

## Name : Dhruvit Savaliya

- Write SQL query to solve the problem given below

➤ Country table :

```
CREATE TABLE Countries (  
    id INT PRIMARY KEY,  
    country_name VARCHAR(100) NOT NULL,  
    country_name_eng VARCHAR(100) NOT NULL,  
    country_code CHAR(3) NOT NULL  
);
```

```
INSERT INTO Countries (id, country_name, country_name_eng, country_code)  
VALUES  
(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');
```

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

➤ city table :

```
CREATE TABLE Cities (  
  id INT PRIMARY KEY,  
  city_name VARCHAR(50) NOT NULL,  
  lat DECIMAL(10, 6),  
  long1 DECIMAL(10, 6),  
  country_id INT,  
  FOREIGN KEY(country_id) REFERENCES countries(id)  
);
```

```
INSERT INTO Cities (id, city_name, lat, long1, country_id) 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	long1	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

➤ Customer table :

```
CREATE TABLE Customers (  
  id INT PRIMARY KEY,  
  customer_name VARCHAR(100) NOT NULL,  
  city_id INT NOT NULL,  
  customer_address VARCHAR(255),  
  next_call_date DATE,
```

```

    ts_inserted DATETIME,
    FOREIGN KEY(city_id) REFERENCES cities(id)
);

```

```

INSERT INTO Customers (id, customer_name, city_id, customer_address,
next_call_date, ts_inserted) VALUES
(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');

```

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	Kurfürstendamm 25	2020-02-21	2020-01-09 17:52:15
3	Café	1	Tauentzienstraße 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

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

- List all Countries and customers related to these countries.

```

SELECT
    Countries.id AS country_id,
    Countries.country_name,
    Countries.country_name_eng,
    Countries.country_code,
    Customers.id AS customer_id,
    Customers.customer_name,
    Customers.customer_address,
    Customers.next_call_date,
    Customers.ts_inserted
FROM
    Countries
LEFT JOIN
    Cities ON Countries.id = Cities.country_id
LEFT JOIN
    Customers ON Cities.id = Customers.city_id;

```

country_id	country_name	country_name_eng	country_code	customer_id	customer_name	customer_address	next_call_date	ts_inserted
4	United States of America	United States of America	USA	1	Jewelry Store	Long Street 120	2020-01-21	2020-01-09 14:01:20
1	Deutschland	Germany	DEU	2	Bakery	Kurfürstendamm 25	2020-02-21	2020-01-09 17:52:15
1	Deutschland	Germany	DEU	3	Café	Tauentzienstraße 44	2020-01-21	2020-01-10 08:02:49
3	Hrvatska	Croatia	HRV	4	Restaurant	Ulica lipa 15	2020-01-21	2020-01-10 09:20:21
2	Srbija	Serbia	SRB	NULL	NULL	NULL	NULL	NULL
4	United States of America	United States of America	USA	NULL	NULL	NULL	NULL	NULL
5	Polska	Poland	POL	NULL	NULL	NULL	NULL	NULL
6	España	Spain	ESP	NULL	NULL	NULL	NULL	NULL
7	Rossija	Russia	RUS	NULL	NULL	NULL	NULL	NULL

- 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.

```

SELECT
  Countries.country_name_eng AS country_name,
  Cities.city_name AS city_name,
  Customers.customer_name AS customer_name
FROM
  Countries
LEFT JOIN
  Cities ON Countries.id = Cities.country_id
LEFT JOIN
  Customers ON Cities.id = Customers.city_id;

```

country_name	city_name	customer_name
United States of America	New York	Jewelry Store
Germany	Berlin	Bakery
Germany	Berlin	Café
Croatia	Zagreb	Restaurant
Serbia	Belgrade	NULL
United States of America	Los Angeles	NULL
Poland	Warsaw	NULL
Spain	NULL	NULL
Russia	NULL	NULL

- Return even countries without related cities and customers.

```

SELECT
  Countries.country_name_eng AS country_name,

```

```

Cities.city_name AS city_name,
Customers.customer_name AS customer_name
FROM
  Countries
LEFT JOIN
  Cities ON Countries.id = Cities.country_id
LEFT JOIN
  Customers ON Cities.id = Customers.city_id;

```

country_name	city_name	customer_name
United States of America	New York	Jewelry Store
Germany	Berlin	Bakery
Germany	Berlin	Café
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.

```

SELECT
  Countries.country_name_eng AS country_name,
  Cities.city_name AS city_name,
  Customers.customer_name AS customer_name
FROM
  Countries
INNER JOIN
  Cities ON Countries.id = Cities.country_id
LEFT JOIN

```

Customers ON Cities.id = Customers.city\_id;

country_name	city_name	customer_name
United States of America	New York	Jewelry Store
Germany	Berlin	Bakery
Germany	Berlin	Café
Croatia	Zagreb	Restaurant
Serbia	Belgrade	NULL
United States of America	Los Angeles	NULL
Poland	Warsaw	NULL

- Return even pairs of not having a single customer

```
SELECT
  Countries.country_name_eng AS country_name,
  Cities.city_name AS city_name,
  Customers.customer_name AS customer_name
FROM
  Countries
INNER JOIN
  Cities ON Countries.id = Cities.country_id
LEFT JOIN
  Customers ON Cities.id = Customers.city_id;
```

country_name	city_name	customer_name
United States of America	New York	Jewelry Store
Germany	Berlin	Bakery
Germany	Berlin	Café
Croatia	Zagreb	Restaurant
Serbia	Belgrade	NULL
United States of America	Los Angeles	NULL
Poland	Warsaw	NULL

