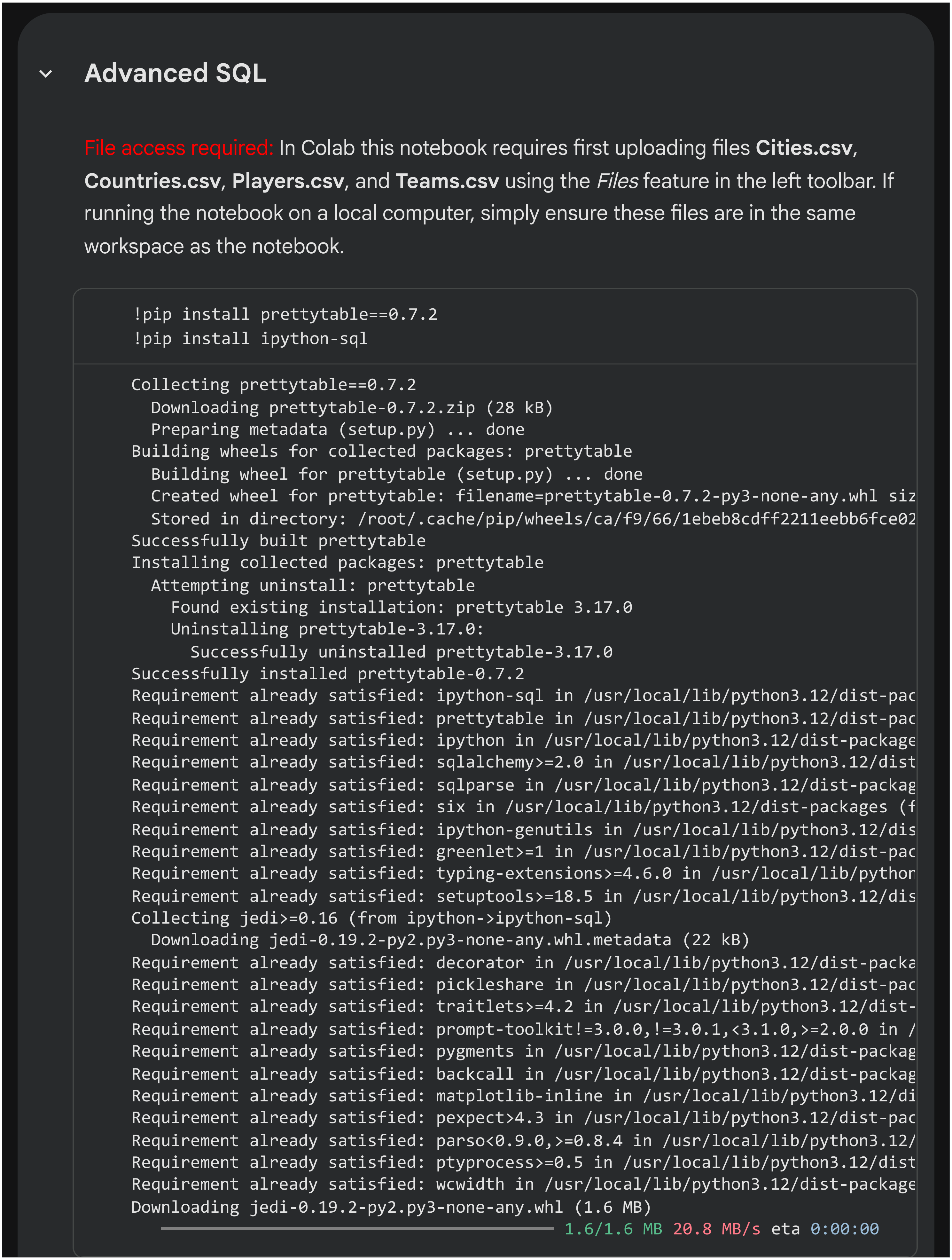


Advanced SQL

File access required: In Colab this notebook requires first uploading files **Cities.csv**, **Countries.csv**, **Players.csv**, and **Teams.csv** using the *Files* feature in the left toolbar. If running the notebook on a local computer, simply ensure these files are in the same workspace as the notebook.

```
!pip install prettytable==0.7.2
!pip install ipython-sql
```

```
Collecting prettytable==0.7.2
  Downloading prettytable-0.7.2.zip (28 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: prettytable
  Building wheel for prettytable (setup.py) ... done
  Created wheel for prettytable: filename=prettytable-0.7.2-py3-none-any.whl siz
  Stored in directory: /root/.cache/pip/wheels/ca/f9/66/1ebeb8cdff2211eebb6fce02
Successfully built prettytable
Installing collected packages: prettytable
  Attempting uninstall: prettytable
    Found existing installation: prettytable 3.17.0
    Uninstalling prettytable-3.17.0:
      Successfully uninstalled prettytable-3.17.0
Successfully installed prettytable-0.7.2
Requirement already satisfied: ipython-sql in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: prettytable in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: ipython in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: sqlalchemy>=2.0 in /usr/local/lib/python3.12/dist
Requirement already satisfied: sqlparse in /usr/local/lib/python3.12/dist-packag
Requirement already satisfied: six in /usr/local/lib/python3.12/dist-packages (f
Requirement already satisfied: ipython-genutils in /usr/local/lib/python3.12/dis
Requirement already satisfied: greenlet>=1 in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: typing-extensions>=4.6.0 in /usr/local/lib/python
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.12/dis
Collecting jedi>=0.16 (from ipython->ipython-sql)
  Downloading jedi-0.19.2-py2.py3-none-any.whl.metadata (22 kB)
Requirement already satisfied: decorator in /usr/local/lib/python3.12/dist-packa
Requirement already satisfied: pickleshare in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.12/dist-
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /
Requirement already satisfied: pygments in /usr/local/lib/python3.12/dist-packag
Requirement already satisfied: backcall in /usr/local/lib/python3.12/dist-packag
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.12/di
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: parso<0.9.0,>=0.8.4 in /usr/local/lib/python3.12/
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.12/dist
Requirement already satisfied: wcwidth in /usr/local/lib/python3.12/dist-package
Downloading jedi-0.19.2-py2.py3-none-any.whl (1.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 20.8 MB/s eta 0:00:00
```

```
Installing collected packages: jedi
Successfully installed jedi-0.19.2
```

```
# Set-up
%load_ext sql
%sql sqlite://
import pandas as pd
```

```
# Create database tables from CSV files
with open('Cities.csv') as f: Cities = pd.read_csv(f, index_col=0)
%sql drop table if exists Cities;
%sql --persist Cities
with open('Countries.csv') as f: Countries = pd.read_csv(f, index_col=0)
%sql drop table if exists Countries;
%sql --persist Countries
```

```
 * sqlite://
Done.
 * sqlite://
 * sqlite://
Done.
 * sqlite://
'Persisted countries'
```

Look at sample of Cities and Countries tables

```
%%sql
select * from Cities limit 5
```

```
 * sqlite://
Done.
   city        country      latitude longitude temperature
Aalborg   Denmark          57.03    9.92      7.52
Aberdeen  United Kingdom   57.17    -2.08     8.1
Abisko    Sweden           63.35    18.83     0.2
Adana     Turkey           36.99    35.32     18.67
Albacete  Spain            39.0     -1.87     12.62
```

```
%%sql
select * from Countries limit 5
```

```
 * sqlite://
Done.
country population EU  coastline
Albania  2.9        no  yes
Andorra  0.07       no  no
Austria  8.57       yes no
Belarus  9.48       no  no
Belgium  11.37      yes yes
```

Duplicates, table variables

Warm-up: Find all cities in the EU with temperature > 15

```
%%sql
select city
from Cities, Countries
where Cities.country = Countries.country
and EU = 'yes' and temperature > 15
```

```
 * sqlite://
Done.
   city
Algeciras
Athens
Badajoz
Barcelona
Bari
Cartagena
Catania
Cosenza
Granada
Huelva
Kalamata
Lisbon
Marbella
Messina
Palermo
Patras
Pescara
Rome
Trikala
Valencia
```

Modify previous query to return EU countries that have a city with temperature > 15, remove duplicates

Find number of countries that have a city with latitude > 60 (start with country list then fix)

```
%%sql
select country
from Cities
where latitude > 60
```

```
 * sqlite://
Done.
country
Sweden
Norway
Norway
Finland
Sweden
Finland
Finland
Norway
Finland
```

