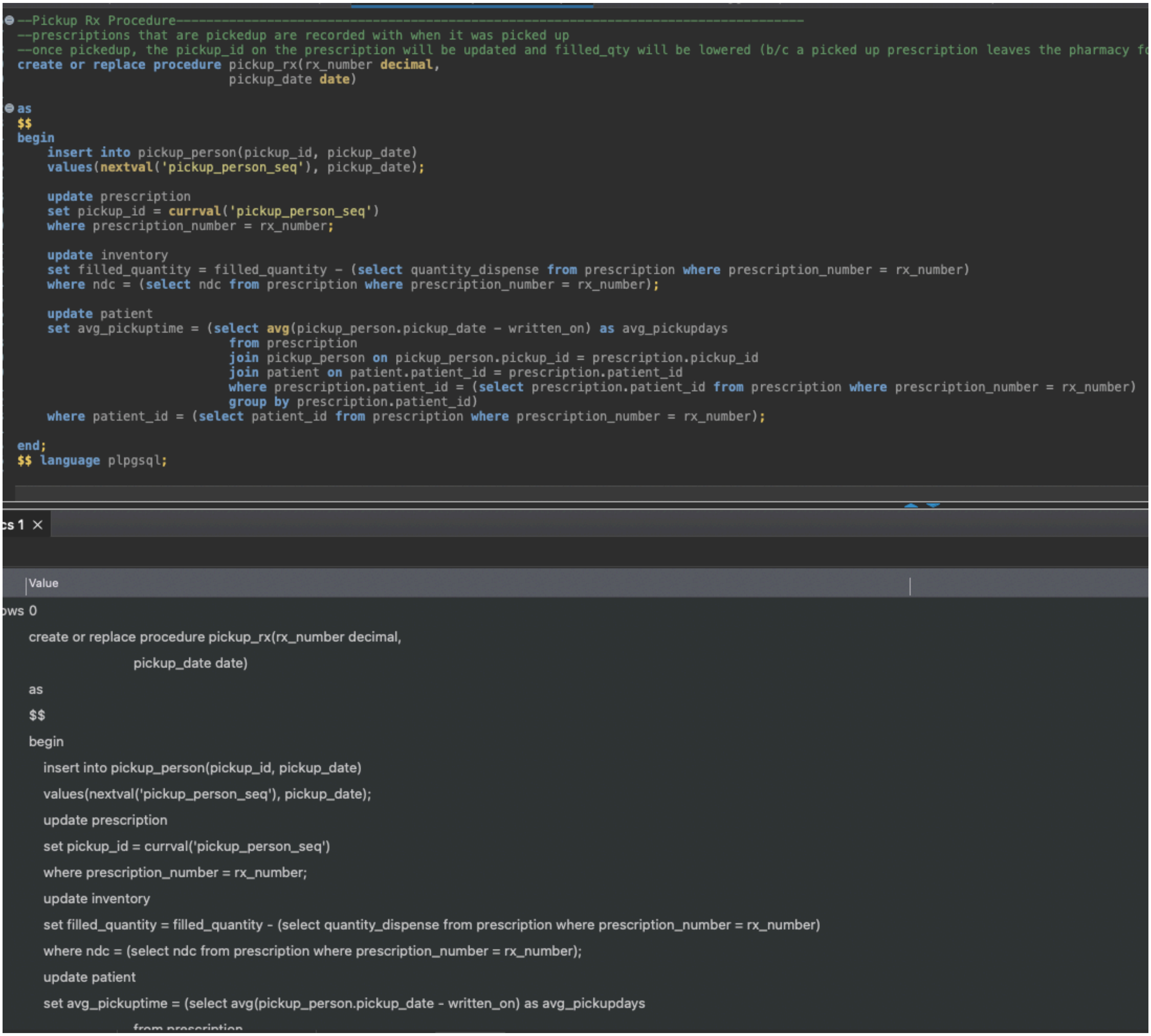
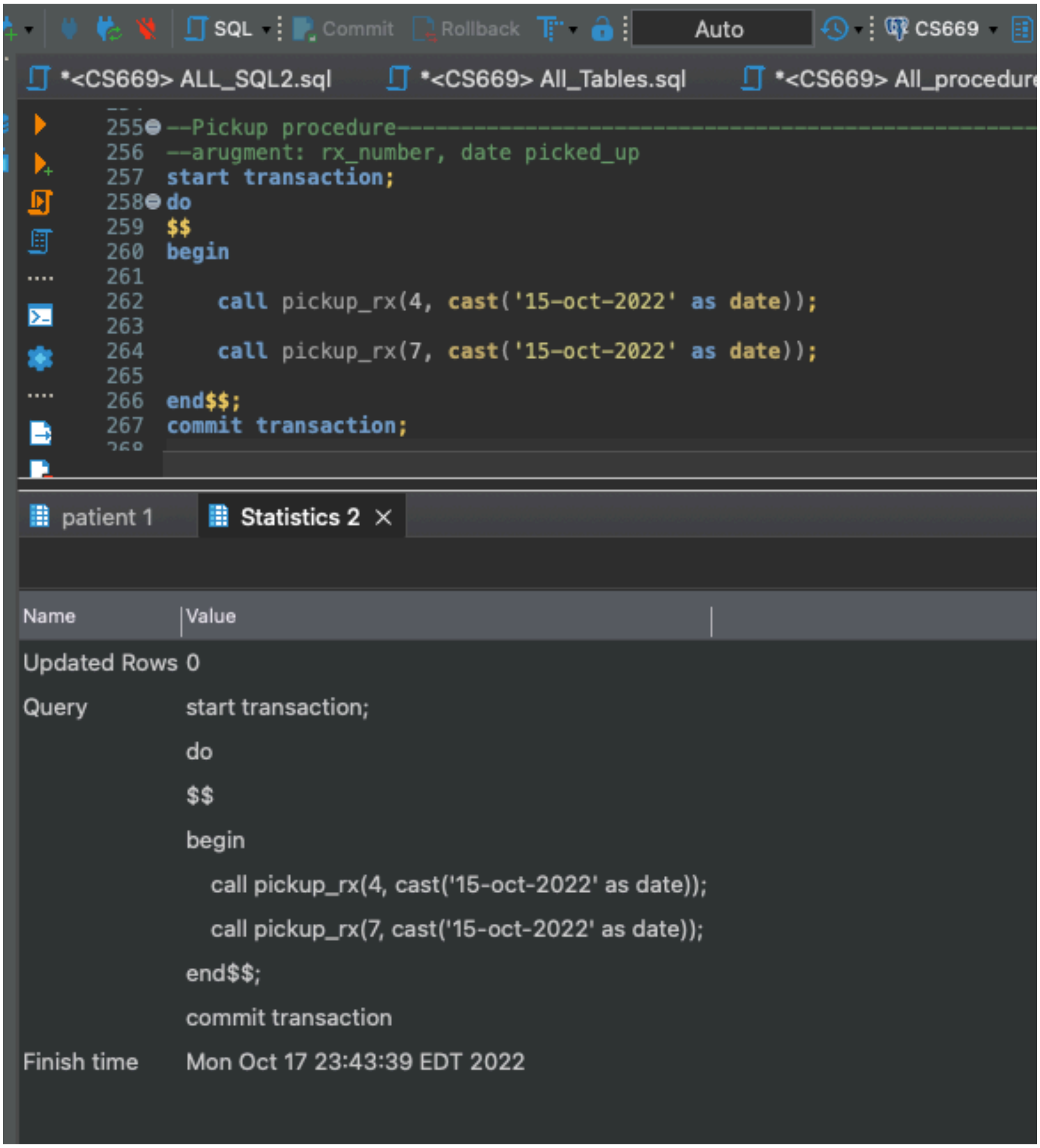
<u>Pickup_rx store procedure</u>



```sql
--Pickup Rx Procedure-----------------------------------------------------------------------------------
--prescriptions that are pickedup are recorded with when it was picked up
--once pickedup, the pickup_id on the prescription will be updated and filled_qty will be lowered (b/c a picked up prescription leaves the pharmacy f
create or replace procedure pickup_rx(rx_number decimal,
                            pickup_date date)

as
$$
begin
    insert into pickup_person(pickup_id, pickup_date)
    values(nextval('pickup_person_seq'), pickup_date);

    update prescription
    set pickup_id = currval('pickup_person_seq')
    where prescription_number = rx_number;

    update inventory
    set filled_quantity = filled_quantity - (select quantity_dispense from prescription where prescription_number = rx_number)
    where ndc = (select ndc from prescription where prescription_number = rx_number);

    update patient
    set avg_pickuptime = (select avg(pickup_person.pickup_date - written_on) as avg_pickupdays
                            from prescription
                            join pickup_person on pickup_person.pickup_id = prescription.pickup_id
                            join patient on patient.patient_id = prescription.patient_id
                            where prescription.patient_id = (select prescription.patient_id from prescription where prescription_number = rx_number)
                            group by prescription.patient_id)
    where patient_id = (select patient_id from prescription where prescription_number = rx_number);

end;
$$ language plpgsql;
```

```
ows 0
create or replace procedure pickup_rx(rx_number decimal,
              pickup_date date)
as
$$
begin
  insert into pickup_person(pickup_id, pickup_date)
  values(nextval('pickup_person_seq'), pickup_date);
  update prescription
  set pickup_id = currval('pickup_person_seq')
