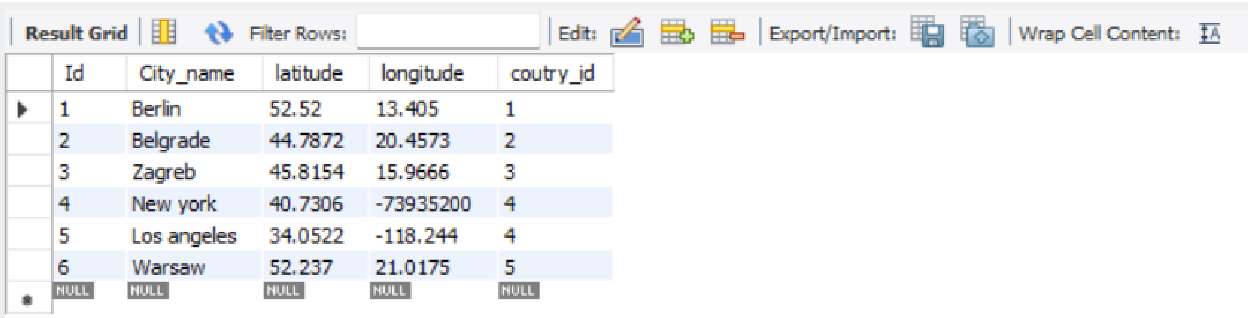
Assessment

1. **Consider three table named as city, customer and country.**

➔ **City table:**



Id	City_name	latitude	longitude	coutry_id
1	Berlin	52.52	13.405	1
2	Belgrade	44.7872	20.4573	2
3	Zagreb	45.8154	15.9666	3
4	New york	40.7306	-73935200	4
5	Los angeles	34.0522	-118.244	4
6	Warsaw	52.237	21.0175	5
NULL	NULL	NULL	NULL	NULL

➔ **City table query:** create database Assessment;

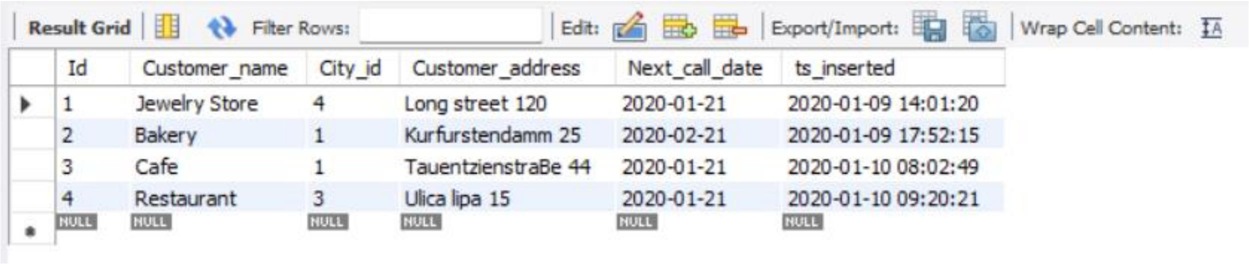
```
use Assessment;

 create table city (
 Id int primary key auto_increment,
 City_name varchar(30) not null,
 latitude float not null,
 longitude float not null,
 coutry_id int);

INSERT INTO city (City_name, latitude, longitude, coutry_id)
VALUES ('Berlin', 52.520008, 13.404954, 1),
('Belgrade', 44.787197, 20.457273, 2),
('Zagreb', 45.815399, 15.966568, 3),
('New york', 40.730610, -73935242, 4),
('Los angeles', 34.052235, -118.243683, 4),
('Warsaw', 52.237049, 21.017532, 5);

select * from city;
```

➔ **Customer table:**



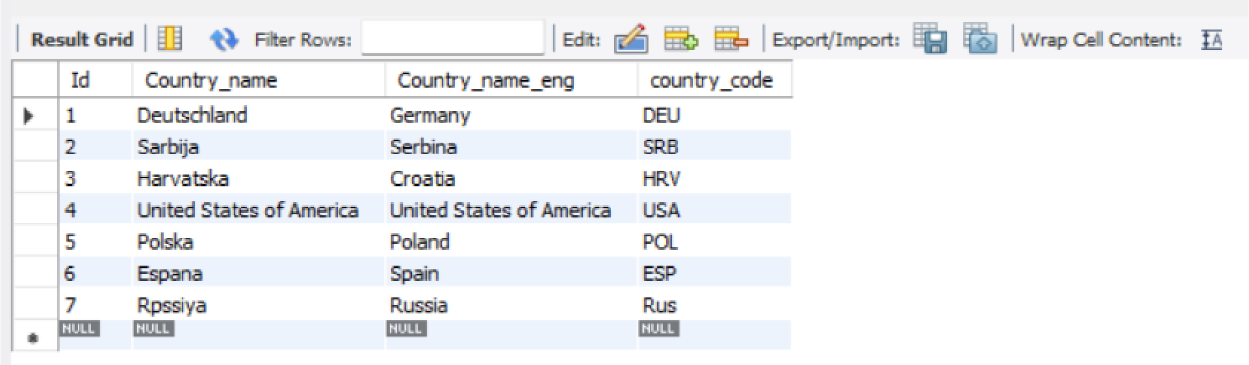
Id	Customer_name	City_id	Customer_address	Next_call_date	ts_inserted
1	Jewelry Store	4	Long street 120	2020-01-21	2020-01-09 14:01:20
2	Bakery	1	Kurfurstendamm 25	2020-02-21	2020-01-09 17:52:15
3	Cafe	1	TauentzienstraBe 44	2020-01-21	2020-01-10 08:02:49
4	Restaurant	3	Ulica lipa 15	2020-01-21	2020-01-10 09:20:21
NULL	NULL	NULL	NULL	NULL	NULL

