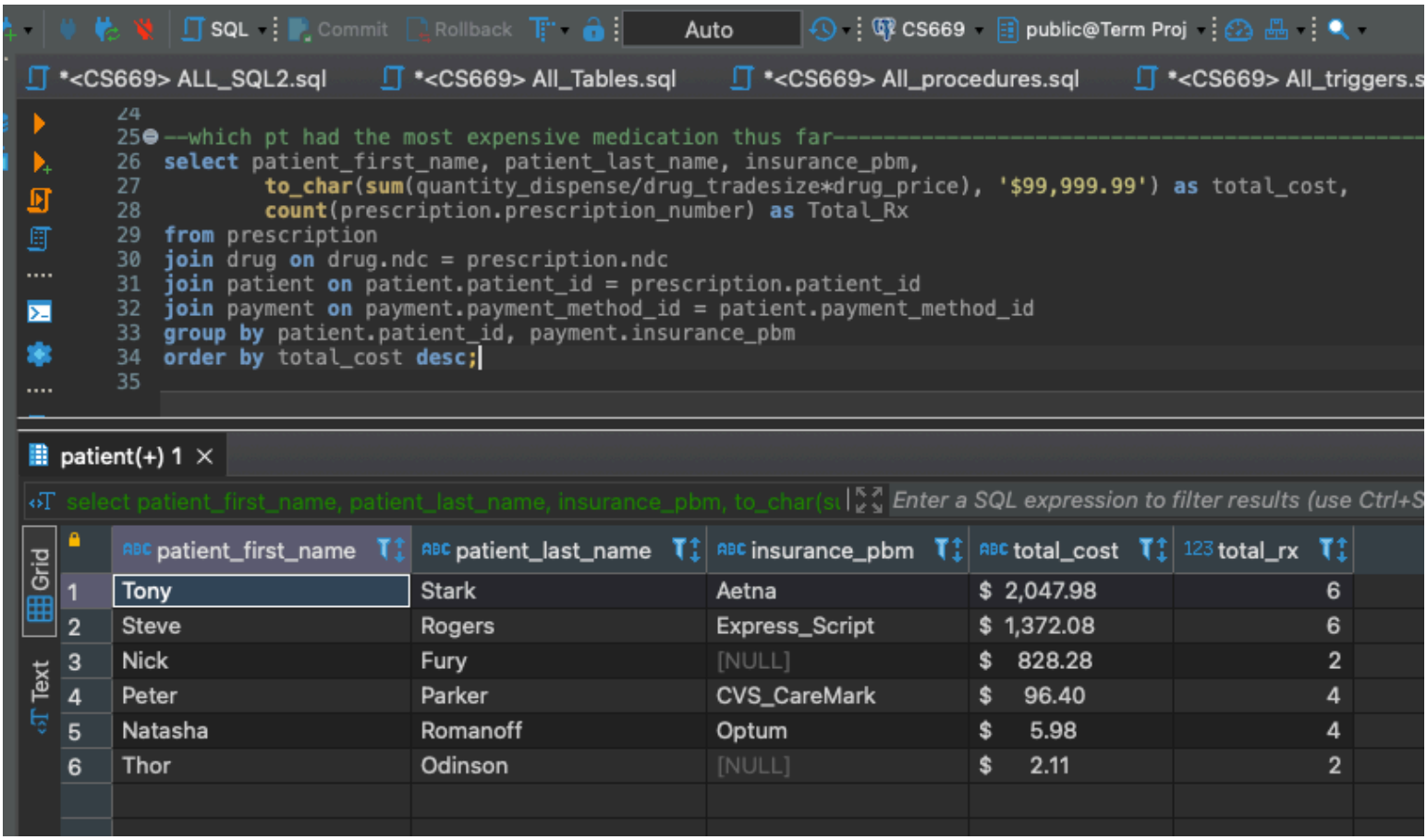
1. Which patient has the most total medication cost?



```
--which pt had the most expensive medication thus far----------
select patient_first_name, patient_last_name, insurance_pbm,
        to_char(sum(quantity_dispense/drug_tradesize*drug_price), '$99,999.99') as total_cost,
        count(prescription.prescription_number) as Total_Rx
from prescription
join drug on drug.ndc = prescription.ndc
join patient on patient.patient_id = prescription.patient_id
join payment on payment.payment_method_id = patient.payment_method_id
group by patient.patient_id, payment.insurance_pbm
order by total_cost desc;
```