Modify first query to use table variables

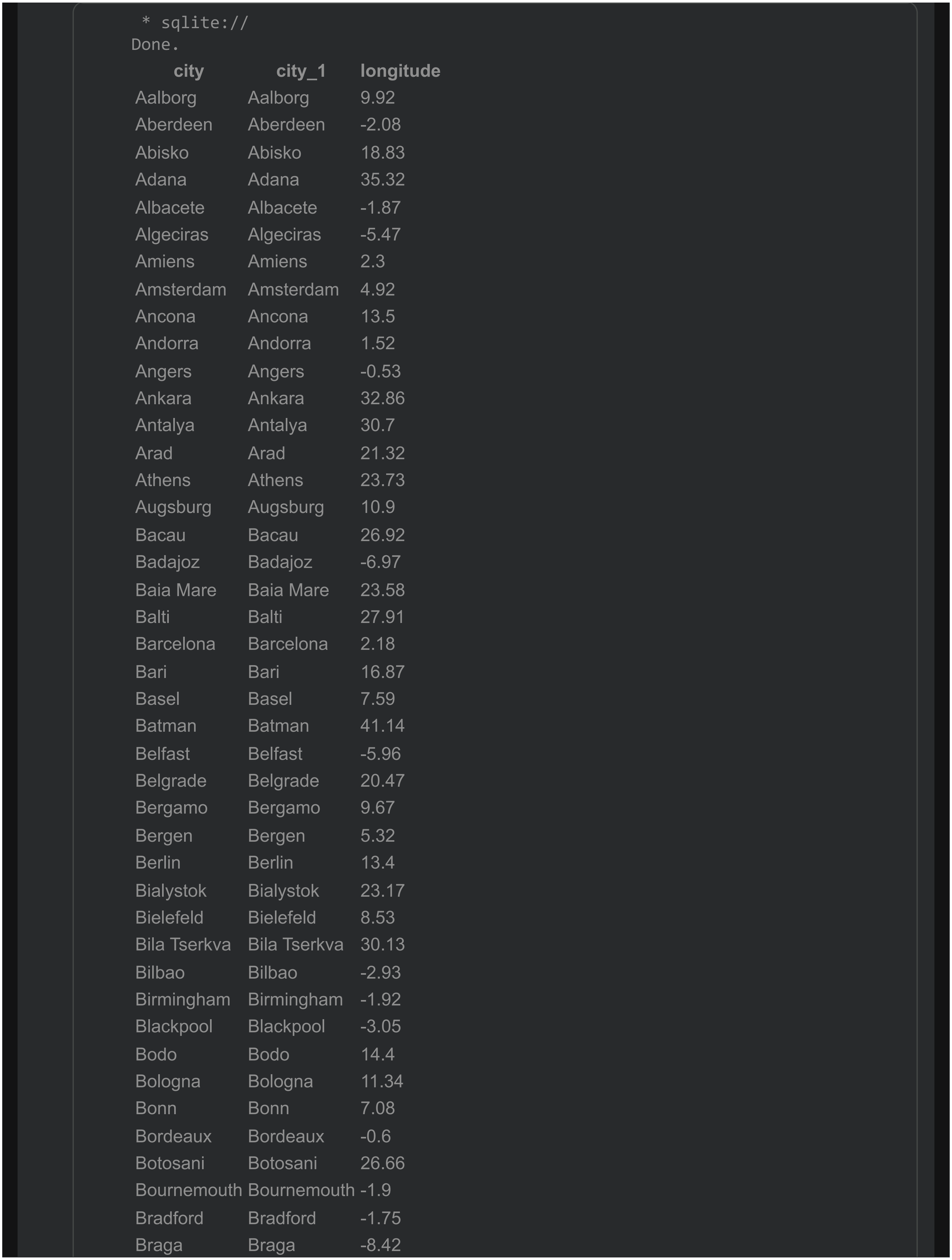
```
%%sql
select city
from Cities, Countries
where Cities.country = Countries.country
and EU = 'yes' and temperature > 15
```

```
 * sqlite://
Done.
```

city
Algeciras
Athens
Badajoz
Barcelona
Bari
Cartagena
Catania
Cosenza
Granada
Huelva
Kalamata
Lisbon
Marbella
Messina
Palermo
Patras
Pescara
Rome
Trikala
Valencia

Find all pairs of cities with the same longitude; return the city pairs and their (shared) longitude - notice what's wrong and fix it

```
%%sql
select C1.city, C2.city, C1.longitude
from Cities C1, Cities C2
where C1.longitude = C2.longitude
```



```
 * sqlite://
Done.
```

city	city_1	longitude
Aalborg	Aalborg	9.92
Aberdeen	Aberdeen	-2.08
Abisko	Abisko	18.83
Adana	Adana	35.32
Albacete	Albacete	-1.87
Algeciras	Algeciras	-5.47
Amiens	Amiens	2.3
Amsterdam	Amsterdam	4.92
Ancona	Ancona	13.5
Andorra	Andorra	1.52
Angers	Angers	-0.53
Ankara	Ankara	32.86
Antalya	Antalya	30.7
Arad	Arad	21.32
Athens	Athens	23.73
Augsburg	Augsburg	10.9
Bacau	Bacau	26.92
Badajoz	Badajoz	-6.97
Baia Mare	Baia Mare	23.58
Balti	Balti	27.91
Barcelona	Barcelona	2.18
Bari	Bari	16.87
Basel	Basel	7.59
Batman	Batman	41.14
Belfast	Belfast	-5.96
Belgrade	Belgrade	20.47
Bergamo	Bergamo	9.67
Bergen	Bergen	5.32
Berlin	Berlin	13.4
Bialystok	Bialystok	23.17
Bielefeld	Bielefeld	8.53
Bila Tserkva	Bila Tserkva	30.13
Bilbao	Bilbao	-2.93
Birmingham	Birmingham	-1.92
Blackpool	Blackpool	-3.05
Bodo	Bodo	14.4
Bologna	Bologna	11.34
Bonn	Bonn	7.08
Bordeaux	Bordeaux	-0.6
Botosani	Botosani	26.66
Bournemouth	Bournemouth	-1.9
Bradford	Bradford	-1.75
Braga	Braga	-8.42

Braila	Braila	27.97
Bratislava	Bratislava	17.12
Bremen	Bremen	8.8
Brest	Brest	-4.5
Brest	Brest	23.7
Brno	Brno	16.61
Brugge	Brugge	3.23
Bucharest	Bucharest	26.1
Budapest	Budapest	19.08
Burgas	Burgas	27.47
Burgos	Burgos	-3.68
Burgos	Madrid	-3.68
Bursa	Bursa	29.07
Bydgoszcz	Bydgoszcz	18.01
Bytom	Bytom	18.91
Caen	Caen	-0.35
Cambridge	Cambridge	0.12
Cartagena	Cartagena	-0.98
Catania	Catania	15.08
Chemnitz	Chemnitz	12.92
Cherkasy	Cherkasy	32.07
Chernihiv	Chernihiv	31.3
Chernivtsi	Chernivtsi	25.92
Chisinau	Chisinau	28.86
Constanta	Constanta	28.61
Cork	Cork	-8.5
Cosenza	Cosenza	16.29
Craiova	Craiova	23.83
Craiova	Hrodna	23.83
Daugavpils	Daugavpils	26.51
Debrecen	Debrecen	21.63
Denizli	Denizli	29.08
Dijon	Dijon	5.03
Dublin	Dublin	-6.25
Dundee	Dundee	-3.0
Edinburgh	Edinburgh	-3.22
Edirne	Edirne	26.57
Elbasan	Elbasan	20.08
Elblag	Elblag	19.4
Erfurt	Erfurt	11.03
Erzincan	Erzincan	39.49
Erzurum	Erzurum	41.29
Eskisehir	Eskisehir	30.53
Exeter	Exeter	-3.53
Foggia	Foggia	15.56
Frankfurt	Frankfurt	8.68

Freiburg	Freiburg	7.87
Galway	Galway	-9.05
Gaziantep	Gaziantep	37.38
Geneva	Geneva	6.14
Genoa	Genoa	8.93
Glasgow	Glasgow	-4.25
Göteborg	Göteborg	12.0
Granada	Granada	-3.59
Graz	Graz	15.41
Grenoble	Grenoble	5.72
Groningen	Groningen	6.58
Gyor	Gyor	17.63
Heidelberg	Heidelberg	8.7
Helsinki	Helsinki	24.93
Horlivka	Horlivka	38.05
Hrodna	Craiova	23.83
Hrodna	Hrodna	23.83
Huelva	Huelva	-6.93
Ingolstadt	Ingolstadt	11.45
Innsbruck	Innsbruck	11.41
Inverness	Inverness	-4.23
Istanbul	Istanbul	29.01
Kalamata	Kalamata	22.11
Karaman	Karaman	33.22
Karlsruhe	Karlsruhe	8.4
Kaunas	Kaunas	23.88
Kayseri	Kayseri	35.49
Kherson	Kherson	32.6
Kielce	Kielce	20.66
Kiev	Kiev	30.52
Kiruna	Kiruna	20.22
Klaipeda	Klaipeda	21.12
Kosice	Kosice	21.25
Koszalin	Koszalin	16.18
Kremenchuk	Kremenchuk	33.43
Kryvyy Rih	Kryvyy Rih	33.34
Le Mans	Le Mans	0.1
Lille	Lille	3.08
Limoges	Limoges	1.25
Linz	Linz	14.29
Lisbon	Lisbon	-9.14
Ljubljana	Ljubljana	14.51
Lvov	Lvov	24.03
Maastricht	Maastricht	5.68
Maastricht	Stavanger	5.68
Madrid	Burgos	-3.68

