

# Assessment

To create the database and then create all three tables (city, customer, country) inside it, here are the exact MySQL commands you can run.

## Step 1: Create the Database

`CREATE DATABASE world_data;`

## Step 2: Use the Database

`USE world_data;`

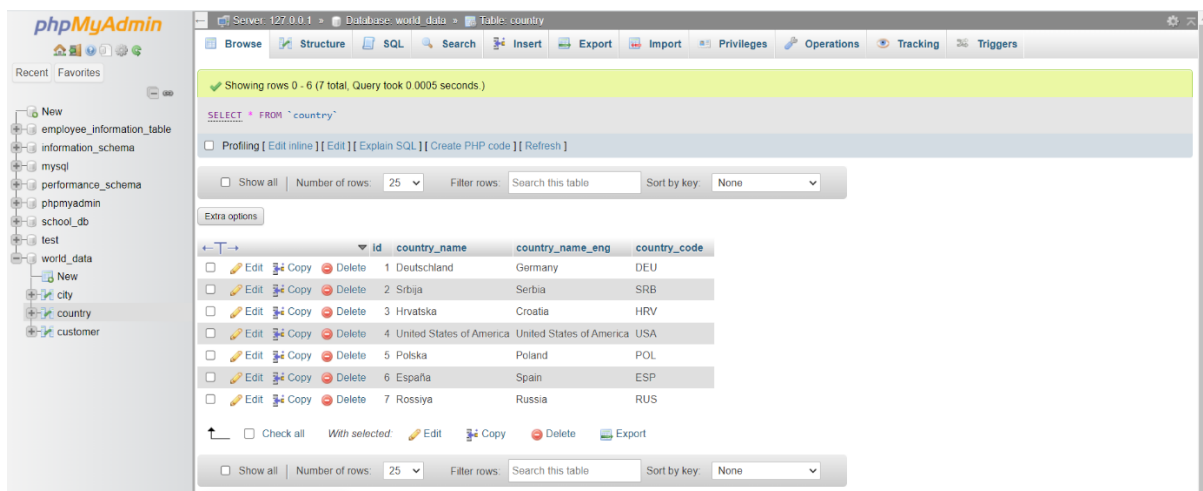
## Step 3: Create COUNTRY Table

- SQL – Create Table

```
CREATE TABLE country (  
  id INT PRIMARY KEY,  
  country_name VARCHAR(100),  
  country_name_eng VARCHAR(100),  
  country_code VARCHAR(10)  
);
```

- SQL – Insert Data

```
INSERT INTO country (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');
```



The screenshot shows the phpMyAdmin interface. On the left, the database structure is visible, with 'world\_data' selected. The main panel shows the 'country' table structure and data. The table has 7 rows of data, each with an 'id', 'country\_name', 'country\_name\_eng', and 'country\_code'.

