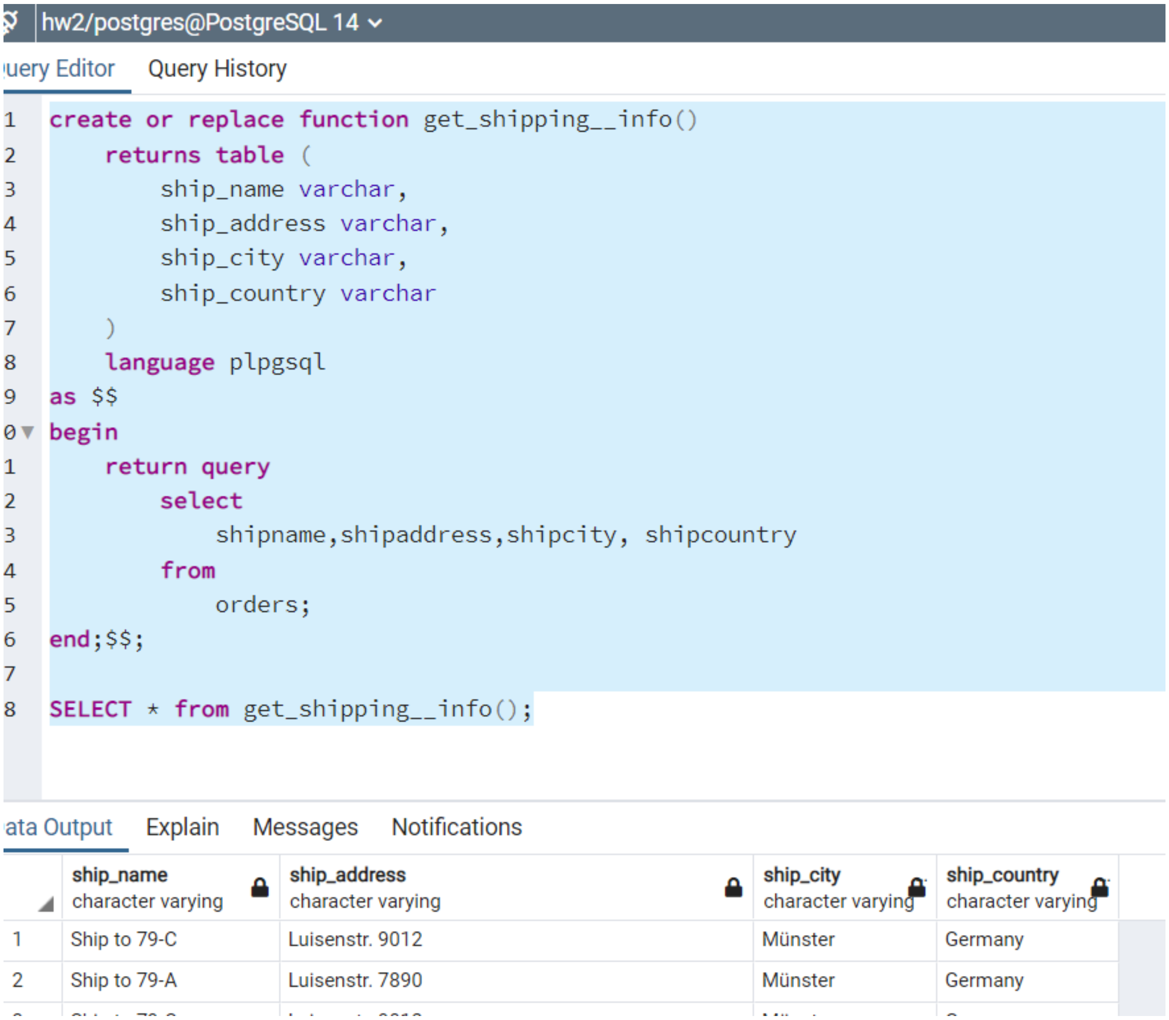
For homework 5 solution, I have tried to solve it and run the c and d number query.

c) Create a function **get_shipping_info (varchar)** that returns a table. The table should have the following columns ***shipname***, ***shipaddress***, ***shipcity***, ***shipcountry.*** The function should return orders where the shipname matches the given string.

