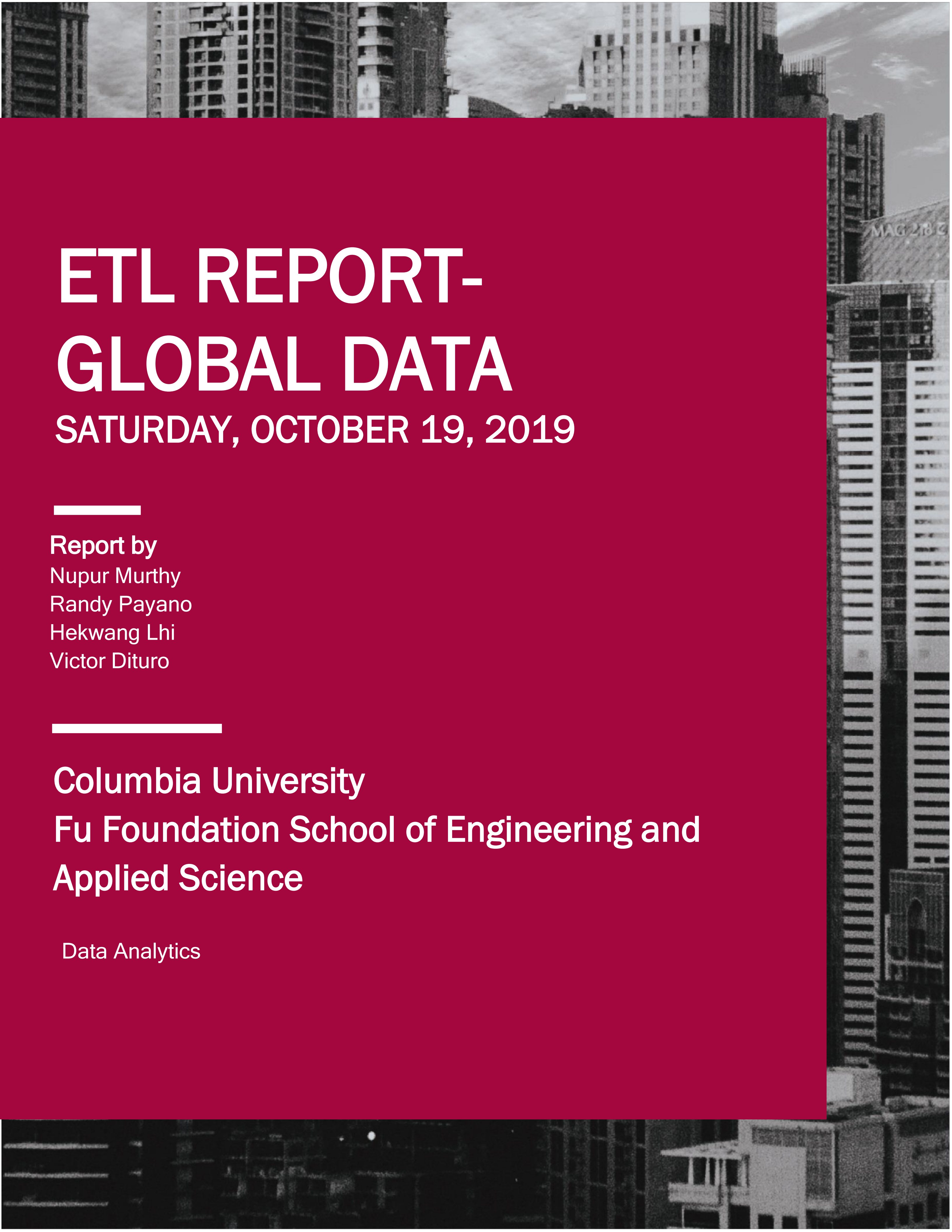


ETL REPORT- GLOBAL DATA

SATURDAY, OCTOBER 19, 2019

Report by

Nupur Murthy
Randy Payano
Hekwang Lhi
Victor Dituro

Columbia University
Fu Foundation School of Engineering and Applied Science

Data Analytics

Executive Summary

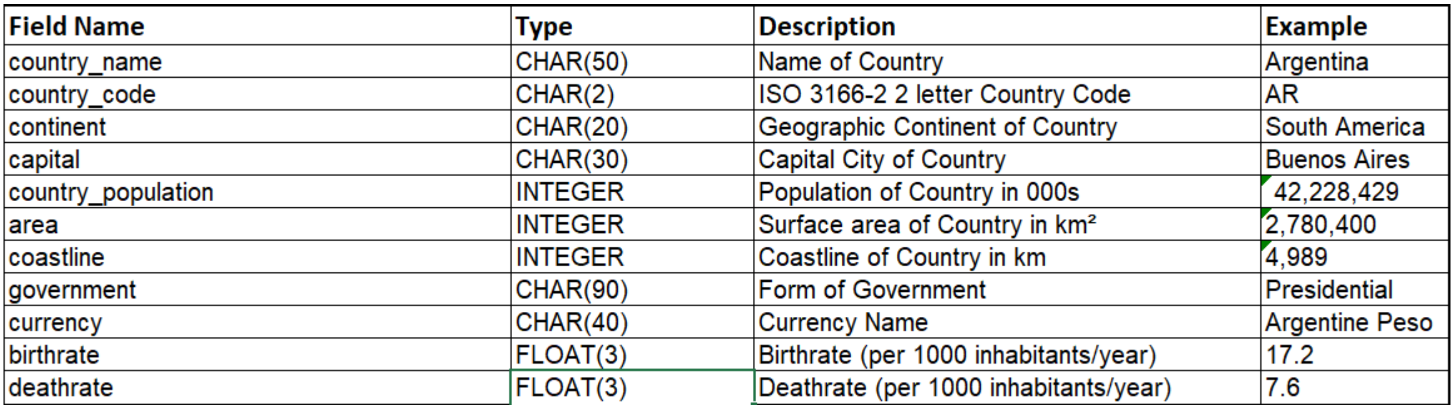
The objective of our project is to obtain datasets referencing global country information and global city information for cities with more than one million inhabitants. Using these datasets we are able to design queries to compare statistics on major cities all over the world.

DATA SOURCES

We extracted our data from the World Data site. We found a dataset which had a variety of statistics on a national level as well as a dataset with statistics on megacities across the globe. We downloaded the csv files from this url: https://www.worlddata.info/downloads/ and cleaned the data on Python using pandas. We verified the credibility of our data by creating graphs using matplotlib. The specific datasets we used were *countries.csv* and *megacities.csv* which can be found at the URL mentioned