Task 1: Left Join to list all countries and related customers SELECT coun.country_name_en AS Country, city.city_name AS City, cust.customer_name AS Customer FROM country AS coun LEFT JOIN city ON coun.country_id = city.country_id LEFT JOIN customer AS cust ON city.city_id = cust.city_id ORDER BY coun.country_name_en, city.city_name, cust.customer_name;

Task 2: Left and Inner Join to list countries with pairs and related customers SELECT coun.country_name_en AS Country, city.city_name AS City, cust.customer_name AS Customer FROM country AS coun INNER JOIN city ON coun.country_id = city.country_id LEFT JOIN customer AS cust ON city.city_id = cust.city_id ORDER BY coun.country_name_en, city.city_name, cust.customer_name;

EXLAINATION:

Task 1 (Left Join):

1.SELECT coun.country_name_en AS Country, city.city_name AS City, cust.customer_name AS Customer: This

selects the desired columns and aliases them for better readability.

FROM country AS coun: This starts with the country table as the base.

1.LEFT JOIN city ON coun.country_id = city.country_id: This performs a LEFT JOIN with the city table, ensuring that all countries are included, even if they don't have matching cities.

2.LEFT JOIN customer AS cust ON city.city_id = cust.city_id: This performs another LEFT JOIN with the customer table, ensuring that all cities are included, even if they don't have matching customers.

3.ORDER BY coun.country_name_en, city.city_name, cust.customer_name;: This orders the result for better presentation.

Task 2 (Inner and Left Join):

SELECT coun.country_name_en AS Country, city.city_name AS City, cust.customer_name AS Customer: Selects the desired columns and aliases them.

1.	**FROM country AS coun**: Starts with the country table.

2.	**INNER JOIN city ON coun.country_id = city.country_id**: This performs an INNER JOIN with the city table. This ensures that only countries that have matching cities are included. This is the main difference from task 1.

3.	**LEFT JOIN customer AS cust ON city.city_id = cust.city_id**: This performs a LEFT JOIN with the customer table, ensuring that all city-country pairs are included, even if they don't have matching customers.

4.	**ORDER BY coun.country_name_en, city.city_name, cust.customer_name;**: Orders the result.