Query code-

```
create or replace function get_shipping__info()
        returns table (
                ship_name varchar,
                ship_address varchar,
                ship_city varchar,
                ship_country varchar
        )
        language plpgsql
as $$
begin
        return query
                select
                        shipname,shipaddress,shipcity, shipcountry
                from
                        orders;
end;$$;

SELECT * from get_shipping__info();
```



```
create or replace function get_shipping__info()
    returns table (
        ship_name varchar,
        ship_address varchar,
        ship_city varchar,
        ship_country varchar
    )
    language plpgsql
as $$
begin
    return query
        select
            shipname,shipaddress,shipcity, shipcountry
        from
            orders;
end;$$;

SELECT * from get_shipping__info();
```

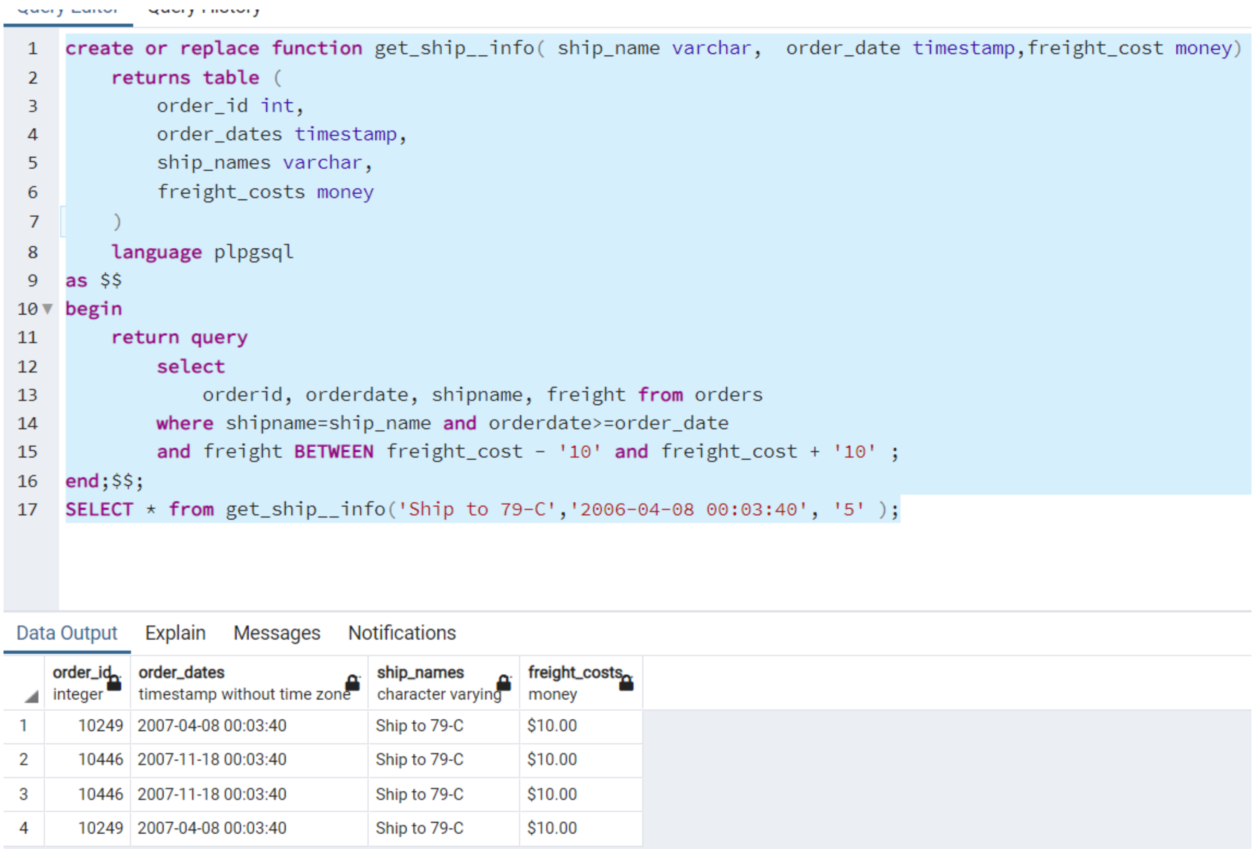
	ship_name character varying	ship_address character varying	ship_city character varying	ship_country character varying
1	Ship to 79-C	Luisenstr. 9012	Münster	Germany
2	Ship to 79-A	Luisenstr. 7890	Münster	Germany

d) Extend the function in c) so that it accepts three parameters: **get_shipping_info(varchar, timestamp, money).** The function should return orders where the shipname matches the given string, orderdate is equal or earlier than the given timestamp and finally, freight cost is +-10€ from the given money.