Madrid	Madrid	-3.68
Magdeburg	Magdeburg	11.62
Makiyivka	Makiyivka	37.97
Malatya	Malatya	38.3
Malmö	Malmö	13.03
Manisa	Manisa	27.44
Marbella	Marbella	-4.88
Marseille	Marseille	5.38
Mazyr	Mazyr	29.27
Messina	Messina	15.55
Metz	Metz	6.18
Milan	Milan	9.21
Minsk	Minsk	27.57
Montpellier	Montpellier	3.87
Mulhouse	Mulhouse	7.35
Murcia	Murcia	-1.13
Nice	Nice	7.27
Nis	Nis	21.9
Novi Sad	Novi Sad	19.85
Odense	Odense	10.38
Ordu	Ordu	37.87
Orsha	Orsha	30.42
Oslo	Oslo	10.75
Ostrava	Ostrava	18.25
Oulu	Oulu	25.47
Oviedo	Oviedo	-5.83
Palermo	Palermo	13.35
Patras	Patras	21.73
Perpignan	Perpignan	2.9
Perugia	Perugia	12.39
Pescara	Pescara	14.22
Pinsk	Pinsk	26.09
Pleven	Pleven	24.61
Plovdiv	Plovdiv	24.75
Podgorica	Podgorica	19.27
Poznan	Poznan	16.9
Prague	Prague	14.47
Reims	Reims	4.03
Riga	Riga	24.1
Rijeka	Rijeka	14.45
Rivne	Rivne	26.25
Rome	Rome	12.48
Rostock	Rostock	12.15
Ruse	Ruse	25.97
Salamanca	Salamanca	-5.67
Salzburg	Salzburg	13.04

Find all pairs of cities that are near each other, i.e., longitude and latitude are both less than 0.5 apart; return city pairs

```
%%sql
select C1.city, C2.city
from Cities C1, Cities C2
where abs(C1.longitude - C2.longitude) < .5
and abs(C1.latitude - C2.latitude) < .5
and C1.city < C2.city
```

```
* sqlite://
Done.
```

city	city_1
Adana	Tarsus
Ancona	Sarajevo
Basel	Freiburg
Basel	Mulhouse
Bergamo	Milan
Cartagena	Murcia
Heidelberg	Karlsruhe
Horlivka	Makiyivka

Your Turn

Find all pairs of cities with the same temperature. Return the city pairs along with their shared temperature. What do you think about the data after seeing the answer?

```
%%sql
select C1.city as city1, C2.city as city2, C1.temperature
from Cities C1, Cities C2
where C1.temperature = C2.temperature
and C1.city < C2.city;
```

Samsun	Samsun	36.34
Santander	Santander	-3.8
Sarajevo	Sarajevo	13.38
Sibiu	Sibiu	24.14
Stavanger	Stavanger	5.68
Stockholm	Stockholm	18.1
Sumy	Sumy	34.78
Swansea	Swansea	-3.95
Szeged	Szeged	20.15
Tallinn	Tallinn	24.73
Tampere	Tampere	23.75
Tarsus	Tarsus	34.88
Tartu	Tartu	26.71
Tekirdag	Tekirdag	27.51
Toulouse	Toulouse	1.45
Trabzon	Trabzon	39.72
Trieste	Trieste	13.8
Trikala	Trikala	21.77
Trondheim	Trondheim	10.42
Turku	Turku	22.25
Uppsala	Uppsala	17.64
Valencia	Valencia	-0.4
Vienna	Vienna	16.37
Zaragoza	Zaragoza	-0.89
Zhytomyr	Zhytomyr	28.66
Zonguldak	Zonguldak	31.78
Zurich	Zurich	8.56

```
 * sqlite://
Done.
```

city1	city2	temperature
Andorra	Sarajevo	9.6
Augsburg	Innsbruck	4.54
Baia Mare	Debrecen	8.87
Balti	Botosani	8.23
Brno	Vienna	7.86
Brugge	Gyor	9.65
Brugge	Lille	9.65
Bucharest	Le Mans	10.59
Bucharest	Ruse	10.59
Bytom	Ostrava	7.66
Chemnitz	Prague	8.05
Chemnitz	Sivas	8.05
Chisinau	Glasgow	8.6
Craiova	Pleven	10.51
Daugavpils	Vilnius	5.38
Edinburgh	Kremenchuk	7.43
Geneva	Grenoble	8.4
Gyor	Lille	9.65
Karlsruhe	Metz	8.88
Koszalin	Odense	7.73
Le Mans	Ruse	10.59
Ljubljana	Rijeka	9.27
Manisa	Rome	15.1
Mulhouse	Zurich	6.68
Nis	Skopje	9.36
Novi Sad	Szeged	10.34
Prague	Sivas	8.05

Subqueries in Where clause

Find all countries in the Countries table with no city in the Cities table

```
%%sql
select country
from Countries
where not exists (

  select * from Cities
  where Cities.country = Countries.country

)
```

```
 * sqlite://
Done.
```

country
Cyprus
Iceland
Kosovo
Liechtenstein
Luxembourg

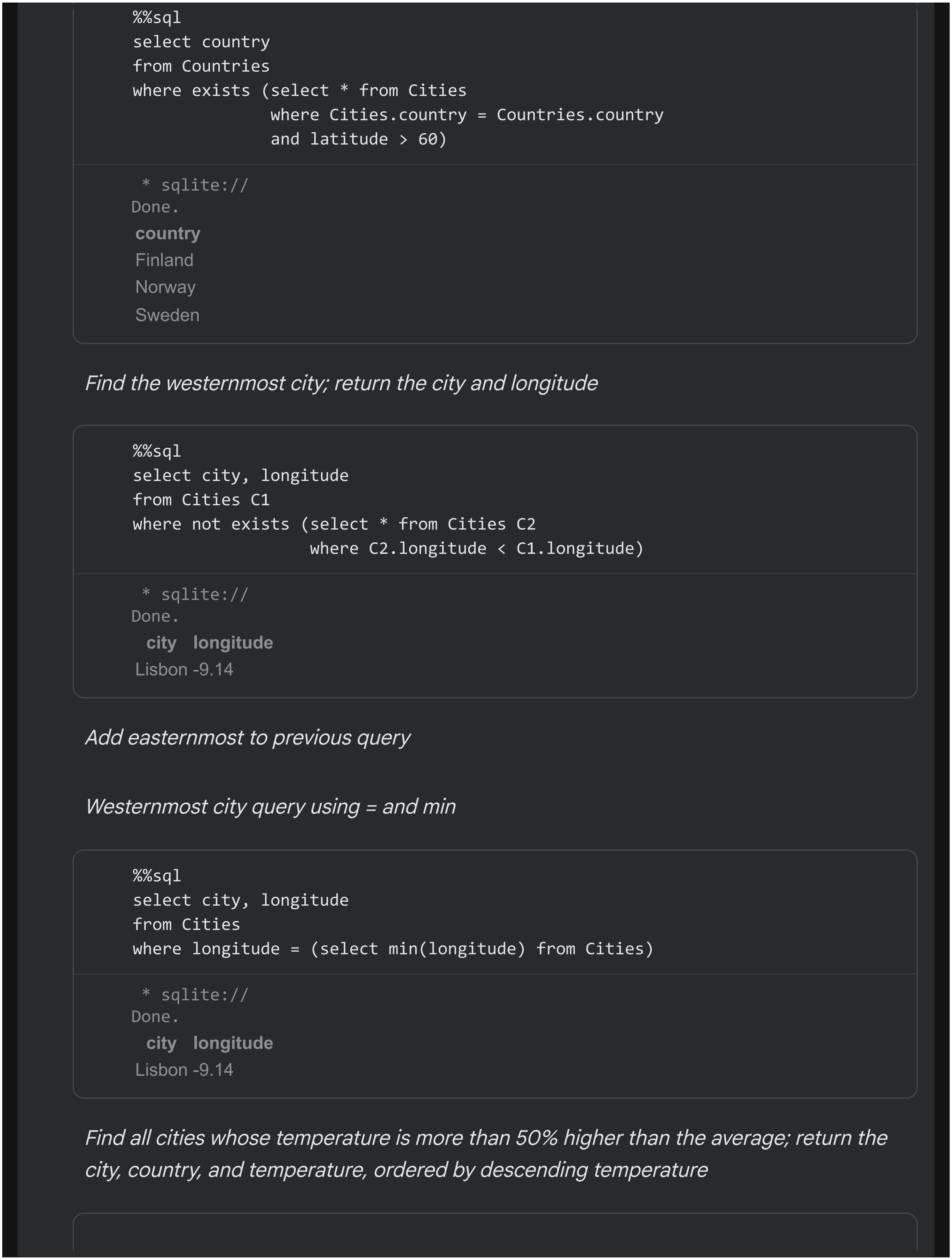
Find countries in the EU that have a city with temperature > 15

```
%%sql
select country
from Countries
where EU = 'yes'
and exists (select * from Cities
            where Cities.country = Countries.country
            and temperature > 15)
```

```
 * sqlite://
Done.
```

country
Greece
Italy
Portugal
Spain

Find number of countries that have a city with latitude > 60 (start with country list)



```
%%sql
select country
from Countries
where exists (select * from Cities
              where Cities.country = Countries.country
              and latitude > 60)
```

```
 * sqlite://
Done.
country
Finland
Norway
Sweden
```