above. matplotlib. The specific datasets we used were *countries.csv* and *megacities.csv* which can be found at the URL mentioned above. Using these datasets, we dropped null values, unnecessary columns, redundant columns, bad data, etc., as well as renamed our columns in order to create clean datasets.

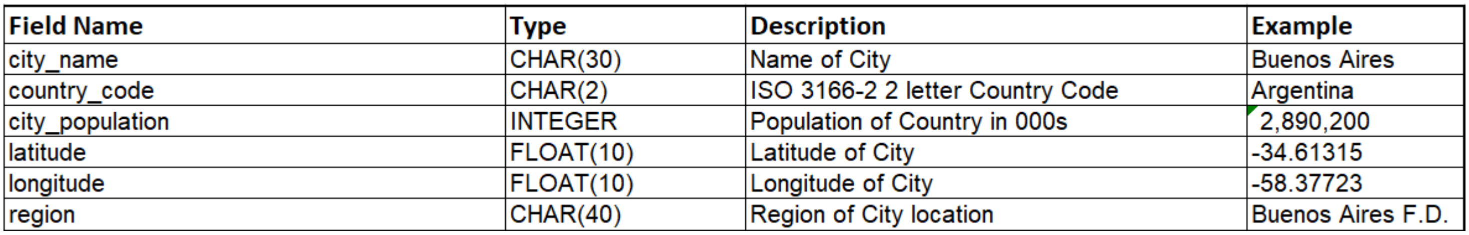
DATA DICTIONARIES

Countries: a dataset of all the countries defined by the ISO standards committee



Field Name	Type	Description	Example
country_name	CHAR(50)	Name of Country	Argentina
country_code	CHAR(2)	ISO 3166-2 2 letter Country Code	AR
continent	CHAR(20)	Geographic Continent of Country	South America
capital	CHAR(30)	Capital City of Country	Buenos Aires
country_population	INTEGER	Population of Country in 000s	42,228,429
area	INTEGER	Surface area of Country in km²	2,780,400
coastline	INTEGER	Coastline of Country in km	4,989
government	CHAR(90)	Form of Government	Presidential
currency	CHAR(40)	Currency Name	Argentine Peso
birthrate	FLOAT(3)	Birthrate (per 1000 inhabitants/year)	17.2
deathrate	FLOAT(3)	Deathrate (per 1000 inhabitants/year)	7.6