| 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          |

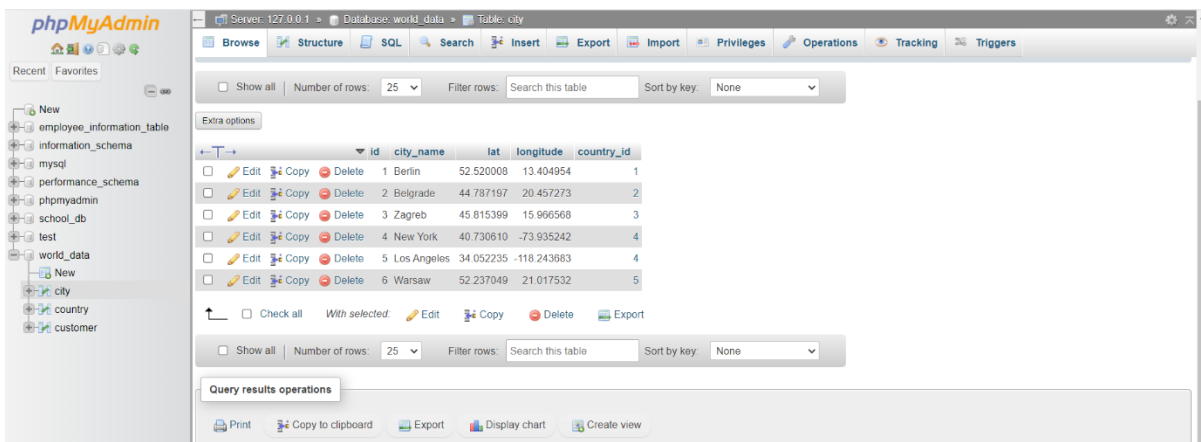
#### Step 4: Create CITY Table

- SQL – Create Table

```
CREATE TABLE city (  
    id INT PRIMARY KEY,  
    city_name VARCHAR(100),  
    lat DECIMAL(10,6),  
    longitude DECIMAL(10,6),  
    country_id INT,  
    FOREIGN KEY (country_id) REFERENCES country(id)  
);
```

- SQL -Insert Data

```
INSERT INTO city (id, city_name, lat, longitude, 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);
```



The screenshot shows the phpMyAdmin interface. On the left is the database navigation tree with 'world\_data' selected. The main panel displays the 'city' table structure and data. The table has 6 rows of data. Below the table are options for 'Query results operations' including Print, Copy to clipboard, Export, Display chart, and Create view.

|                          | id | city_name   | lat       | longitude   | country_id |
|--------------------------|----|-------------|-----------|-------------|------------|
| <input type="checkbox"/> | 1  | Berlin      | 52.520008 | 13.404954   | 1          |
| <input type="checkbox"/> | 2  | Belgrade    | 44.787197 | 20.457273   | 2          |
| <input type="checkbox"/> | 3  | Zagreb      | 45.815399 | 15.966568   | 3          |
| <input type="checkbox"/> | 4  | New York    | 40.730610 | -73.935242  | 4          |
| <input type="checkbox"/> | 5  | Los Angeles | 34.052235 | -118.243683 | 4          |
| <input type="checkbox"/> | 6  | Warsaw      | 52.237049 | 21.017532   | 5          |

#### Step 5: Create CUSTOMER Table

- SQL – Create Table

```
CREATE TABLE customer (  
    id INT PRIMARY KEY,  
    customer_name VARCHAR(100),  
    city_id INT,  
    customer_address VARCHAR(150),  
    next_call_date DATE,  
    ts_inserted DATETIME,  
    FOREIGN KEY (city_id) REFERENCES city(id)  
);
```

- SQL – Insert Data

**INSERT INTO** customer (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'),  
 (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 Ipa 15', '2020-01-21', '2020-01-10 09:20:21');

The screenshot shows the phpMyAdmin interface. On the left is the database navigation tree with 'world\_data' selected. The main panel shows the 'customer' table structure and data. A status bar at the top indicates 'Showing rows 0 - 3 (4 total, Query took 0.0009 seconds.)'. Below it, a SQL query editor shows 'SELECT \* FROM `customer`'. A table of 4 rows is displayed with columns: id, customer\_name, city\_id, customer\_address, next\_call\_date, and ts\_inserted. At the bottom, there are controls for selecting, editing, deleting, and exporting rows.

|                          | id | customer_name | city_id | customer_address    | next_call_date | ts_inserted         |
|--------------------------|----|---------------|---------|---------------------|----------------|---------------------|
| <input type="checkbox"/> | 1  | Jewelry Store | 4       | Long Street 120     | 2020-01-21     | 2020-01-09 14:01:20 |
| <input type="checkbox"/> | 2  | Bakery        | 1       | Kurfürstendamm 25   | 2020-02-21     | 2020-01-09 17:52:15 |
| <input type="checkbox"/> | 3  | Café          | 1       | Tauentzienstraße 44 | 2020-01-21     | 2020-01-10 08:02:49 |
| <input type="checkbox"/> | 4  | Restaurant    | 3       | Ulica Ipa 15        | 2020-01-21     | 2020-01-10 09:20:21 |

## ❖ Task 1: LEFT JOIN (Return all countries + related cities + customers)

### Requirement:

- Show all countries, including those without cities (Spain, Russia).
- Show country\_name\_eng, city\_name (if exists), customer\_name (if exists).
- LEFT JOIN must be used so missing cities/customers still appear.