Find the westernmost city; return the city and longitude

```
%%sql
select city, longitude
from Cities C1
where not exists (select * from Cities C2
                  where C2.longitude < C1.longitude)
```

```
 * sqlite://
Done.
  city   longitude
Lisbon -9.14
```

Add easternmost to previous query

Westernmost city query using = and min

```
%%sql
select city, longitude
from Cities
where longitude = (select min(longitude) from Cities)
```

```
 * sqlite://
Done.
  city   longitude
Lisbon -9.14
```

Find all cities whose temperature is more than 50% higher than the average; return the city, country, and temperature, ordered by descending temperature

```
%%sql
select city, country, temperature
from Cities
where temperature > (select avg(temperature) * 1.5 from Cities)
order by temperature desc
```

```
 * sqlite://
Done.
   city     country  temperature
Adana       Turkey   18.67
Palermo     Italy    17.9
Athens      Greece   17.41
Algeciras   Spain    17.38
Cartagena   Spain    17.32
Kalamata    Greece   17.3
Marbella    Spain    17.19
Huelva      Spain    17.09
Patras      Greece   16.9
Cosenza     Italy    16.6
Messina     Italy    16.54
Granada     Spain    16.33
Valencia    Spain    16.02
Trikala     Greece   16.0
Barcelona   Spain    15.78
Badajoz     Spain    15.61
Lisbon      Portugal 15.52
Elbasan     Albania  15.18
Bari        Italy    15.15
Pescara     Italy    15.13
Manisa      Turkey   15.1
Rome        Italy    15.1
Catania     Italy    15.04
Denizli     Turkey   15.02
Murcia      Spain    15.0
Marseille   France   14.98
```

Number of cities in the EU

```
%%sql
select count()
from Cities
where country in (select country from Countries where EU = 'yes')
```

```
 * sqlite://
Done.
count()
150
```

Modify previous query to use "not in"

Same query using join instead of subquery

```
%%sql
select count()
from Cities, Countries
where Cities.country = Countries.country
and EU = 'yes'
```

```
 * sqlite://
Done.
count()
150
```

Number of countries with no coastline and a city with longitude < 20

```
%%sql
select count()
from Countries
where coastline = 'no'
and exists (select * from Cities where country = Countries.country
            and longitude < 20)
```

```
 * sqlite://
Done.
count()
7
```

Same query using join instead of subquery (see what's wrong and fix it)

```
%%sql
select count()
from Countries, Cities
where Countries.country = Cities.country
and coastline = 'no' and longitude < 20
```

```
 * sqlite://
Done.
count()
16
```

Find countries in Countries table with no city in Cities table using join instead of subquery (subquery version repeated first)

```
%%sql
select country
from Countries
where not exists (select * from Cities
                  where Cities.country = Countries.country)
```

```
 * sqlite://
Done.
  country
Cyprus
Iceland
Kosovo
Liechtenstein
Luxembourg
```

```
%%sql
FILL IN
```

```
 * sqlite://
(sqlite3.OperationalError) near "FILL": syntax error
[SQL: FILL IN]
(Background on this error at: https://sqlalche.me/e/20/e3q8)
```

Your Turn

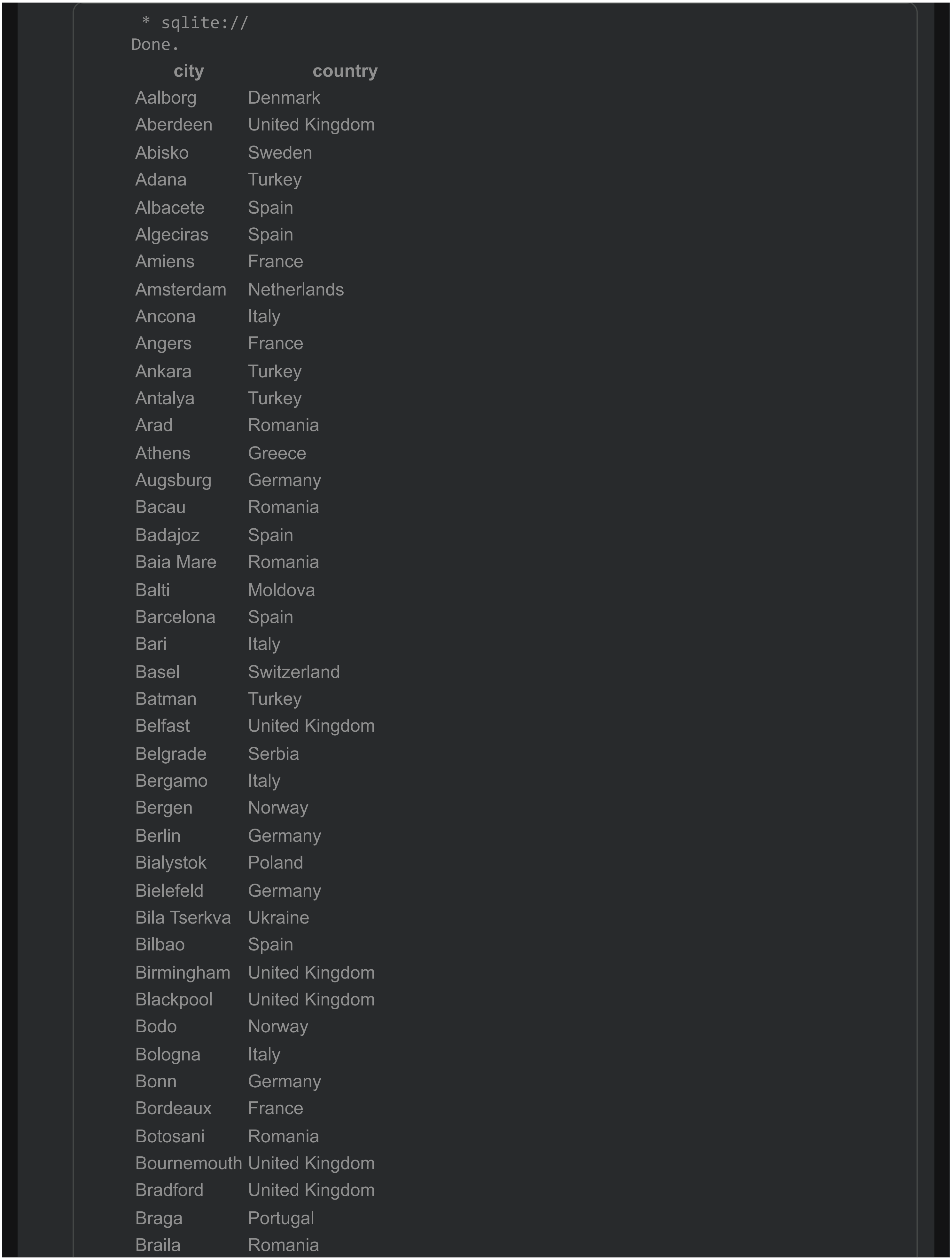
Find all cities in a country whose population is < 2; return the city and country. First write the query without a subquery.

```
%%sql
select city, Cities.country
from Cities join Countries
on Cities.country = Countries.country
where population < 2;
```

```
 * sqlite://
Done.
   city       country
Andorra     Andorra
Daugavpils  Latvia
Podgorica   Montenegro
Riga        Latvia
Tallinn     Estonia
Tartu       Estonia
```