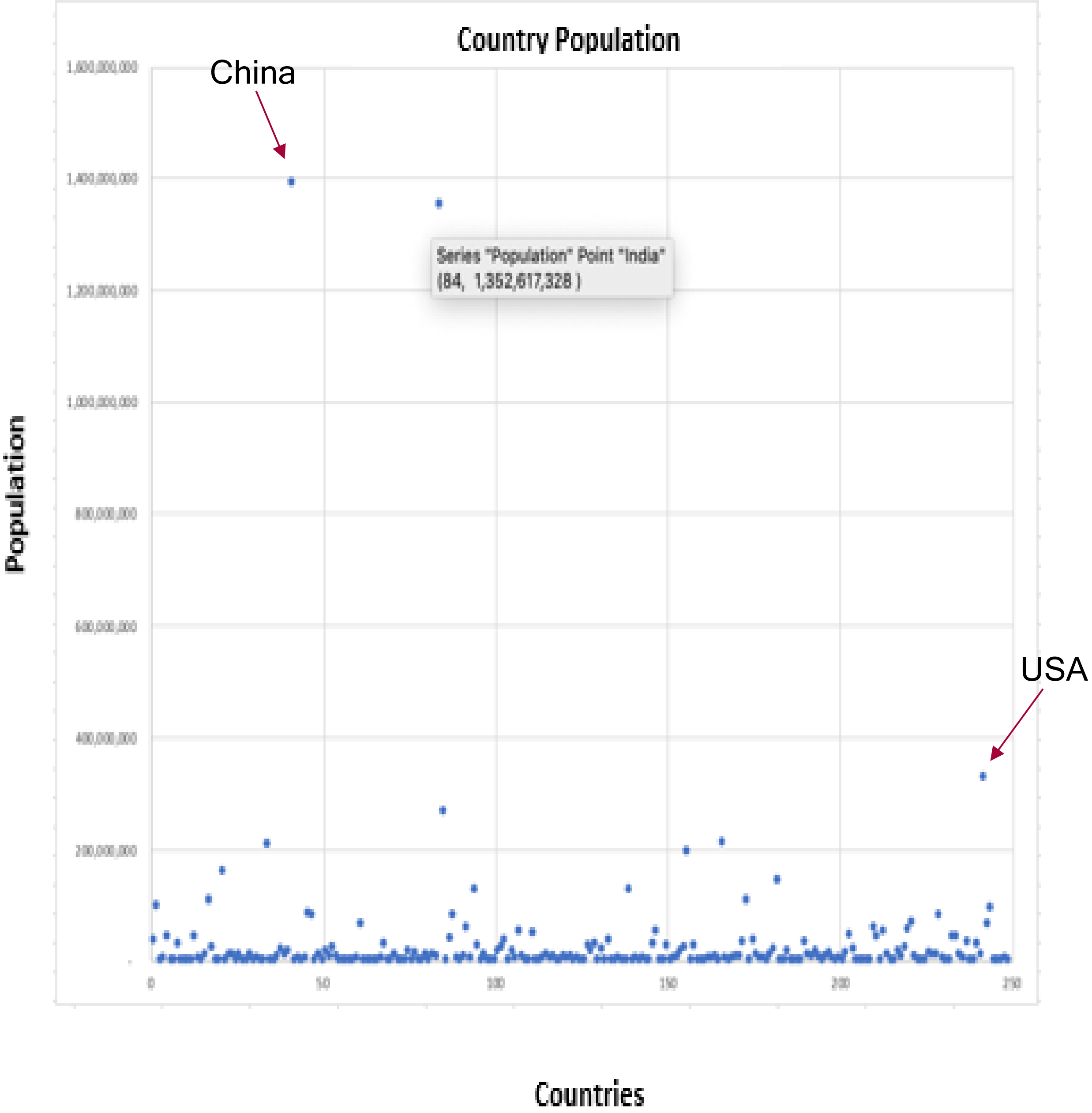
Megacities: a dataset containing the cities of each country with a population greater than 1 million