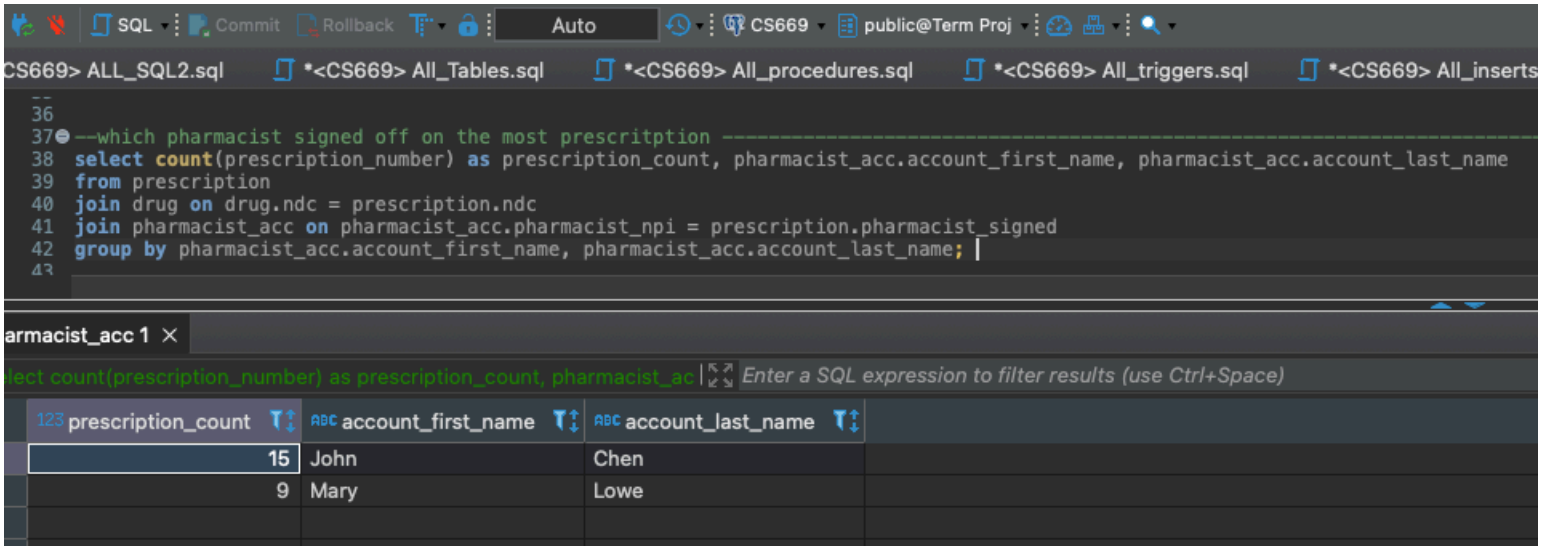
patient(+) 1

	patient_first_name	patient_last_name	insurance_pbm	total_cost	total_rx
1	Tony	Stark	Aetna	$ 2,047.98	6
2	Steve	Rogers	Express_Script	$ 1,372.08	6
3	Nick	Fury	[NULL]	$ 828.28	2
4	Peter	Parker	CVS_CareMark	$ 96.40	4
5	Natasha	Romanoff	Optum	$ 5.98	4
6	Thor	Odinson	[NULL]	$ 2.11	2

This query joins prescription, drug, patient, and payment together and returns the patient's full name, their insurance PBM, and the total_cost and total_rx. Total_cost is calculated by determine the cost of each prescription that's grouped by patient_ID. Each prescription cost is determined by the quantity dispense divided by its trade size and multiplying it by each trade size price. And Total_rx is the count of each prescription number grouped by each patient_ID.

This shows that Tony Stark has the most costly medication even though he and Steven Rogers shares the same number of prescriptions.