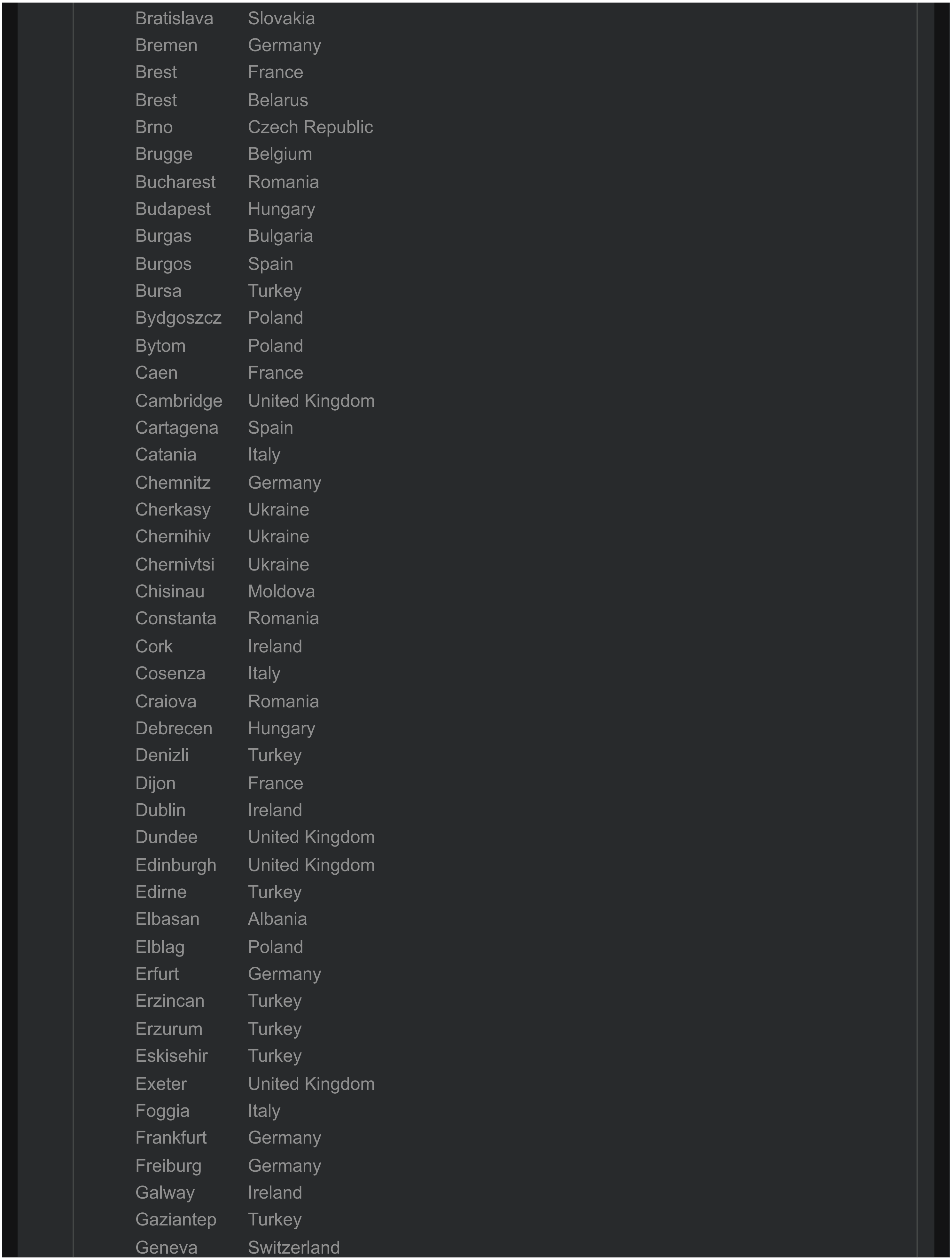
Now write the same query using a subquery instead of a join.

```
%%sql
select city,
       country
from Cities
where country in (
    select country
    from Countries
    where population >= 2
);
```

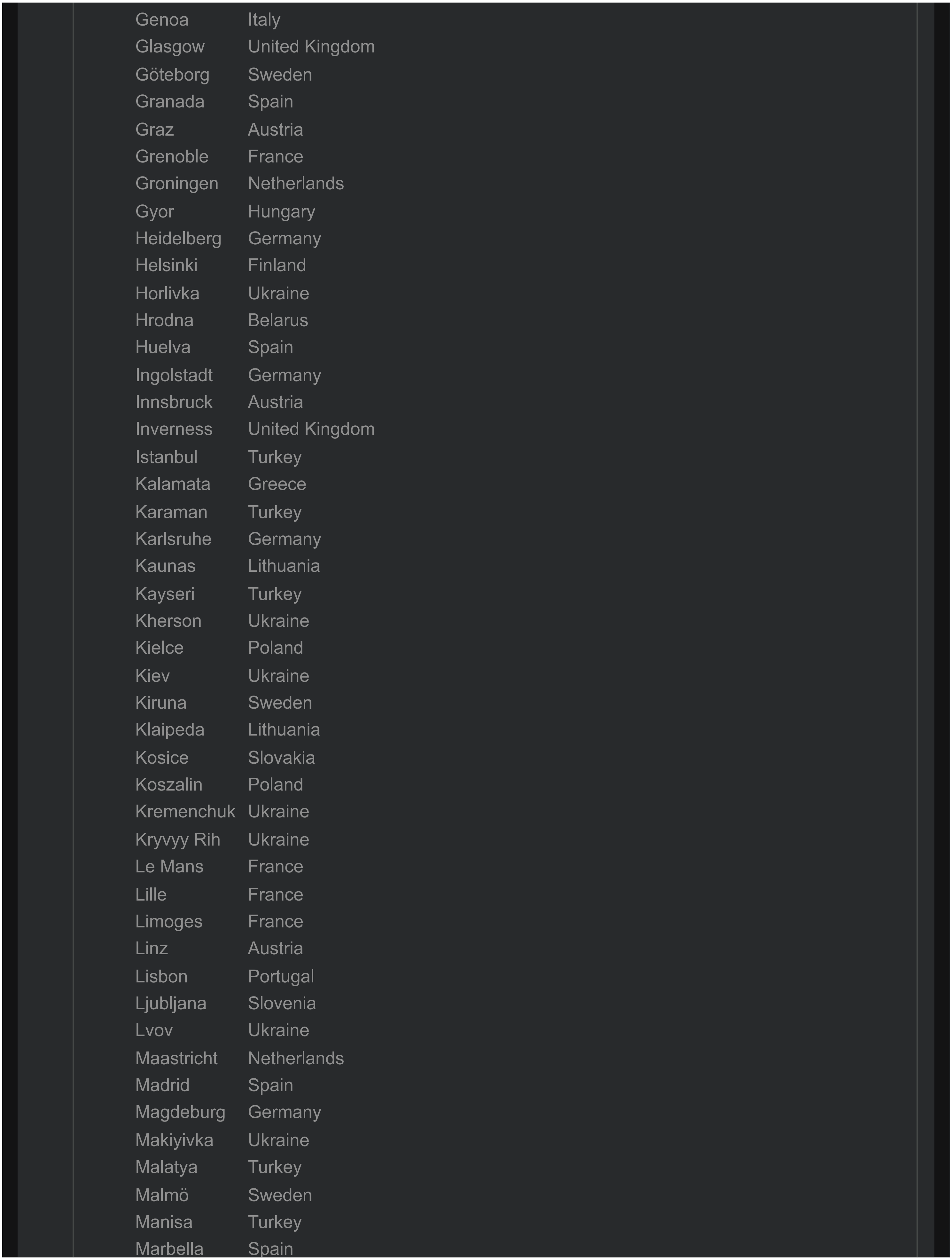


```
 * sqlite://
Done.
```

city	country
Aalborg	Denmark
Aberdeen	United Kingdom
Abisko	Sweden
Adana	Turkey
Albacete	Spain
Algeciras	Spain
Amiens	France
Amsterdam	Netherlands
Ancona	Italy
Angers	France
Ankara	Turkey
Antalya	Turkey
Arad	Romania
Athens	Greece
Augsburg	Germany
Bacau	Romania
Badajoz	Spain
Baia Mare	Romania
Balti	Moldova
Barcelona	Spain
Bari	Italy
Basel	Switzerland
Batman	Turkey
Belfast	United Kingdom
Belgrade	Serbia
Bergamo	Italy
Bergen	Norway
Berlin	Germany
Bialystok	Poland
Bielefeld	Germany
Bila Tserkva	Ukraine
Bilbao	Spain
Birmingham	United Kingdom
Blackpool	United Kingdom
Bodo	Norway
Bologna	Italy
Bonn	Germany
Bordeaux	France
Botosani	Romania
Bournemouth	United Kingdom
Bradford	United Kingdom
Braga	Portugal
Braila	Romania



Bratislava	Slovakia
Bremen	Germany
Brest	France
Brest	Belarus
Brno	Czech Republic
Brugge	Belgium
Bucharest	Romania
Budapest	Hungary
Burgas	Bulgaria
Burgos	Spain
Bursa	Turkey
Bydgoszcz	Poland
Bytom	Poland
Caen	France
Cambridge	United Kingdom
Cartagena	Spain
Catania	Italy
Chemnitz	Germany
Cherkasy	Ukraine
Chernihiv	Ukraine
Chernivtsi	Ukraine
Chisinau	Moldova
Constanta	Romania
Cork	Ireland
Cosenza	Italy
Craiova	Romania
Debrecen	Hungary
Denizli	Turkey
Dijon	France
Dublin	Ireland
Dundee	United Kingdom
Edinburgh	United Kingdom
Edirne	Turkey
Elbasan	Albania
Elblag	Poland
Erfurt	Germany
Erzincan	Turkey
Erzurum	Turkey
Eskisehir	Turkey
Exeter	United Kingdom
Foggia	Italy
Frankfurt	Germany
Freiburg	Germany
Galway	Ireland
Gaziantep	Turkey
Geneva	Switzerland



Genoa	Italy
Glasgow	United Kingdom
Göteborg	Sweden
Granada	Spain
Graz	Austria
Grenoble	France
Groningen	Netherlands
Gyor	Hungary
Heidelberg	Germany
Helsinki	Finland
Horlivka	Ukraine
Hrodna	Belarus
Huelva	Spain
Ingolstadt	Germany
Innsbruck	Austria
Inverness	United Kingdom
Istanbul	Turkey
Kalamata	Greece
Karaman	Turkey
Karlsruhe	Germany
Kaunas	Lithuania
Kayseri	Turkey
Kherson	Ukraine
Kielce	Poland
Kiev	Ukraine
Kiruna	Sweden
Klaipeda	Lithuania
Kosice	Slovakia
Koszalin	Poland
Kremenchuk	Ukraine
Kryvyy Rih	Ukraine
Le Mans	France
Lille	France
Limoges	France
Linz	Austria
Lisbon	Portugal
Ljubljana	Slovenia
Lvov	Ukraine
Maastricht	Netherlands
Madrid	Spain
Magdeburg	Germany
Makiyivka	Ukraine
Malatya	Turkey
Malmö	Sweden
Manisa	Turkey
Marbella	Spain