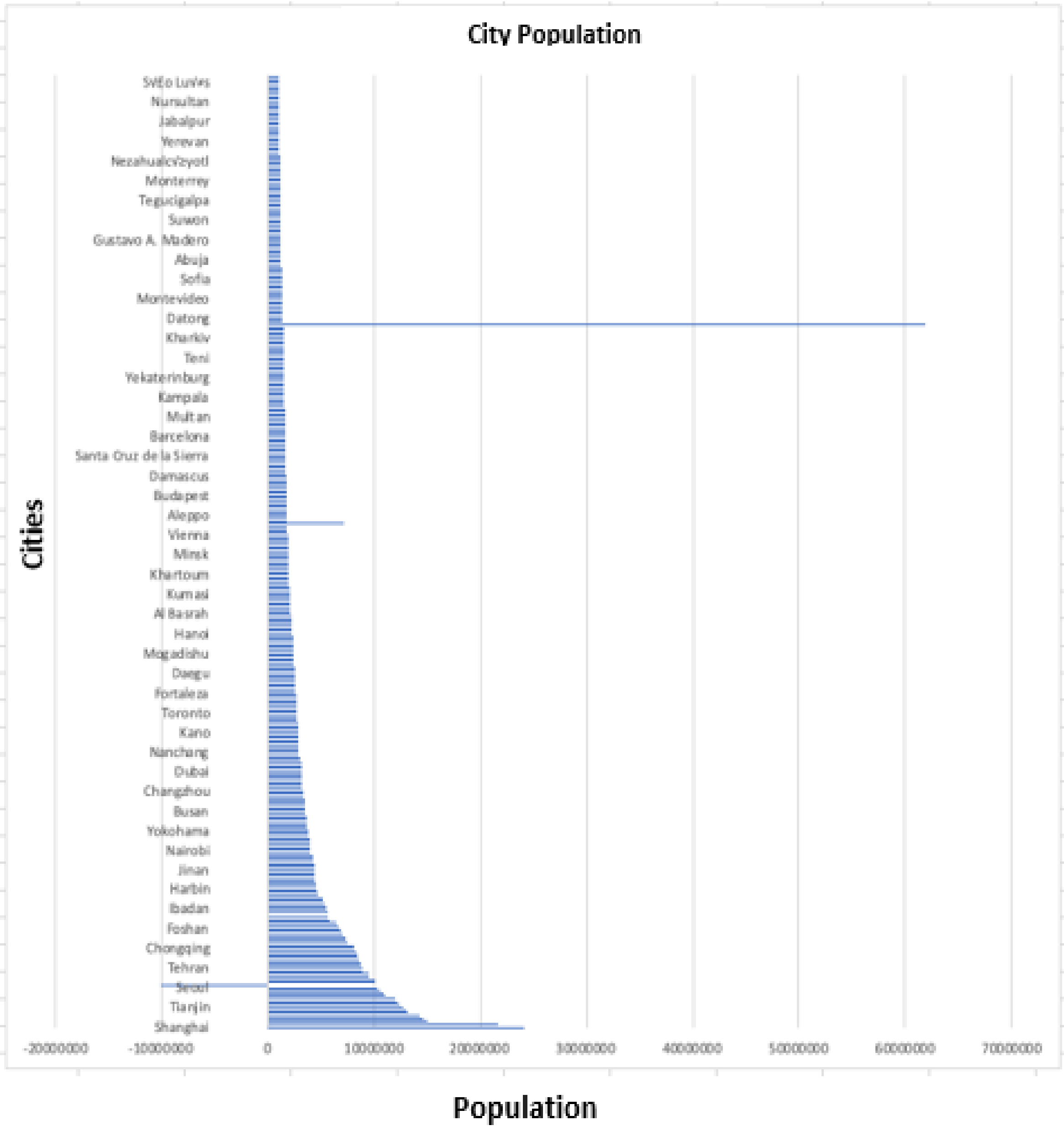
Field Name	Type	Description	Example
city_name	CHAR(30)	Name of City	Buenos Aires
country_code	CHAR(2)	ISO 3166-2 2 letter Country Code	Argentina
city_population	INTEGER	Population of Country in 000s	2,890,200
latitude	FLOAT(10)	Latitude of City	-34.61315
longitude	FLOAT(10)	Longitude of City	-58.37723
region	CHAR(40)	Region of City location	Buenos Aires F.D.

GRAPHS

This dataset shows us that our country population data is consistent. For example, it shows that China and India have the highest populations and USA has the third highest- which is true.

