



## DataBase - SQL query

### \*\*\* Assessment \*\*\*

Qua.1 -> Write SQL query to solve the problem given below

Consider three table named as city, customer and country

The city table is given below :

id	city_name	lat	long	country_id
1	Berlin	52.520008	13.404954	1
2	Belgrade	44.787197	20.457273	2
3	Zagreb	45.815399	15.966568	3
4	New York	40.730610	-73.935242	4
5	Los Angeles	34.052235	-118.243683	4
6	Warsaw	52.237049	21.017532	5

The 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 14:01:20.000
2	Bakery	1	Kurfürstendamm 25	2020-02-21	2020-01-09 17:52:15.000
3	Café	1	Tauentzienstraße 44	2020-01-21	2020-01-10 08:02:49.000
4	Restaurant	3	Ulica lipa 15	2020-01-21	2020-01-10 09:20:21.000

The Country table :

id	country_name	country_name_eng	country_code
1	Deutschland	Germany	DEU
2	Srbija	Serbia	SRB
3	Hrvatska	Croatia	HRV
4	United States of America	United States of America	USA
5	Polska	Poland	POL
6	España	Spain	ESP
7	Rossiya	Russia	RUS

**Note :**

- While each city has a related country, not all countries have related cities (Spain & Russia don't have them)

- Same stands for the customers. Each customer has the city\_id value defined, but only 3 cities are being used (Berlin, Zagreb & New York)

Now commute the following tasks :

**Task : 1 (join multiple tables using left join)**

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

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.

Return even countries without related cities and customers.

**Task : 2 (join multiple tables using both left and inner join)**

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

Return even pairs of not having a single customer

Make sure to make your code clean neat

Ans.  
TABLE 1  
city

```
CREATE TABLE city
(
    id INT PRIMARY KEY,
    city_name VARCHAR(100),
    lat DECIMAL(10, 7),
    lon DECIMAL(10, 7),
    country_id INT
);
INSERT INTO city VALUES
(1, 'Berlin', 52.520008, 13.404954, 1),
(2, 'Belgrade', 44.787197, 20.457273, 2),
(3, 'Zagreb', 45.815399, 15.966568, 3),
(4, 'New York', 40.730610, -73.935242, 4),
(5, 'Los Angeles', 34.052235, -118.243683, 4),
(6, 'Warsaw', 52.237049, 21.017532, 5);
```

id	city_name	lat	lon	country_id
1	Berlin	52.5200080	13.4049540	1
2	Belgrade	44.7871970	20.4572730	2
3	Zagreb	45.8153990	15.9665680	3
4	New York	40.7306100	-73.9352420	4
5	Los Angeles	34.0522350	-118.2436830	4
6	Warsaw	52.2370490	21.0175320	5

TABLE 2  
customer

```
CREATE TABLE customer
(
    id INT PRIMARY KEY,
    customer name VARCHAR(100),
    city_id INT
);
INSERT INTO customer VALUES
(1, 'Jewelry Store', 4),
(2, 'Bakery', 1),
(3, 'Café', 1),
(4, 'Restaurant', 3);
```

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	2020-01-10	2020-01-10 08:02:49
4	Restaurant	3	Ulica lipa 15	2020-01-21	2020-01-10 09:20:21

TABLE 3  
country

```
CREATE TABLE country
(
    id INT PRIMARY KEY,
    country_name VARCHAR(100)
);
INSERT INTO country VALUES
(1, 'Germany'),
(2, 'Serbia'),
(3, 'Croatia'),
(4, 'United States'),
(5, 'Poland'),
(6, 'Spain'),
(7, 'Russia');
```

id	country_name	country_name_eng	country_code
1	Deutschland	Germany	DEU
2	Srbija	Serbia	SRB
3	Hrvatska	Croatia	HRV
4	United States of America	United States of America	USA
5	Polska	Poland	POL
6	España	Spain	ESP
7	Rossiya	Russia	RUS

## Task : 1

**(join multiple tables using left join)**

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

**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.**

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

```
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.country\_id  
LEFT JOIN customer ON city.id = customer.city\_id;

country	city	customer
United States of America	New York	Jewelry Store
Germany	Berlin	Bakery
Germany	Berlin	Cafe
Croatia	Zagreb	Restaurant
Serbia	Belgrade	NULL
United States of America	Los Angeles	NULL
Poland	Warsaw	NULL
Spain	NULL	NULL
Russia	NULL	NULL

## Task : 2

(join multiple tables using both left and inner join)

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

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

**Make sure to make your code clean neat**

```
SELECT
    country.country_name_eng AS country,
    city.city_name AS city,
    customer.customer_name AS customer
FROM
    country
INNER JOIN city ON country.id = city.country_id
LEFT JOIN customer ON city.id = customer.city_id;
```

country	city	customer
United States of America	New York	Jewelry Store
Germany	Berlin	Bakery
Germany	Berlin	Cafe
Croatia	Zagreb	Restaurant
Serbia	Belgrade	NULL
United States of America	Los Angeles	NULL
Poland	Warsaw	NULL