Marseille	France
Mazyr	Belarus
Messina	Italy
Metz	France
Milan	Italy
Minsk	Belarus
Montpellier	France
Mulhouse	France
Murcia	Spain
Nice	France
Nis	Serbia
Novi Sad	Serbia
Odense	Denmark
Ordu	Turkey
Orsha	Belarus
Oslo	Norway
Ostrava	Czech Republic
Oulu	Finland
Oviedo	Spain
Palermo	Italy
Patras	Greece
Perpignan	France
Perugia	Italy
Pescara	Italy
Pinsk	Belarus
Pleven	Bulgaria
Plovdiv	Bulgaria
Poznan	Poland
Prague	Czech Republic
Reims	France
Rijeka	Croatia
Rivne	Ukraine
Rome	Italy
Rostock	Germany
Ruse	Bulgaria
Salamanca	Spain
Salzburg	Austria
Samsun	Turkey
Santander	Spain
Sarajevo	Bosnia and Herzegovina
Sibiu	Romania
Siirt	Turkey
Sivas	Turkey
Skopje	Macedonia
Split	Croatia
Stara Zagora	Bulgaria

```
Stavanger     Norway
Stockholm     Sweden
Trikala       Greece
Trondheim     Norway
Turku         Finland
Uppsala       Sweden
Valencia      Spain
Vienna        Austria
Vigo          Spain
Vilnius       Lithuania
Warsaw        Poland
Wroclaw       Poland
Yevpatoriya   Ukraine
Zaragoza      Spain
Zhytomyr      Ukraine
Zonguldak     Turkey
Zurich        Switzerland
```

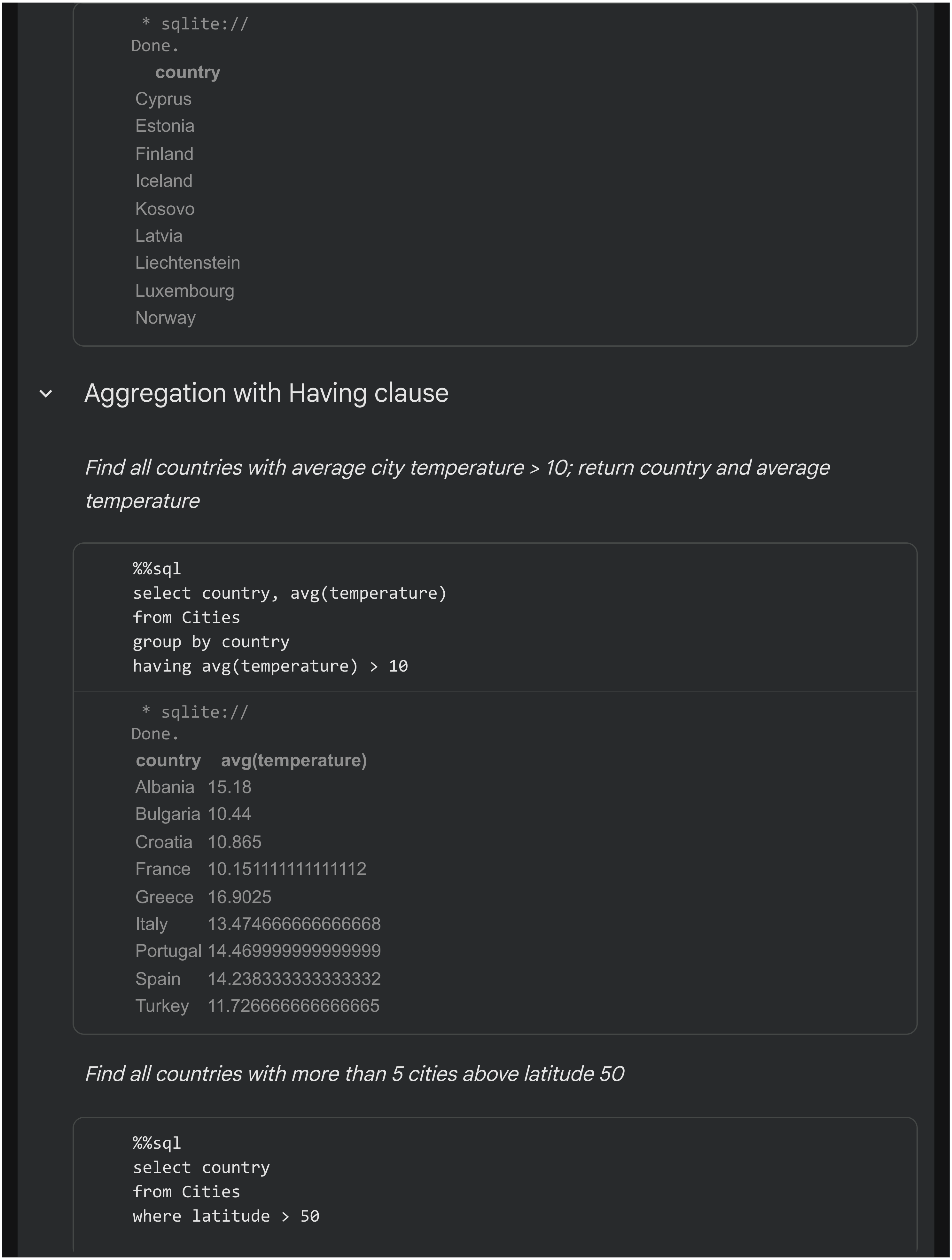
Find all countries with no city having a temperature > 6

```
%%sql
select Country
from Countries
where not exists (
  select *
  from Cities
  where Cities.country = Countries.country
  and temperature > 6
);
```

```
 * sqlite://
Done.
  country
Cyprus
Estonia
Finland
Iceland
Kosovo
Latvia
Liechtenstein
Luxembourg
Norway
```

Now try to write the same query without a subquery; can you?

```
%%sql
select Countries.country
from Countries
left join Cities
on Countries.country = Cities.country
group by Countries.country
having max(
  case when Cities.temperature > 6 then 1
  else 0 end
) = 0
```



```
 * sqlite://
Done.
  country
Cyprus
Estonia
Finland
Iceland
Kosovo
Latvia
Liechtenstein
Luxembourg
Norway
```

Aggregation with Having clause

Find all countries with average city temperature > 10; return country and average temperature

```
%%sql
select country, avg(temperature)
from Cities
group by country
having avg(temperature) > 10
```

```
 * sqlite://
Done.
country   avg(temperature)
Albania   15.18
Bulgaria  10.44
Croatia   10.865
France    10.151111111111112
Greece    16.9025
Italy     13.474666666666668
Portugal  14.469999999999999
Spain     14.238333333333332
Turkey    11.726666666666665
```

Find all countries with more than 5 cities above latitude 50

```
%%sql
select country
from Cities
where latitude > 50
```

```
group by country
having count() > 5
```

```
 * sqlite://
Done.
   country
Belarus
Germany
Poland
Sweden
United Kingdom
```

Same query without Having clause

```
%%sql
select distinct country
from Cities C1
where 5 < (select count() from Cities C2
           where C1.country=C2.country
           and latitude > 50)
```

```
 * sqlite://
Done.
   country
United Kingdom
Sweden
Germany
Poland
Belarus
```

Which combinations of EU versus non-EU and coastline versus no-coastline have a minimum population greater than 0.5?

```
%%sql
select EU, coastline, min(population)
from Countries
group by EU, coastline
having min(population) > 0.5
```

```
 * sqlite://
Done.
EU  coastline min(population)
yes no        0.58
yes yes       1.18
```

Find all countries with average city temperature more than 50% higher than the overall average; return country and average temperature

```
%%sql
select country, avg(temperature)
from Cities
group by country
having avg(temperature) > (select 1.5 * avg(temperature) from Cities)
```

```
 * sqlite://
Done.
```

country	avg(temperature)
Albania	15.18
Greece	16.9025
Portugal	14.469999999999999

