

3 TABLES : CITY, CUSTOMER, COUNTRY

COUNTRY TABLE :

```
+ CREATE TABLE Country (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    country_name VARCHAR(25),  
    country_name_eng VARCHAR(25),  
    country_code VARCHAR(10)  
);
```

id	country_name	country_name_eng	country_code
----	--------------	------------------	--------------

```
+ INSERT INTO Country (country_name,  
country_name_eng, country_code) VALUES  
("Deutschland", "Germany", "DEU"),  
("Srbija", "Serbia", "SRB"),  
("Hrvatska", "Croatia", "HRV"),  
("United States of America", "United  
States of America", "USA"),  
("Polska", "Poland", "POL"),  
("Espana", "Spain", "ESP"),  
("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	Espana	Spain	ESP
7	Rossiya	Russia	RUS

CITY TABLE :

```

+ CREATE TABLE City (
    id INT PRIMARY KEY AUTO_INCREMENT,
    city_name VARCHAR(25),
    lat DECIMAL(9,6),
    long_ DECIMAL(9,6),
    country_id INT,
    FOREIGN KEY (country_id)
REFERENCES Country(id)
);

```

id	city_name	lat	long_	country_id
----	-----------	-----	-------	------------

```

+ INSERT INTO City (city_name, lat,
  long_, country_id) VALUES
  ("Berlin", 52.520008, 13.404954, 1),
  ("Belgrade", 44.787197, 20.457273, 2),
  ("Zagreb", 45.815399, 15.966568, 3),
  ("New York", 40.730610, -73.935242,
4),
  ("Los Angeles", 34.052235, -
118.243683, 4),
  ("Warsaw", 52.237049, 21.017532, 5);

```

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

CUSTOMER TABLE :

```
+ CREATE TABLE Customer (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    customer_name VARCHAR(25),  
    city_id INT,  
    customer_address VARCHAR(25),  
    next_call_date DATE,  
    ts_inserted DATETIME,  
    FOREIGN KEY (city_id) REFERENCES  
    City(id)  
);
```

id	customer_name	city_id	customer_address	next_call_date	ts_inserted
----	---------------	---------	------------------	----------------	-------------

```
+ INSERT INTO Customer (customer_name,  
    city_id, customer_address,  
    next_call_date, ts_inserted) VALUES  
    ("Jewelry Store", 4, "Long Street  
120", "2020-01-21", "2020-01-09  
14:01:20.000"),  
    ("Bakery", 1, "Kurfurstendamm 25",  
    "2020-02-21", "2020-01-09  
17:52:15.000"),  
    ("Cafe", 1, "Tauentzienstrabe 44",  
    "2020-01-21", "2020-01-10  
08:02:49.000"),
```

```
("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	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

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,  
City.city_name,  
Customer.customer_name  
FROM Country  
LEFT JOIN City ON Country.id =  
City.country_id  
LEFT JOIN Customer ON City.id =  
Customer.city_id;
```

country_name_eng	city_name	customer_name
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 table 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.

```
+ SELECT country.country_name,  
  city.city_name, customer.customer_name  
FROM Country  
INNER JOIN City ON country.ID =  
  city.country_id  
LEFT JOIN Customer ON city.id =  
  customer.city_id;
```

country_name	city_name	customer_name
United States of America	New York	Jewelry Store
Deutschland	Berlin	Bakery
Deutschland	Berlin	Cafe
Hrvatska	Zagreb	Restaurant
Srbija	Belgrade	NULL
United States of America	Los Angeles	NULL
Polska	Warsaw	NULL