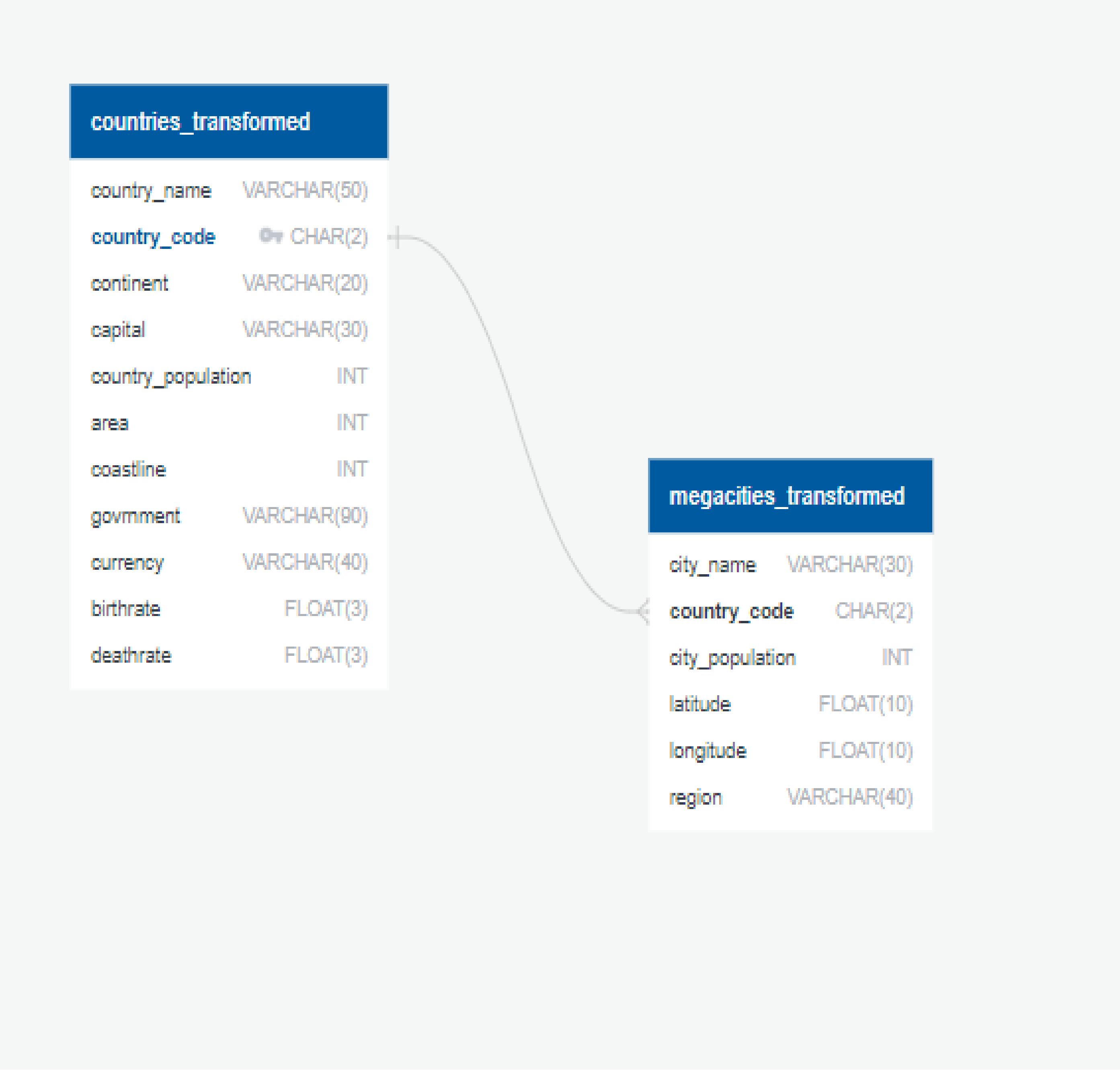


This dataset is inconsistent. As shown in the graph, the data claims that one city has a population of over 60,000,000 people. The data also shows that another city has a negative population. Both these statistics are clearly false. We made sure to rid of any data which was inconsistent so we are able to make accurate observations in future analysis.



ER DIAGRAM

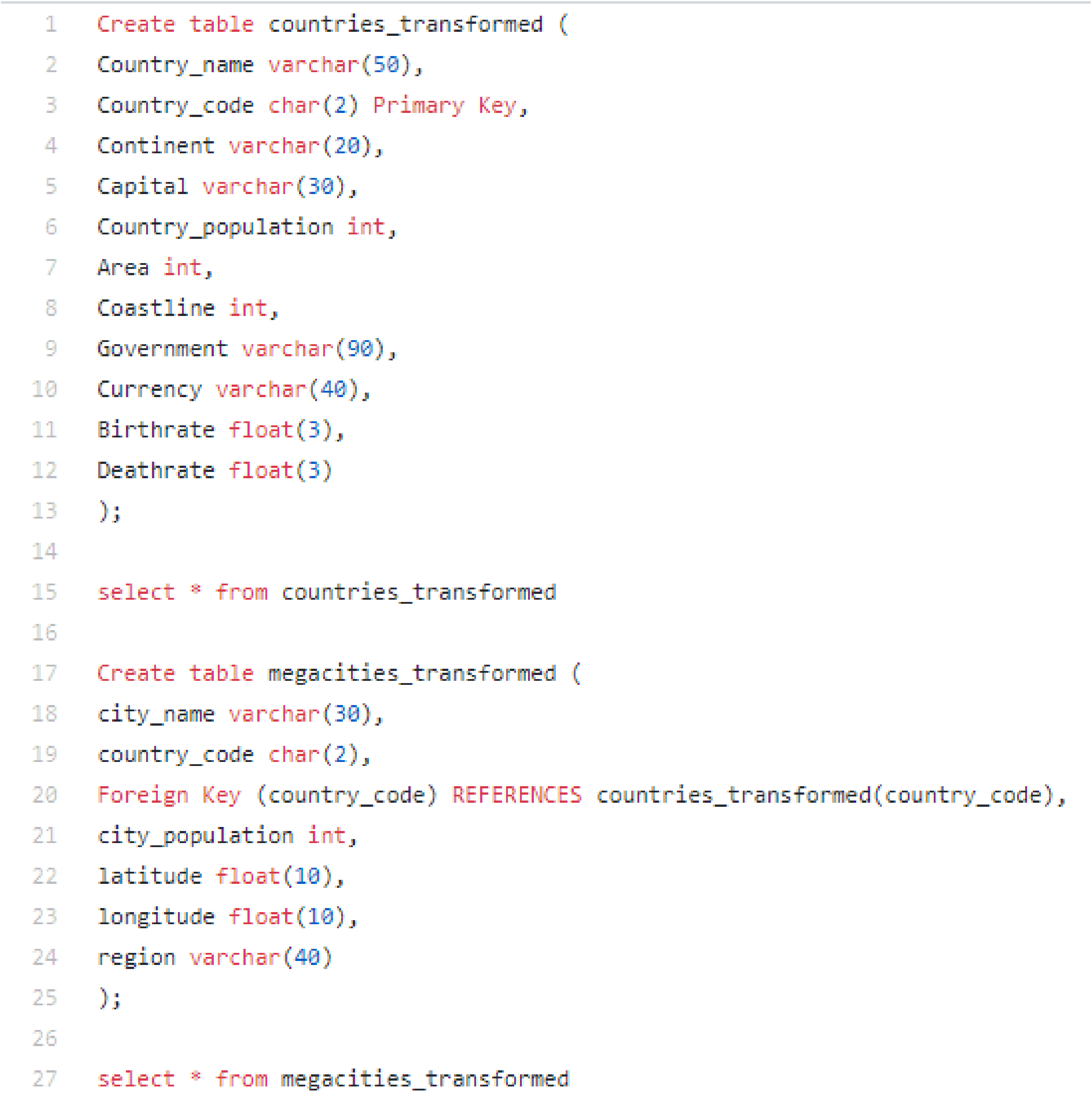
As shown below, there is a relationship between the datasets by *country_code.*



