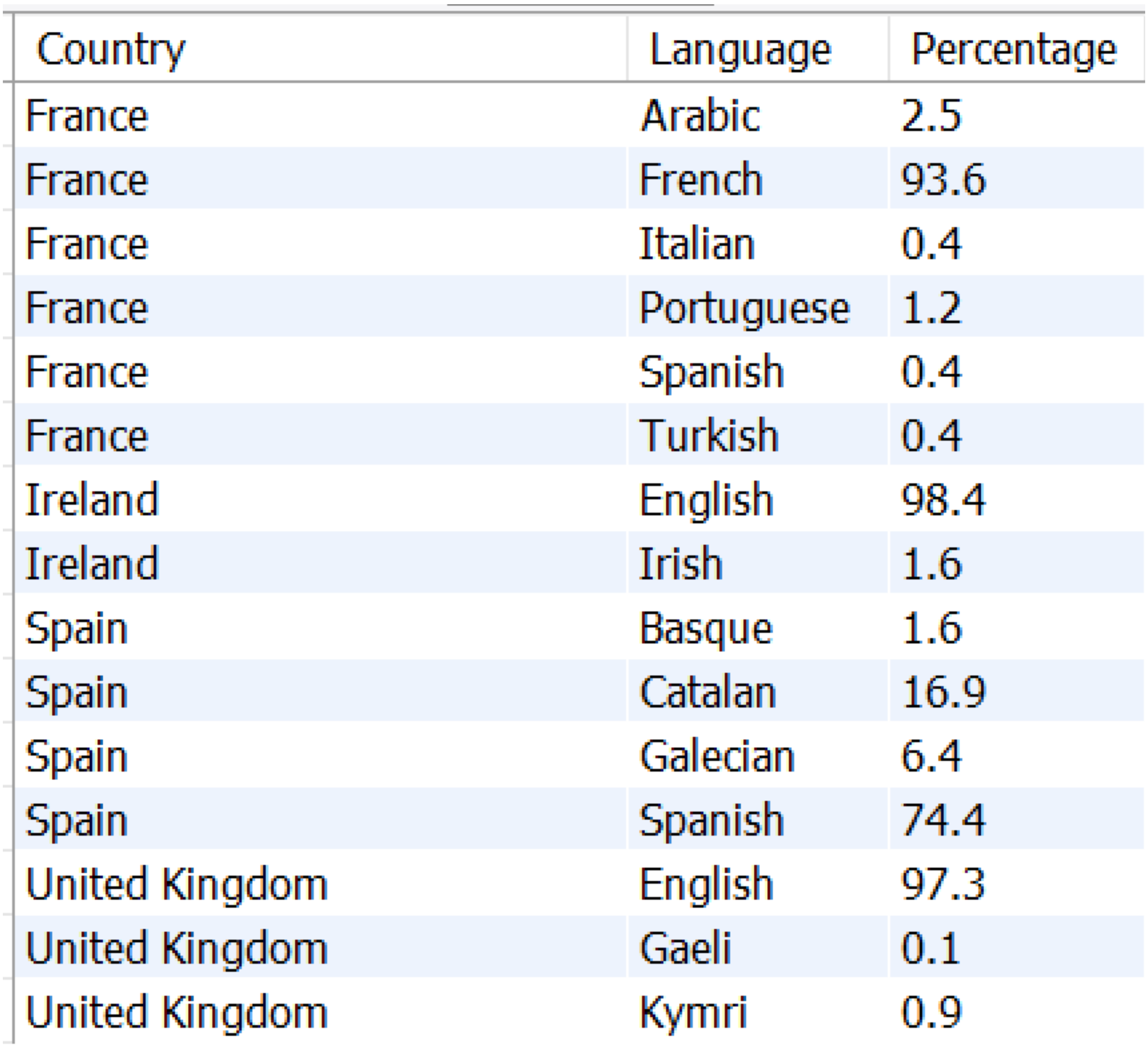
Country	District	City
Ireland	Leinster	Dublin
Ireland	Munster	Cork
United Kingdom	England	Basildon
United Kingdom	England	Birkenhead
United Kingdom	England	Birmingham
United Kingdom	England	Blackburn
United Kingdom	England	Blackpool
United Kingdom	England	Bolton
United Kingdom	England	Bournemouth
United Kingdom	England	Bradford
United Kingdom	England	Brighton
United Kingdom	England	Bristol
United Kingdom	England	Cambridge

