SQL LAB- In pgAdmin, create a database called `northwind`

-- 1. Select all the records from the customers table.

```
SELECT * FROM customers;
```

-- 2. Get distinct countries from the customers table.

```
SELECT distinct country from customers;
```

-- 3. Get all the records from the table customers where the customer_id starts with “BL”.

```
SELECT * FROM customers where customer_id like 'BL%';
```

--4. Get the first 100 records of the orders table.

```
SELECT * FROM orders limit 100;
```

--5. Get all customers that live in the postal codes 1010, 3012, 12209, and 05023.

```
SELECT * FROM customers where postal_code in ('1010', '3012', '12209', '05023');
```

--6. Get all orders where the ShipRegion is not equal to NULL.

```
SELECT * FROM orders WHERE ship_region IS NOT null;
```

--7. Get all customers ordered by the country, then by the city.

```
SELECT * FROM customers ORDER BY country, city;
```

-- 8. Add a new customer to the customers table. You can use whatever values/

```
INSERT INTO customers(customer_id, company_name, contact_name, contact_title,
address, city, region, postal_code, country, phone, fax)
VALUES ('Abgil', 'digi chef', 'Antonio', 'Owner', 'street 00', 'Detroit','[null]', '4318',
'USA', '2345678', '020-0065700' )
```

--9. Update all ship_region to the value 'EuroZone' in the orders table, where the ship_country is 'France'.

```
UPDATE orders SET ship_region = 'EuroZone'
WHERE ship_country ='France';
```

--10. Delete all rows from order_details that have a quantity of 1.

```
DELETE FROM order_details WHERE quantity = 1;
```

--11. Calculate the average, max, and min of the quantity in the order_details table

```
SELECT AVG(quantity) AS "avg qty",
MAX(quantity) AS "max qty",
MIN(quantity) AS "min qty"
FROM order_details;
```

--12. Calculate the average, max, and min of the quantity in the order_details table, grouped by the order_id.

```
SELECT AVG(quantity), MAX(quantity), MIN(quantity) FROM order_details
GROUP BY order_id;
```

--13. Find the customer_id that placed order 10290 (orders table)

```
SELECT customer_id FROM orders WHERE order_id = '10290';
```

-- 14. Do an inner join, left join, right join on orders and customers tables. (These are three separate queries, one for each type of join.)

```
SELECT * FROM customers
INNER JOIN orders
ON customers.customer_id = orders.customer_id;

SELECT * FROM customers
LEFT JOIN orders
ON customers.customer_id = orders.customer_id;
```

```
SELECT * FROM customers
RIGHT JOIN orders
ON customers.customer_id = orders.customer_id;
```

--15. Use a join to get the ship_city and ship_country of all the orders which are associated with an employee who is in London.

```
SELECT orders.ship_city, orders.ship_country
FROM orders
JOIN employees
ON orders.employee_id = employees.employee_id
WHERE employees.city = 'London';
```

--16. Use a join to get the ship_name of all orders that include a discontinued product. (See orders, order_details, and products. 1 means discontinued.)

```
SELECT orders.ship_name
FROM orders
INNER JOIN order_details
ON orders.order_id = order_details.order_id
INNER JOIN products
ON order_details.product_id = products.product_id
WHERE products.discontinued = 1;
```

--17. Get first names of all employees who report to no one.

```
SELECT employees.first_name FROM employees WHERE reports_to IS null;
```

--18. Get first names of all employees who report to 'Andrew'

```
SELECT first_name
FROM employees
WHERE reports_to =
        (SELECT employee_id
```

```
                FROM employees
                WHERE first_name='Andrew');
///             OR
            SELECT subordinate.first_name
            FROM employees AS subordinate
            JOIN employees AS manager
            ON subordinate.reports_to = manager.employee_id
            WHERE manager.first_name = 'Andrew'
```