countries_transformed	
country_name	VARCHAR(50)
country_code	🔑 CHAR(2)
continent	VARCHAR(20)
capital	VARCHAR(30)
country_population	INT
area	INT
coastline	INT
govrnment	VARCHAR(90)
currency	VARCHAR(40)
birthrate	FLOAT(3)
deathrate	FLOAT(3)

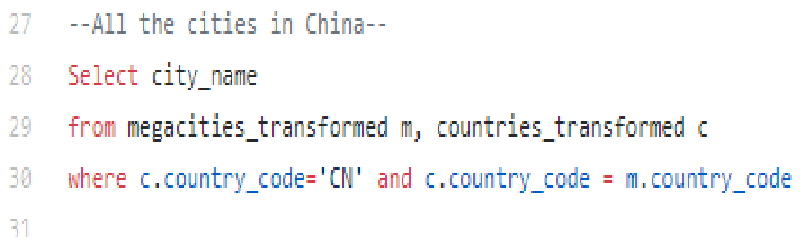
megacities_transformed	
city_name	VARCHAR(30)
country_code	CHAR(2)
city_population	INT
latitude	FLOAT(10)
longitude	FLOAT(10)
region	VARCHAR(40)

TABLE SCHEMA

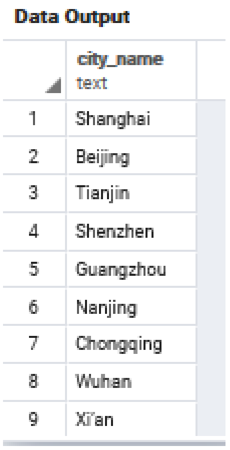


```
Create table countries_transformed (
Country_name varchar(50),
Country_code char(2) Primary Key,
Continent varchar(20),
Capital varchar(30),
Country_population int,
Area int,
Coastline int,
Government varchar(90),
Currency varchar(40),
Birthrate float(3),
Deathrate float(3)
);

select * from countries_transformed

Create table megacities_transformed (
city_name varchar(30),
country_code char(2),
Foreign Key (country_code) REFERENCES countries_transformed(country_code),
city_population int,
latitude float(10),
longitude float(10),
region varchar(40)
);

select * from megacities_transformed
```

QUERIES

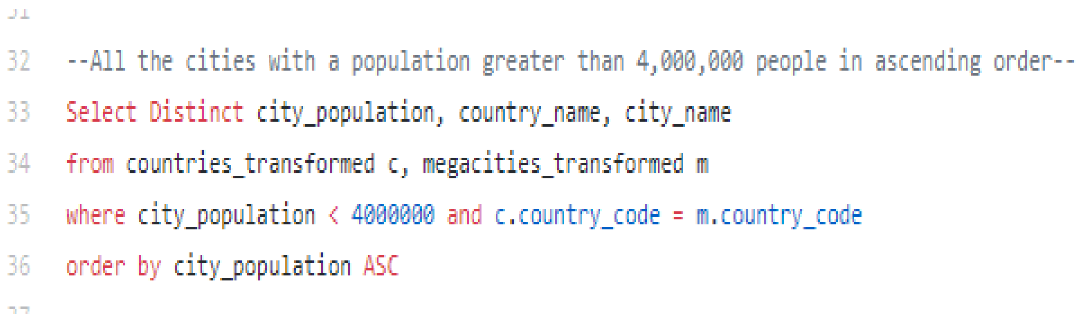


```
--All the cities in China--
Select city_name
from megacities_transformed m, countries_transformed c
where c.country_code='CN' and c.country_code = m.country_code
```