### ➤ SQL Query (Task 1)

**SELECT**

c.country\_name\_eng,  
 ci.city\_name,  
 cu.customer\_name

**FROM** country c

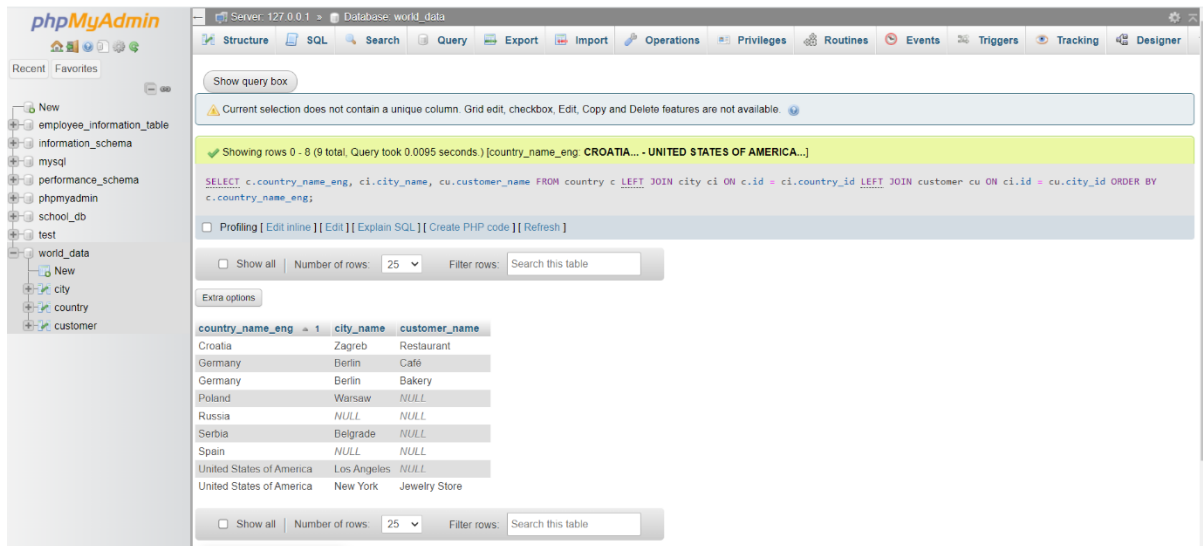
**LEFT JOIN** city ci

**ON** c.id = ci.country\_id

**LEFT JOIN** customer cu

**ON** ci.id = cu.city\_id

**ORDER BY** c.country\_name\_eng;



## ❖ Task 2: LEFT + INNER JOIN (Return only countries that have cities, and show their customers)

### Requirement:

- Return ONLY countries that have cities (so exclude Spain & Russia).
- For each country–city pair, return all customers, even if a city has zero customers.
- Use INNER JOIN + LEFT JOIN.

### ➤ SQL Query (Task 2)

#### SELECT

c.country\_name\_eng,  
ci.city\_name,  
cu.customer\_name

#### FROM country c

#### INNER JOIN city ci

ON c.id = ci.country\_id -- only countries that have cities

#### LEFT JOIN customer cu

ON ci.id = cu.city\_id -- return cities even without customers

ORDER BY c.country\_name\_eng, ci.city\_name;

phpMyAdmin

Server: 127.0.0.1 - Database: world\_data

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer

Show query box

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 6 (7 total, Query took 0.0084 seconds) [country\_name\_eng CROATIA... - UNITED STATES OF AMERICA...] [city\_name ZAGREB... - NEW YORK...]

```
SELECT c.country_name_eng, ci.city_name, cu.customer_name FROM country c INNER JOIN city ci ON c.id = ci.country_id LEFT JOIN customer cu ON ci.id = cu.city_id ORDER BY c.country_name_eng, ci.city_name;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

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

☐ Show all | Number of rows: 25 | Filter rows: Search this table