2. Which pharmacist is verifying the most prescriptions

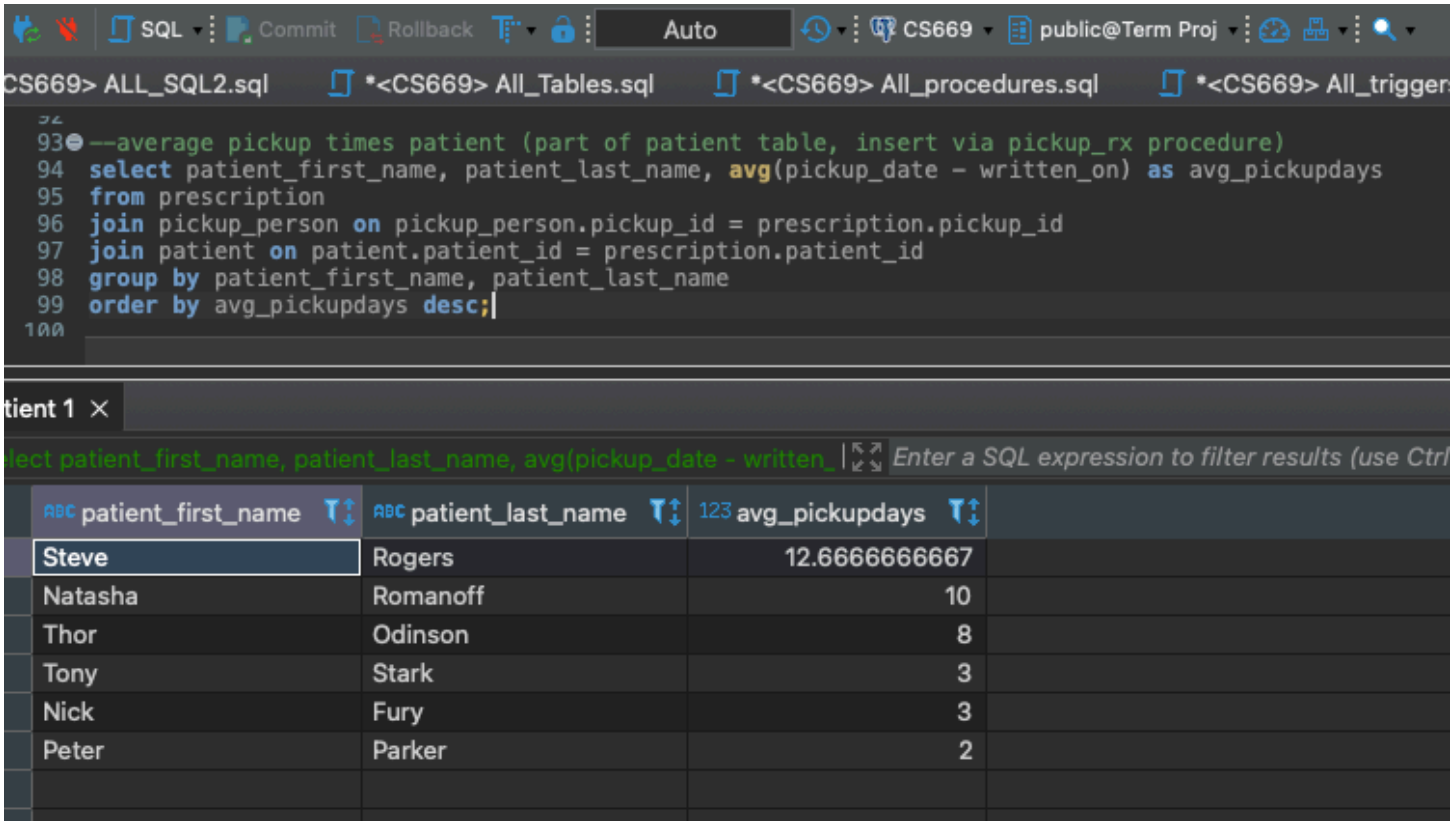


```
--which pharmacist signed off on the most prescritption
select count(prescription_number) as prescription_count, pharmacist_acc.account_first_name, pharmacist_acc.account_last_name
from prescription
join drug on drug.ndc = prescription.ndc
join pharmacist_acc on pharmacist_acc.pharmacist_npi = prescription.pharmacist_signed
group by pharmacist_acc.account_first_name, pharmacist_acc.account_last_name;
```

prescription_count	account_first_name	account_last_name
15	John	Chen
9	Mary	Lowe

This query joins prescription, drug, and pharmacist_acc together. It groups all the prescriptions that have been pharmacist signed off by pharmacist's first and last name. It then counts each prescription as prescription_count. I've added a small sample of data for September to demonstrate this table. It's important to note that this table would not count any prescriptions that have not been signed off on by a pharmacist.
This query shows that John Chen verifies almost 50% more than Mary Lowe.