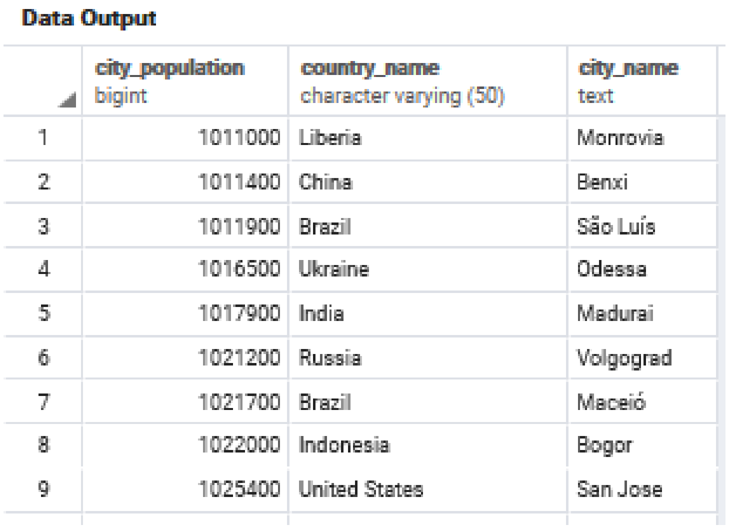


Data Output

	city_name text
1	Shanghai
2	Beijing
3	Tianjin
4	Shenzhen
5	Guangzhou
6	Nanjing
7	Chongqing
8	Wuhan
9	Xi'an

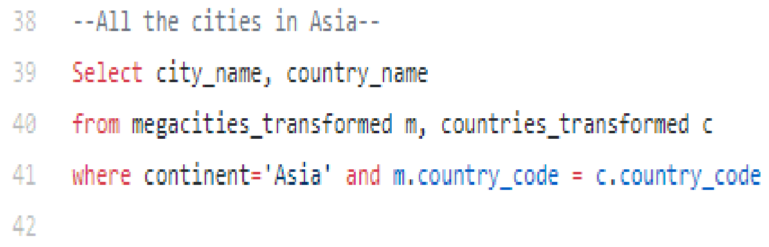


```
--All the cities with a population greater than 4,000,000 people in ascending order--
Select Distinct city_population, country_name, city_name
from countries_transformed c, megacities_transformed m
where city_population < 4000000 and c.country_code = m.country_code
order by city_population ASC
```

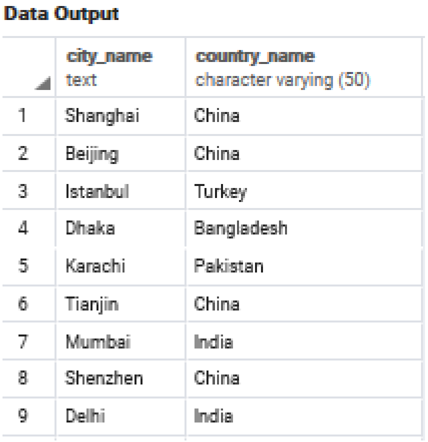


Data Output

	city_population bigint	country_name character varying (50)	city_name text
1	1011000	Liberia	Monrovia
2	1011400	China	Benxi
3	1011900	Brazil	São Luís
4	1016500	Ukraine	Odessa
5	1017900	India	Madurai
6	1021200	Russia	Volgograd
7	1021700	Brazil	Maceió
8	1022000	Indonesia	Bogor
9	1025400	United States	San Jose

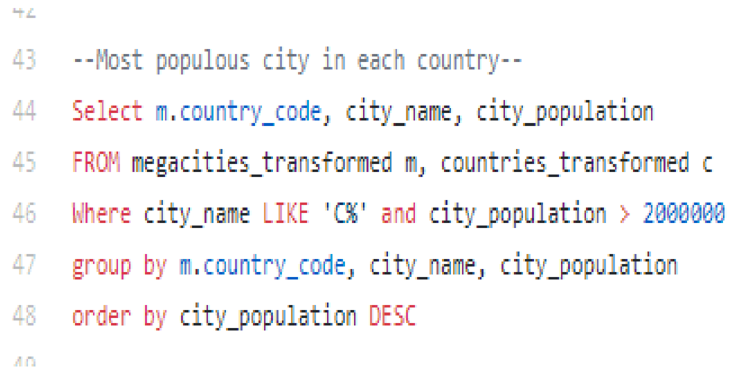


```
--All the cities in Asia--
Select city_name, country_name
from megacities_transformed m, countries_transformed c
where continent='Asia' and m.country_code = c.country_code
```