➔ **Customer table query:** create table customer (

```
Id int primary key auto_increment,
Customer_name varchar(30) not null,
City_id int not null,
Customer_address varchar(30),
Next_call_date date,
ts_inserted datetime);

insert into customer (Customer_name, City_id, Customer_address, Next_call_date, ts_inserted)
value
('Jewelry Store', 4, 'Long street 120', '2020-01-21', '2020-01-09 14:01:20'),
('Bakery', 1, 'Kurfurstendamm 25', '2020-02-21', '2020-01-09 17:52:15'),
('Cafe', 1, 'TauentzienstraBe 44', '2020-01-21', '2020-01-10 08:02:49'),
('Restaurant', 3, 'Ulica lipa 15', '2020-01-21', '2020-01-10 09:20:21');
```

➔ **Country table:**



	Id	Country_name	Country_name_eng	country_code
▶	1	Deutschland	Germany	DEU
	2	Sarbija	Serbina	SRB
	3	Harvatska	Croatia	HRV
	4	United States of America	United States of America	USA
	5	Polska	Poland	POL
	6	Espana	Spain	ESP
	7	Rpssiya	Russia	Rus
*	NULL	NULL	NULL	NULL

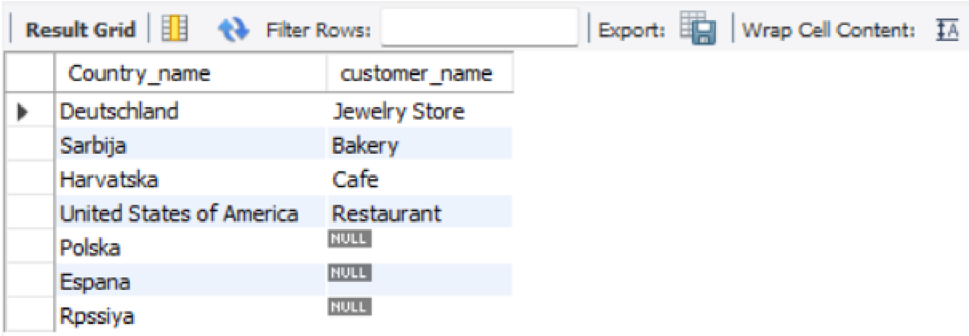
➔ **Country table query:** create table Country (

```
Id int primary key auto_increment,
Country_name varchar(30) not null,
Country_name_eng varchar(30) not null,
country_code varchar(30) not null);

insert into Country (Country_name, Country_name_eng, country_code) values
('Deutschland', 'Germany', 'DEU'),
('Sarbija', 'Serbina', 'SRB'),
('Harvatska', 'Croatia', 'HRV'),
('United States of America', 'United States of America', 'USA'),
('Polska', 'Poland', 'POL'),
('Espana', 'Spain', 'ESP'),
('Rpssiya', 'Russia', 'Rus');

select * from Country;
```

➔ **List all Countries and customers related to these countries.**

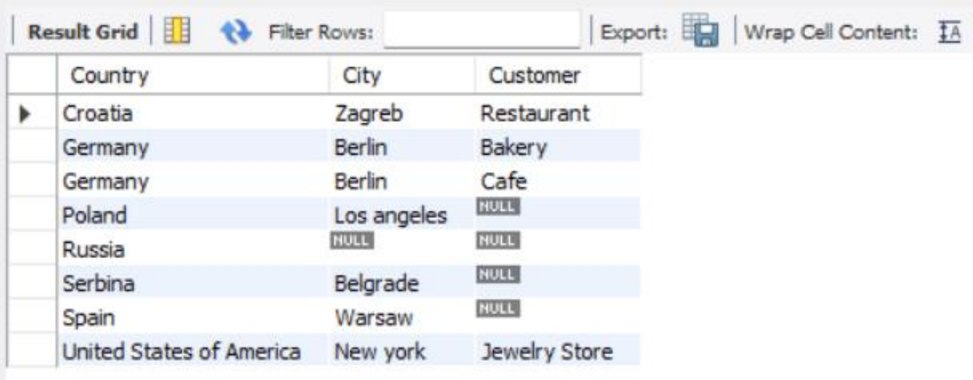


Country_name	customer_name
Deutschland	Jewelry Store
Sarbija	Bakery
Harvatska	Cafe
United States of America	Restaurant
Polska	NULL
Espana	NULL
Rpssiya	NULL

➔ **Query:** select * from country join customer;

select Country_name, customer_name from country left join customer on country.id = customer.id;

