

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');
```

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

	id	city_name	lat	longitude	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

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');
```

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

❖ **TASK 1 — LEFT JOIN (List ALL countries + related cities + related customers)**

➤ Requirements:

- Show ALL countries (even without cities)
- For each country → show city → show customer
- Countries with no cities (Spain, Russia) must appear
- Cities with no customers must appear
- Customers linked to cities (Berlin, Zagreb, New York)

➤ Query: 1

```
SELECT
```

```
co.country_name_eng AS country,
ci.city_name AS city,
cu.customer_name AS customer
```

```
FROM country co
```

```
LEFT JOIN city ci
```

```
ON co.id = ci.country_id
```

```
LEFT JOIN customer cu
```

```
ON ci.id = cu.city_id ORDER BY co.id;
```

The screenshot shows the phpMyAdmin interface for a database named 'world_data'. The left sidebar lists various databases and their tables. The main area displays a query result for a LEFT JOIN between three tables: country, city, and customer. The SQL query is:

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

The results show data from three tables:

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

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

➤ Requirements:

- Return ONLY countries referenced by at least one city
 - Exclude: Spain, Russia
 - For each such country → show all city-customer pairs
- Countries with no customers must still appear (Serbia, Los Angeles, Warsaw)

➤ Query: 2

`SELECT`

```
co.country_name_eng AS country,
ci.city_name AS city,
cu.customer_name AS customer
```

`FROM country co`

`INNER JOIN city ci`

`ON co.id = ci.country_id`

`LEFT JOIN customer cu`

`ON ci.id = cu.city_id`

`ORDER BY co.id, ci.id;`

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 0.0014 seconds)

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

Profiling | Edit inline | Edit | Explain SQL | Create PHP code | Refresh

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

Extra options

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

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

Query results operations

Print Copy to clipboard Export Display chart Create view

The screenshot shows the phpMyAdmin interface for a database named 'world_data'. On the left, there's a sidebar with a tree view of databases and tables. The main area displays a query result for a joined table. The query is:`SELECT co.country_name_eng AS country, ci.city_name AS city, cu.customer_name AS customer FROM country co INNER JOIN city ci ON co.id = ci.country_id LEFT JOIN customer cu ON ci.id = cu.city_id ORDER BY co.id, ci.id;`

The result table has three columns: 'country', 'city', and 'customer'. It contains 7 rows of data. The 'customer' column has several NULL values.

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