…



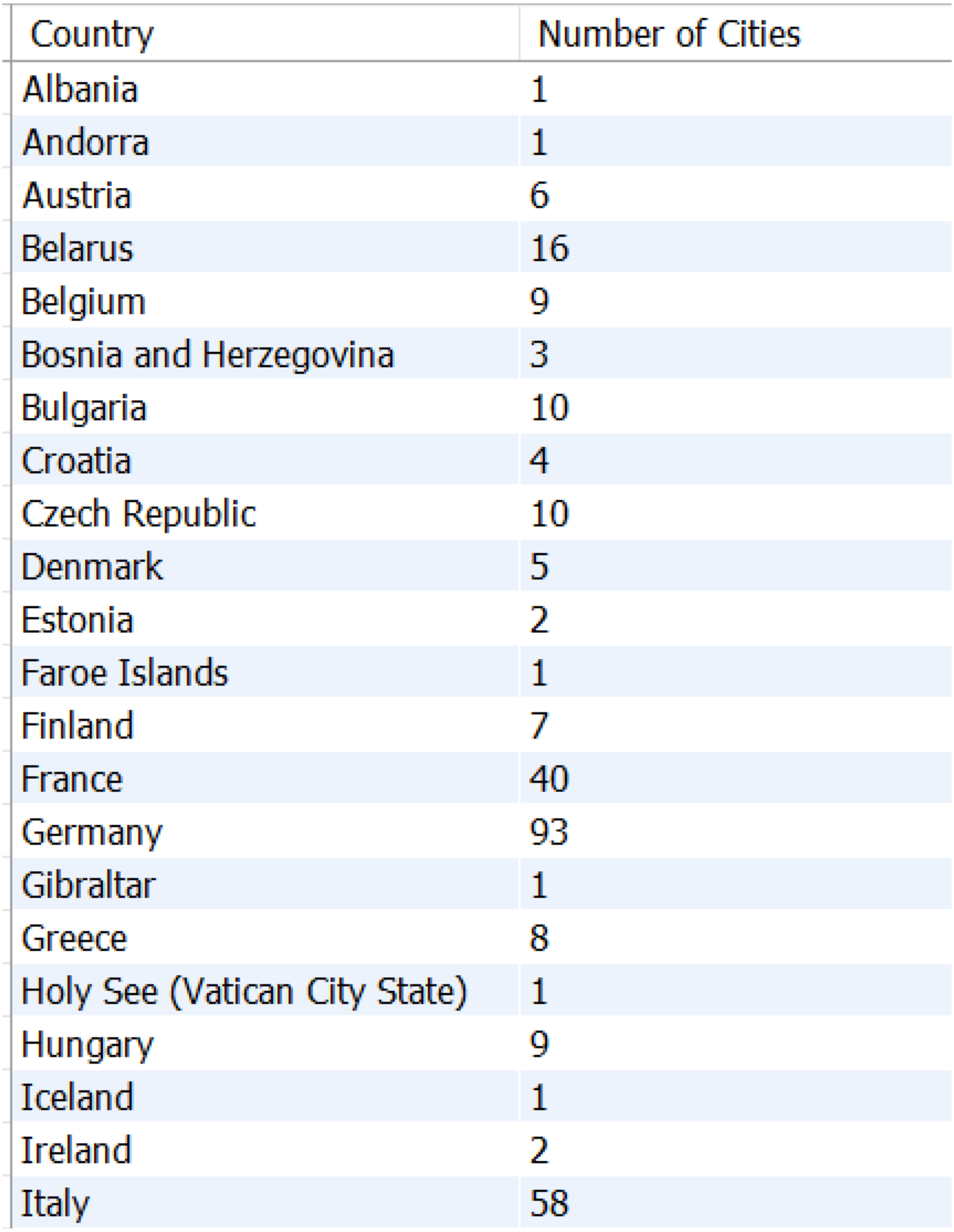
United Kingdom	England	Worcester
United Kingdom	England	Worthing
United Kingdom	England	York
United Kingdom	Isle of Man	Douglas
United Kingdom	Jersey	Saint Helier
United Kingdom	North Ireland	Belfast
United Kingdom	Scotland	Aberdeen
United Kingdom	Scotland	Dundee
United Kingdom	Scotland	Edinburgh
United Kingdom	Scotland	Glasgow
United Kingdom	Wales	Cardiff
United Kingdom	Wales	Newport
United Kingdom	Wales	Swansea

5. Return the country (by name), language and population (speaking the language) for Ireland, United Kingdom, France, and Spain. Sort in alphabetical order of country, and then language.