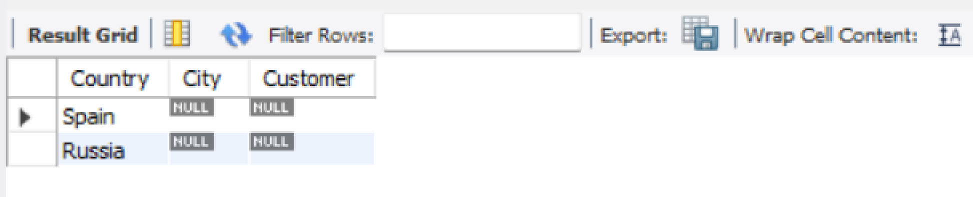
➔ **For each country displaying its name in English, the name of the city customer is located in as well as the name of the customer.**



Country	City	Customer
Croatia	Zagreb	Restaurant
Germany	Berlin	Bakery
Germany	Berlin	Cafe
Poland	Los angeles	NULL
Russia	NULL	NULL
Serbina	Belgrade	NULL
Spain	Warsaw	NULL
United States of America	New york	Jewelry Store

➔ **Query:** SELECT
country.country_name_eng AS Country,
city.city_name AS City,
customer.Customer_name AS Customer
FROM country
LEFT JOIN city ON country.id = city.id
LEFT JOIN customer ON city.id = customer.City_id
ORDER BY country.country_name_eng, city.city_name, customer.Customer_name;

➔ **Return even countries without related cities and customers.**



Country	City	Customer
Spain	NULL	NULL
Russia	NULL	NULL

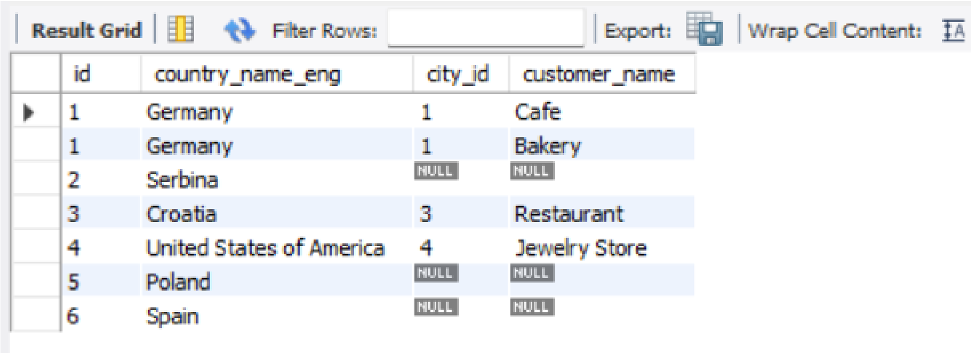
➔ **Query:** SELECT

```
    country.country_name_eng AS Country,
    city.city_name AS City,
    customer.Customer_name AS Customer
FROM
    country
LEFT JOIN city ON country.id = city.coutry_id
LEFT JOIN customer ON city.id = customer.City_id
where city.id is null and customer.id is null;
```

➔ **TASK : 2**

➔ **Return the list of all countries that have pairs(exclude countries which are not referenced by any city). For such pairs return all customers.**

➔

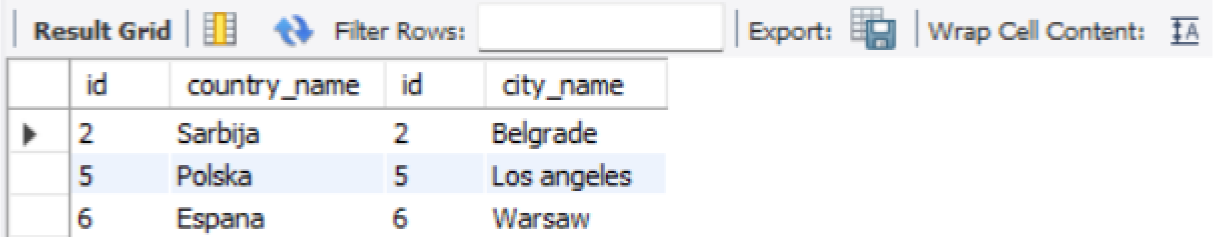


id	country_name_eng	city_id	customer_name
1	Germany	1	Cafe
1	Germany	1	Bakery
2	Serbina	NULL	NULL
3	Croatia	3	Restaurant
4	United States of America	4	Jewelry Store
5	Poland	NULL	NULL
6	Spain	NULL	NULL

➔ **Query:**

```
SELECT country.id, country.country_name_eng, customer.city_id, customer.customer_name
FROM country
INNER JOIN city ON country.id = city.id
left JOIN customer ON city.id = customer.city_id;
```

➔ **Return even pairs of not having a single customer:**



id	country_name	id	city_name
2	Sarbija	2	Belgrade
5	Polska	5	Los angeles
6	Espana	6	Warsaw

➔ **Query:**

```
select country.id,country_name,city.id,city.city_name from country
inner join city on country.id = city.id
left join customer on city.id = customer.city_id
where customer.id is null;
```