Your Turn

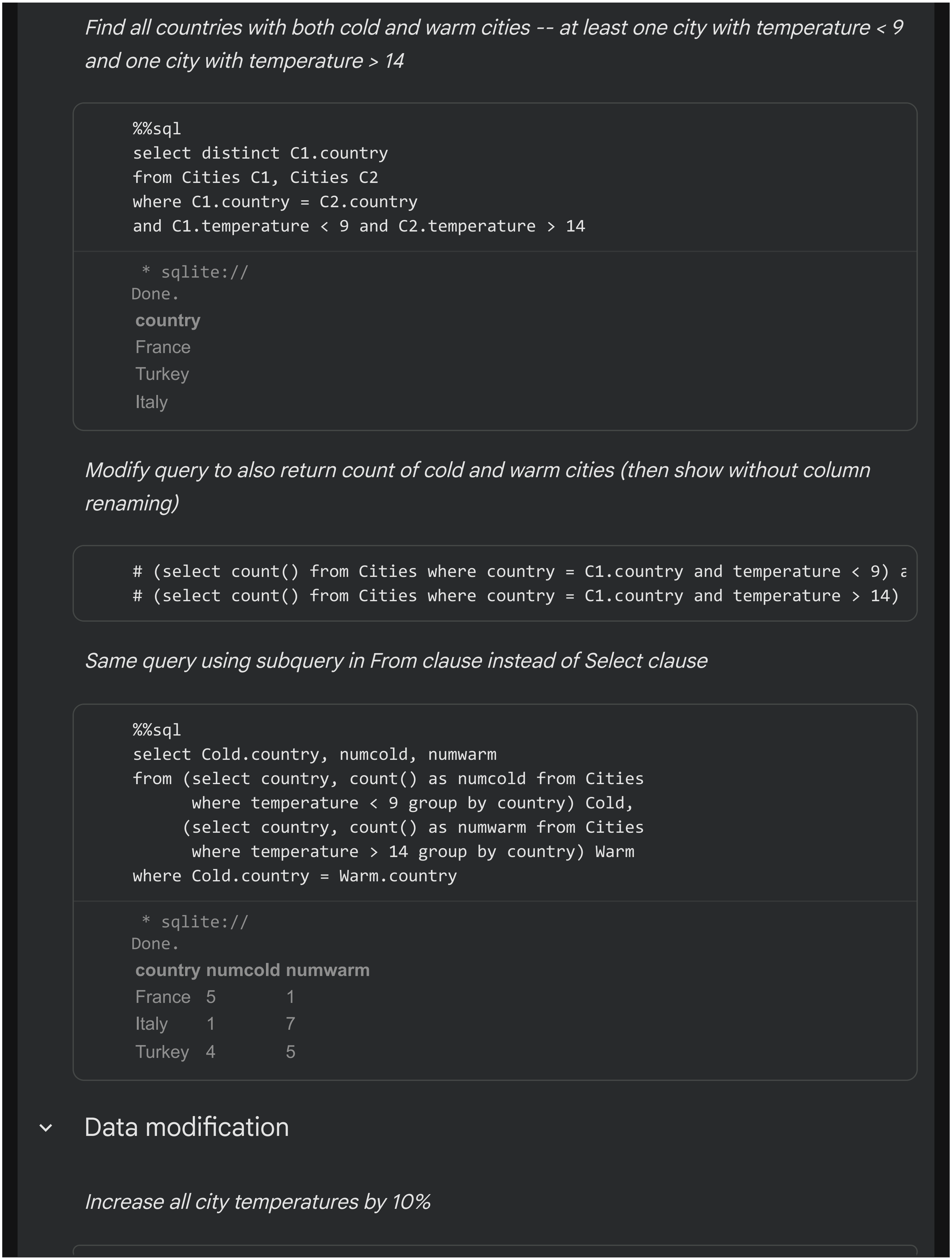
Find all countries whose average city longitude is lower than the overall average longitude, and whose average city latitude is higher than the overall average latitude. Return the countries. Note: Yes, you can use "and" in Having clauses!

```
%%sql
select country
from Cities
group by country
having avg(longitude) < (select avg(longitude) from Cities)
and avg(latitude) > (select avg(latitude) from Cities)
```

```
 * sqlite://
Done.
```

country
Austria
Belgium
Denmark
Germany
Ireland
Netherlands
Norway
United Kingdom

Subqueries in From and Select clauses



Find all countries with both cold and warm cities -- at least one city with temperature < 9 and one city with temperature > 14

```
%%sql
select distinct C1.country
from Cities C1, Cities C2
where C1.country = C2.country
and C1.temperature < 9 and C2.temperature > 14
```

```
 * sqlite://
Done.
country
France
Turkey
Italy
```

Modify query to also return count of cold and warm cities (then show without column renaming)

```
# (select count() from Cities where country = C1.country and temperature < 9) a
# (select count() from Cities where country = C1.country and temperature > 14)
```

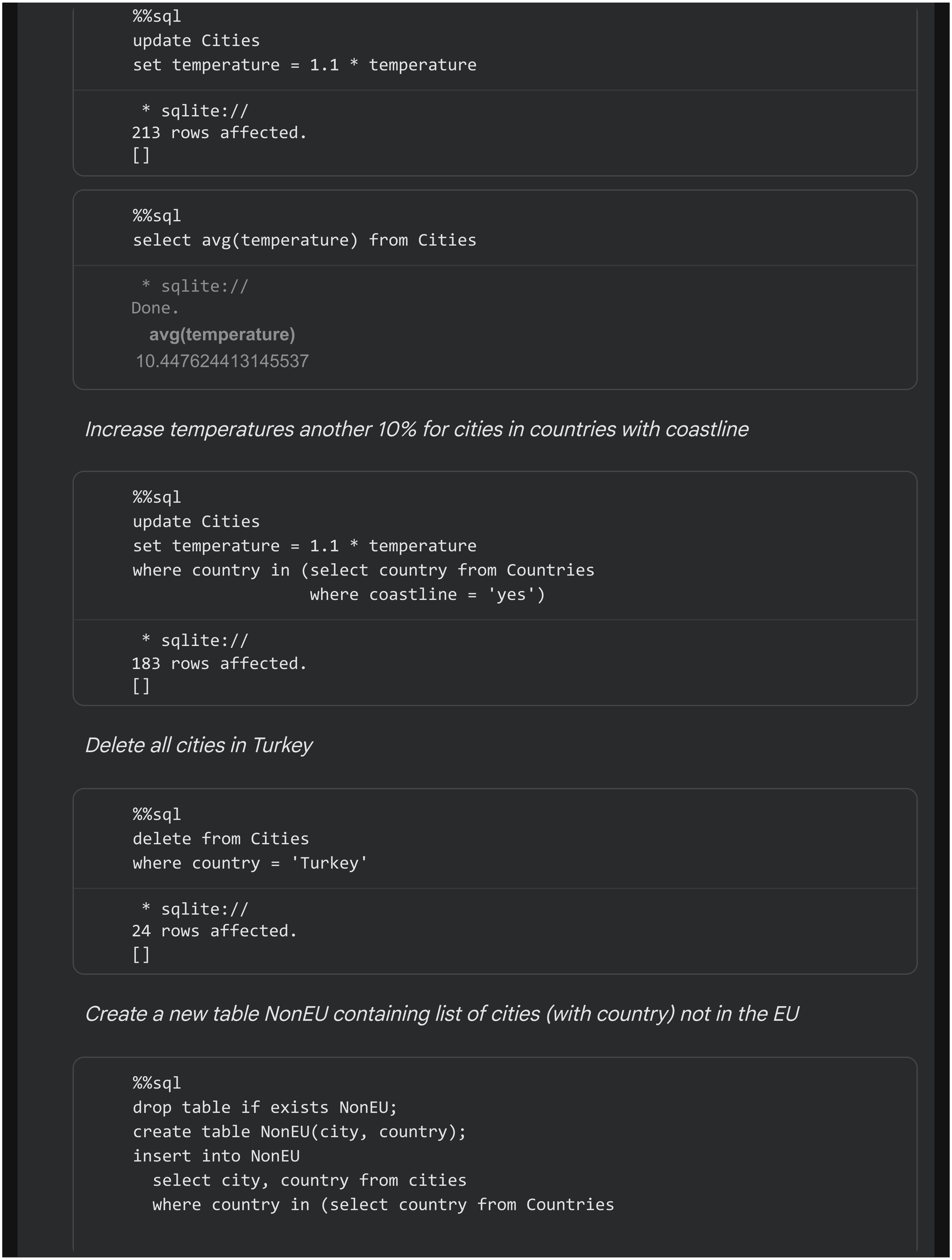
Same query using subquery in From clause instead of Select clause

```
%%sql
select Cold.country, numcold, numwarm
from (select country, count() as numcold from Cities
      where temperature < 9 group by country) Cold,
     (select country, count() as numwarm from Cities
      where temperature > 14 group by country) Warm
where Cold.country = Warm.country
```

```
 * sqlite://
Done.
country numcold numwarm
France  5       1
Italy   1       7
Turkey  4       5
```

Data modification

Increase all city temperatures by 10%



```
%%sql
update Cities
set temperature = 1.1 * temperature
```

```
 * sqlite://
213 rows affected.
[]
```

```
%%sql
select avg(temperature) from Cities
```

```
 * sqlite://
Done.
```

avg(temperature)
10.447624413145537

Increase temperatures another 10% for cities in countries with coastline

```
%%sql
update Cities
set temperature = 1.1 * temperature
where country in (select country from Countries
                  where coastline = 'yes')
```

```
 * sqlite://
183 rows affected.
[]
```