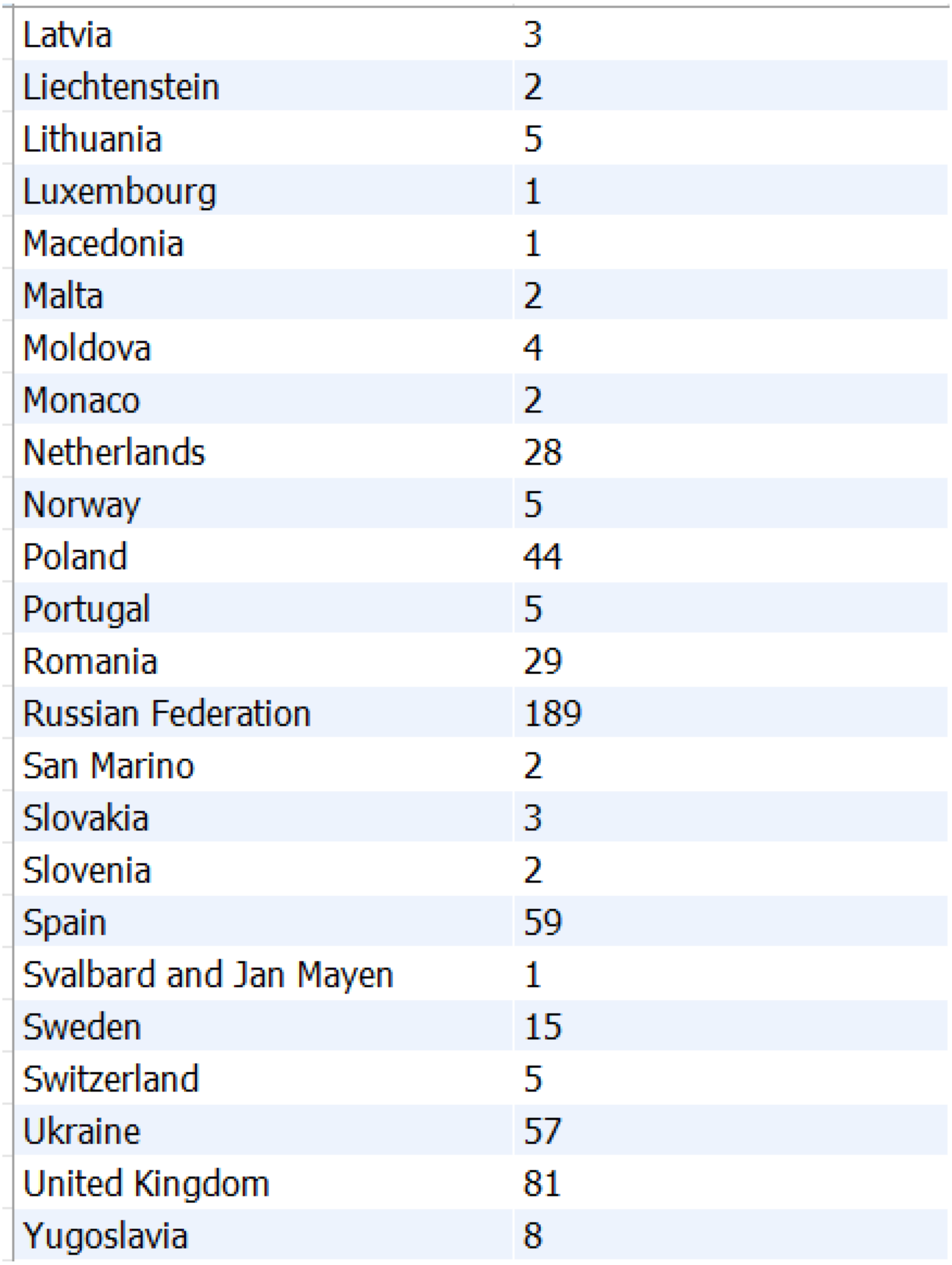


Country	Language	Percentage
France	Arabic	2.5
France	French	93.6
France	Italian	0.4
France	Portuguese	1.2
France	Spanish	0.4
France	Turkish	0.4
Ireland	English	98.4
Ireland	Irish	1.6
Spain	Basque	1.6
Spain	Catalan	16.9
Spain	Galecian	6.4
Spain	Spanish	74.4
United Kingdom	English	97.3
United Kingdom	Gaeli	0.1
United Kingdom	Kymri	0.9

6. Return the country (by name) and number of cities per country for all countries in Europe. Sort in alphbetical order.



Country	Number of Cities
Albania	1
Andorra	1
Austria	6
Belarus	16
Belgium	9
Bosnia and Herzegovina	3
Bulgaria	10
Croatia	4
Czech Republic	10
Denmark	5
Estonia	2
Faroe Islands	1
Finland	7
France	40
Germany	93
Gibraltar	1
Greece	8
Holy See (Vatican City State)	1
Hungary	9
Iceland	1
Ireland	2
Italy	58



Latvia	3
Liechtenstein	2
Lithuania	5
Luxembourg	1
Macedonia	1
Malta	2
Moldova	4
Monaco	2
Netherlands	28
Norway	5
Poland	44
Portugal	5
Romania	29
Russian Federation	189
San Marino	2
Slovakia	3
Slovenia	2
Spain	59
Svalbard and Jan Mayen	1
Sweden	15
Switzerland	5
Ukraine	57
United Kingdom	81
Yugoslavia	8