Query code-

```
create or replace function get_ship__info( ship_name varchar,  order_date
timestamp,freight_cost money)
            returns table (
                        order_id int,
                        order_dates timestamp,
                        ship_names varchar,
                        freight_costs money
            )
```

```
        language plpgsql
as $$
begin
        return query
                select
                        orderid, orderdate, shipname, freight from orders
                where shipname=ship_name and orderdate>=order_date
                and freight BETWEEN freight_cost - '10' and freight_cost + '10' ;
end;$$;
SELECT * from get_ship__info('Ship to 79-C','2006-04-08 00:03:40', '5' );
```



```
create or replace function get_ship__info( ship_name varchar,  order_date timestamp,freight_cost money)
    returns table (
        order_id int,
        order_dates timestamp,
        ship_names varchar,
        freight_costs money
    )
    language plpgsql
as $$
begin
    return query
        select
            orderid, orderdate, shipname, freight from orders
        where shipname=ship_name and orderdate>=order_date
        and freight BETWEEN freight_cost - '10' and freight_cost + '10' ;
end;$$;
SELECT * from get_ship__info('Ship to 79-C','2006-04-08 00:03:40', '5' );
```

	order_id integer	order_dates timestamp without time zone	ship_names character varying	freight_costs money
1	10249	2007-04-08 00:03:40	Ship to 79-C	$10.00
2	10446	2007-11-18 00:03:40	Ship to 79-C	$10.00
3	10446	2007-11-18 00:03:40	Ship to 79-C	$10.00
4	10249	2007-04-08 00:03:40	Ship to 79-C	$10.00

a) Create the following roles: **user**, **manager**, **owner**. Grant all privileges to owner, read privileges to user, and insert privileges to manager.

Query code-

```
CREATE ROLE users;
GRANT pg_read_all_data TO users;
```

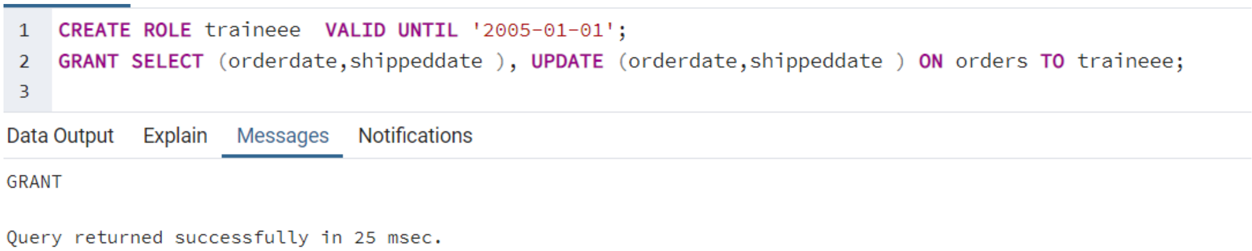
```
CREATE ROLE manager;
GRANT INSERT ON  orders TO  manager;

CREATE ROLE Owner with SUPERUSER;
GRANT SELECT, INSERT, UPDATE, DELETE ON orders TO Owner;
```

b) Create a new role: **trainee**. Grant privileges only to columns *orderdate* and *shippeddate* to **trainee** and set the role valid until 30.5.2022.

Query code-

```
CREATE ROLE trainee  VALID UNTIL '2005-01-01';
GRANT SELECT (orderdate,shippeddate ), UPDATE (orderdate,shippeddate ) ON orders TO trainee;
```



```
CREATE ROLE traineee  VALID UNTIL '2005-01-01';
GRANT SELECT (orderdate,shippeddate ), UPDATE (orderdate,shippeddate ) ON orders TO traineee;
```

```
GRANT

Query returned successfully in 25 msec.
```

Here I have given select and update privilege to trainee for columns shipdate and orderdate.