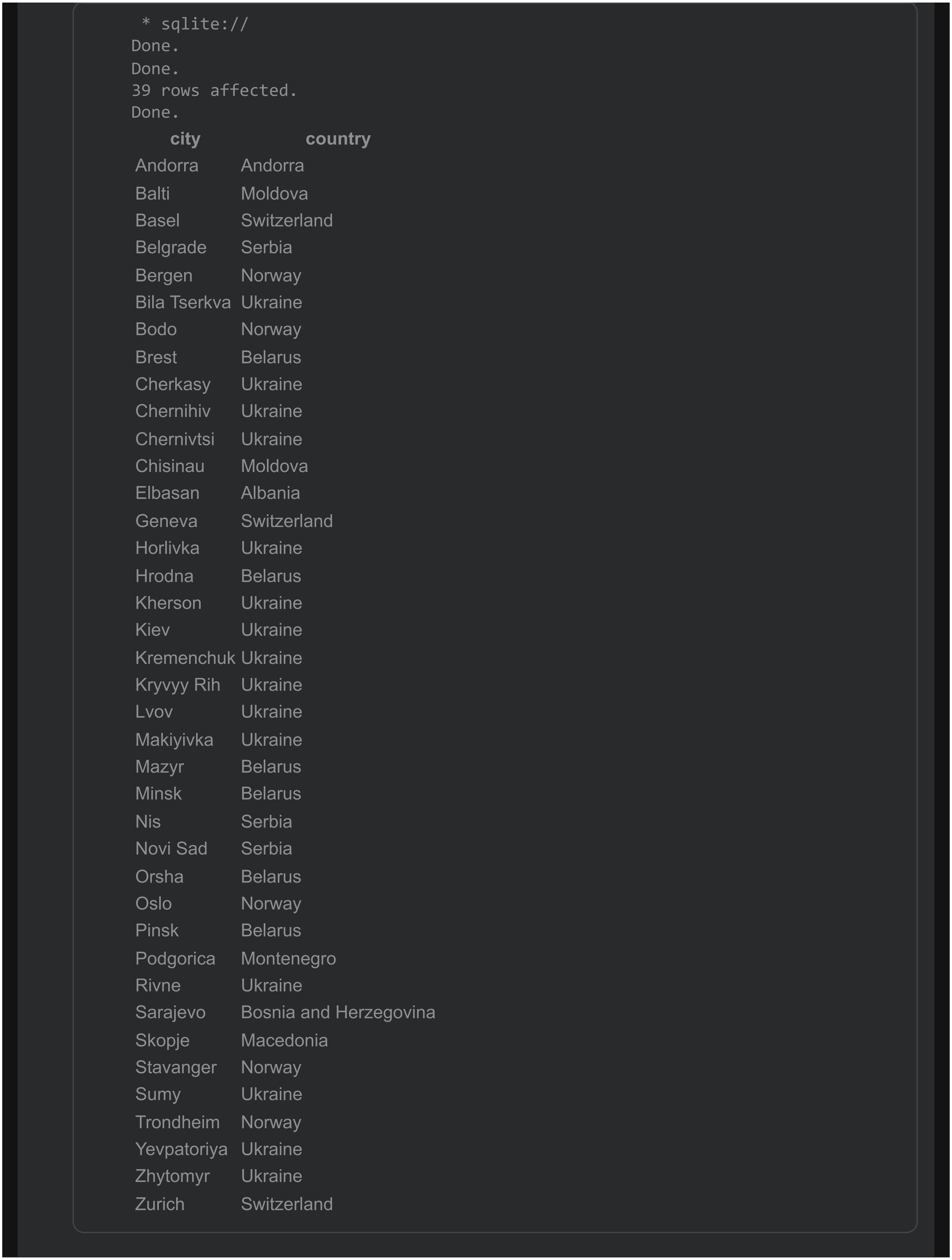
Delete all cities in Turkey

```
%%sql
delete from Cities
where country = 'Turkey'
```

```
 * sqlite://
24 rows affected.
[]
```

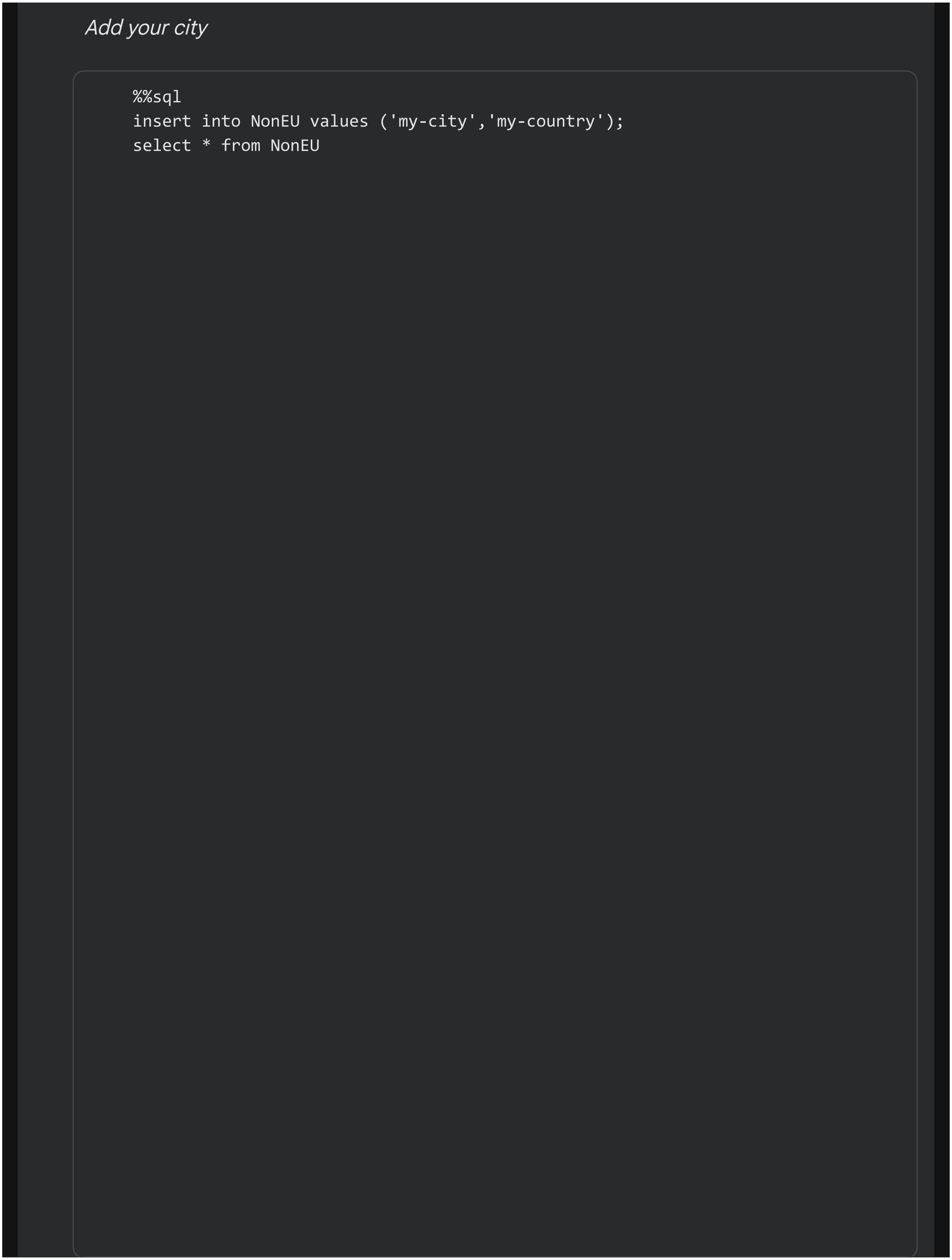
Create a new table NonEU containing list of cities (with country) not in the EU

```
%%sql
drop table if exists NonEU;
create table NonEU(city, country);
insert into NonEU
  select city, country from cities
  where country in (select country from Countries
                    where EU = 'no');
select * from NonEU
```



```
 * sqlite://
Done.
Done.
39 rows affected.
Done.
```

city	country
Andorra	Andorra
Balti	Moldova
Basel	Switzerland
Belgrade	Serbia
Bergen	Norway
Bila Tserkva	Ukraine
Bodo	Norway
Brest	Belarus
Cherkasy	Ukraine
Chernihiv	Ukraine
Chernivtsi	Ukraine
Chisinau	Moldova
Elbasan	Albania
Geneva	Switzerland
Horlivka	Ukraine
Hrodna	Belarus
Kherson	Ukraine
Kiev	Ukraine
Kremenchuk	Ukraine
Kryvyy Rih	Ukraine
Lvov	Ukraine
Makiyivka	Ukraine
Mazyr	Belarus
Minsk	Belarus
Nis	Serbia
Novi Sad	Serbia
Orsha	Belarus
Oslo	Norway
Pinsk	Belarus
Podgorica	Montenegro
Rivne	Ukraine
Sarajevo	Bosnia and Herzegovina
Skopje	Macedonia
Stavanger	Norway
Sumy	Ukraine
Trondheim	Norway
Yevpatoriya	Ukraine
Zhytomyr	Ukraine
Zurich	Switzerland



Add your city

```
%%sql
insert into NonEU values ('my-city','my-country');
select * from NonEU
```