3. What is the average pick up time for each patient



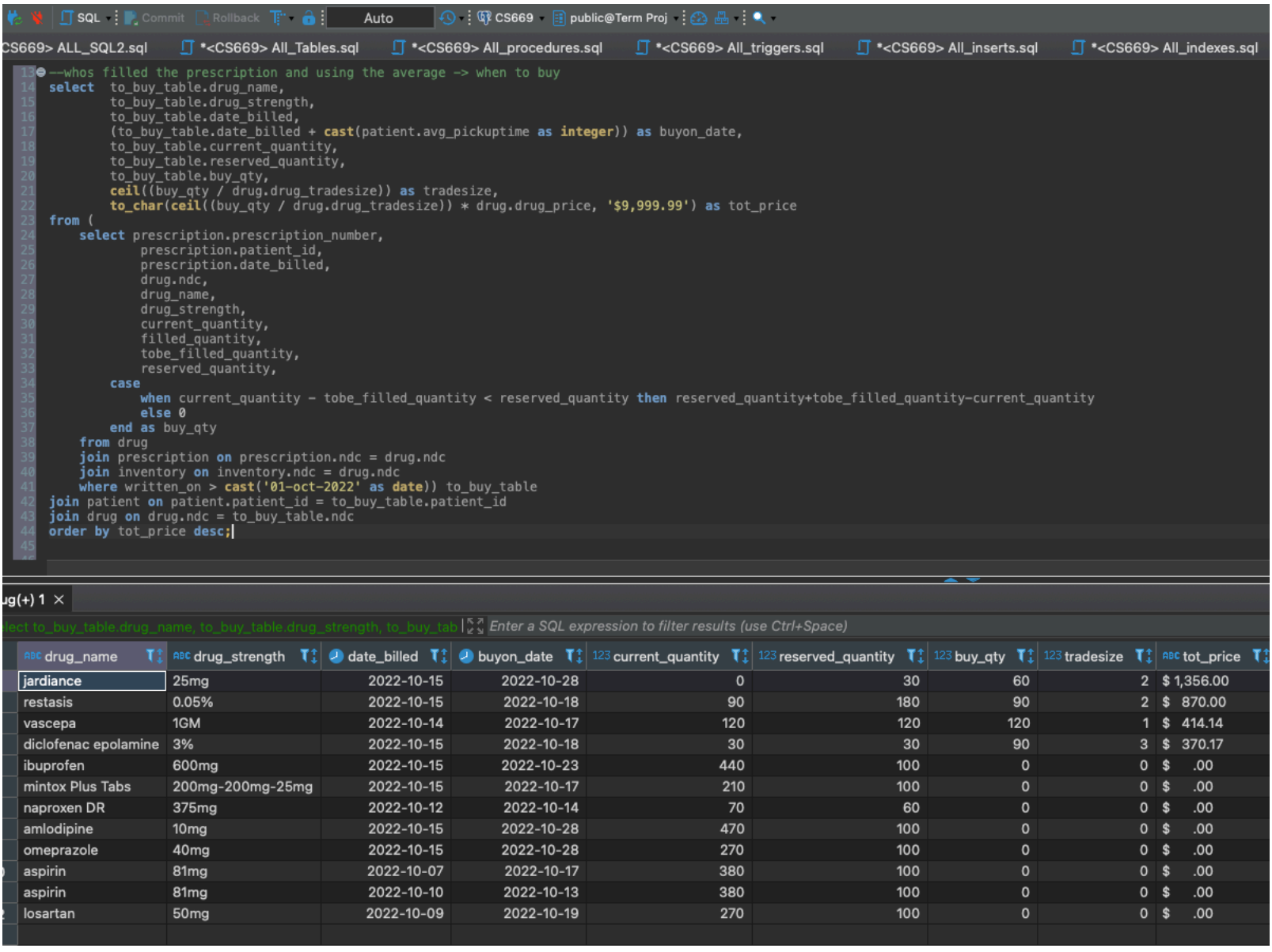
```
--average pickup times patient (part of patient table, insert via pickup_rx procedure)
select patient_first_name, patient_last_name, avg(pickup_date - written_on) as avg_pickupdays
from prescription
join pickup_person on pickup_person.pickup_id = prescription.pickup_id
join patient on patient.patient_id = prescription.patient_id
group by patient_first_name, patient_last_name
order by avg_pickupdays desc;
```

patient_first_name	patient_last_name	avg_pickupdays
Steve	Rogers	12.6666666667
Natasha	Romanoff	10
Thor	Odinson	8
Tony	Stark	3
Nick	Fury	3
Peter	Parker	2

This query joins prescription, pickup_person, and the patient table together. The query groups each prescription by patient's name and takes the average of the difference between the day a prescription was written and the day a prescription was pickup and orders by that average.