  where prescription_number = rx_number;
  update inventory
  set filled_quantity = filled_quantity - (select quantity_dispense from prescription where prescription_number = rx_number)
  where ndc = (select ndc from prescription where prescription_number = rx_number);
  update patient
  set avg_pickuptime = (select avg(pickup_person.pickup_date - written_on) as avg_pickupdays
              from prescription
```

This stored procedure takes the prescription number and date as arguments. The stored procedure will insert the argument pickup_date into the pickup_date column of the 'Pickup_person' table and generates a 'Pickup_person_seq' sequence that references the pickup_date. Once a prescription number is picked up, serval updates are made. First, the pickup_id of the prescription number in the prescription table will be updated to the current ‘Pickup_person_seq’ sequence. Second, the filled_quantity of the drug that corresponds to the prescription number is subtracted by the quantity of the prescription using subqueries. Third, the avg_pickuptime in the patient table is updated. I took the average of written_on date and the pickup_date group by patient_id. This means that I first determine the average pickup days for all the patients, and using that as a subquery, I assigned the avg_pickuptime according to the patient_ID of the given rx_number argument (using subquery again).



```
--Pickup procedure--------------------------------------------------
--arugment: rx_number, date picked_up
start transaction;
do
$$
begin

    call pickup_rx(4, cast('15-oct-2022' as date));

    call pickup_rx(7, cast('15-oct-2022' as date));

end$$;
commit transaction;
```

```
Name          Value
Updated Rows  0
Query         start transaction;
              do
              $$
              begin
                  call pickup_rx(4, cast('15-oct-2022' as date));
                  call pickup_rx(7, cast('15-oct-2022' as date));
              end$$;
              commit transaction
Finish time   Mon Oct 17 23:43:39 EDT 2022
```