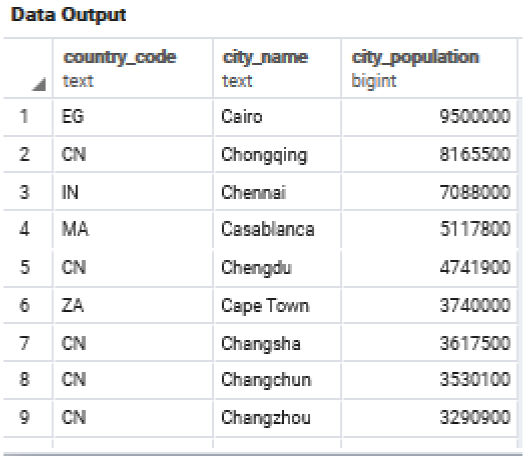


Data Output

	city_name text	country_name character varying (50)
1	Shanghai	China
2	Beijing	China
3	Istanbul	Turkey
4	Dhaka	Bangladesh
5	Karachi	Pakistan
6	Tianjin	China
7	Mumbai	India
8	Shenzhen	China
9	Delhi	India

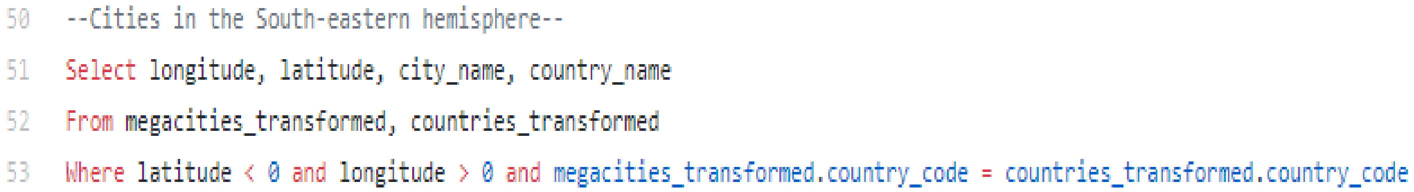


```
--Most populous city in each country--
Select m.country_code, city_name, city_population
FROM megacities_transformed m, countries_transformed c
Where city_name LIKE 'C%' and city_population > 2000000
group by m.country_code, city_name, city_population
order by city_population DESC
```

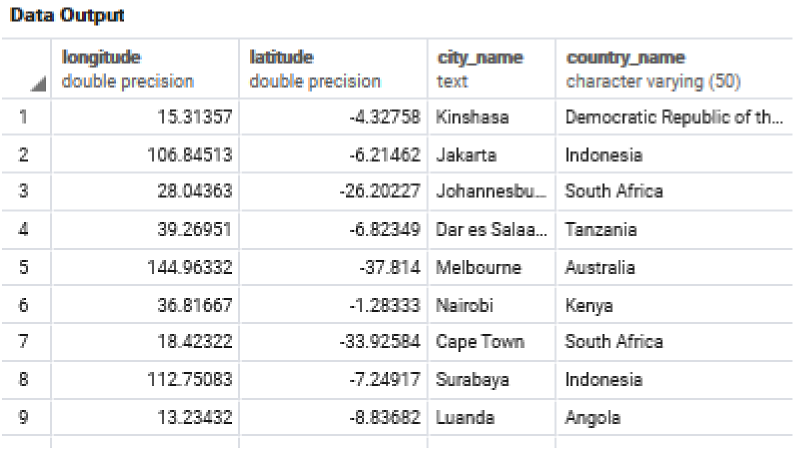


Data Output

	country_code text	city_name text	city_population bigint
1	EG	Cairo	9500000
2	CN	Chongqing	8165500
3	IN	Chennai	7088000
4	MA	Casablanca	5117800
5	CN	Chengdu	4741900
6	ZA	Cape Town	3740000
7	CN	Changsha	3617500
8	CN	Changchun	3530100
9	CN	Changzhou	3290900



```
--Cities in the South-eastern hemisphere--
Select longitude, latitude, city_name, country_name
From megacities_transformed, countries_transformed
Where latitude < 0 and longitude > 0 and megacities_transformed.country_code = countries_transformed.country_code
```



Data Output

	longitude double precision	latitude double precision	city_name text	country_name character varying (50)
1	15.31357	-4.32758	Kinshasa	Democratic Republic of th...
2	106.84513	-6.21462	Jakarta	Indonesia
3	28.04363	-26.20227	Johannesbu...	South Africa
4	39.26951	-6.82349	Dar es Salaa...	Tanzania
5	144.96332	-37.814	Melbourne	Australia
6	36.81667	-1.28333	Nairobi	Kenya
7	18.42322	-33.92584	Cape Town	South Africa
8	112.75083	-7.24917	Surabaya	Indonesia
9	13.23432	-8.83682	Luanda	Angola