This table shows that Steven Rogers takes on average almost 13 days for a prescription to be picked up. While Peter Parker usually only takes 2 days.

4. How much drug and/or dollar amount do I need to buy for a given medication and when do I need to buy it by?



```sql
--whos filled the prescription and using the average -> when to buy
select  to_buy_table.drug_name,
        to_buy_table.drug_strength,
        to_buy_table.date_billed,
        (to_buy_table.date_billed + cast(patient.avg_pickuptime as integer)) as buyon_date,
        to_buy_table.current_quantity,
        to_buy_table.reserved_quantity,
        to_buy_table.buy_qty,
        ceil((buy_qty / drug.drug_tradesize)) as tradesize,
        to_char(ceil((buy_qty / drug.drug_tradesize)) * drug.drug_price, '$9,999.99') as tot_price
from (
    select prescription.prescription_number,
           prescription.patient_id,
           prescription.date_billed,
           drug.ndc,
           drug_name,
           drug_strength,
           current_quantity,
           filled_quantity,
           tobe_filled_quantity,
           reserved_quantity,
        case
            when current_quantity - tobe_filled_quantity < reserved_quantity then reserved_quantity+tobe_filled_quantity-current_quantity
            else 0
        end as buy_qty
    from drug
    join prescription on prescription.ndc = drug.ndc
    join inventory on inventory.ndc = drug.ndc
    where written_on > cast('01-oct-2022' as date)) to_buy_table
join patient on patient.patient_id = to_buy_table.patient_id
join drug on drug.ndc = to_buy_table.ndc
order by tot_price desc;
```

drug_name	drug_strength	date_billed	buyon_date	current_quantity	reserved_quantity	buy_qty	tradesize	tot_price
jardiance	25mg	2022-10-15	2022-10-28	0	30	60	2	$ 1,356.00
restasis	0.05%	2022-10-15	2022-10-18	90	180	90	2	$ 870.00
vascepa	1GM	2022-10-14	2022-10-17	120	120	120	1	$ 414.14
diclofenac epolamine	3%	2022-10-15	2022-10-18	30	30	90	3	$ 370.17
ibuprofen	600mg	2022-10-15	2022-10-23	440	100	0	0	$ .00
mintox Plus Tabs	200mg-200mg-25mg	2022-10-15	2022-10-17	210	100	0	0	$ .00
naproxen DR	375mg	2022-10-12	2022-10-14	70	60	0	0	$ .00
amlodipine	10mg	2022-10-15	2022-10-28	470	100	0	0	$ .00
omeprazole	40mg	2022-10-15	2022-10-28	270	100	0	0	$ .00
aspirin	81mg	2022-10-07	2022-10-17	380	100	0	0	$ .00
aspirin	81mg	2022-10-10	2022-10-13	380	100	0	0	$ .00
losartan	50mg	2022-10-09	2022-10-19	270	100	0	0	$ .00

This query references a table made from a subquery. The subquery from the FROM clause joins prescription, drug, and inventory table together. When the current quantity subtracts

tobe_filled_quantity (both from the inventory table) and that is less than the reserved_quantity (a reserved_quantity must remain levels at all times), then the difference between the reserved quantity plus the tobe_filled_quantity less the current available quantity is the quantity that needs to be brought.
I also added the WHERE clause because the prescription table currently has data from September (even though those data point does not affect our inventory table).

The outer query takes the inner query and joins it with patient and drug. The outer query takes most of the columns from the inner query, but also adds on new columns. From the inner query, date_billed is added the average pick up time from the patient table and returns as buyon_date. This new column indicates the average date that the patient will come and pick up and thus medication should be filled and ready for them to pick up. The trade size column is essential how many bottles or packages the buy_qty is. The buy_qty was referenced fro the inner query. Since trade size does not come in partial, all trade sizes are rounded to the next trade size. Tot_price is just the trade size times the price per each trade size.

This table shows that the most costly item is 2 bottles of 'Jardiance', but that doesn't need to be brought right away since it's average pickup date is around the 10/28. We should focus on buying 3 packs of diclofenac and/or 1 bottle of vascepa since the patient will be coming to pick up soon (around 10/17-18).