SQL QUEIRIES (LAB-11)

1) **List the names of doctors from each department with highest consultation charges**.

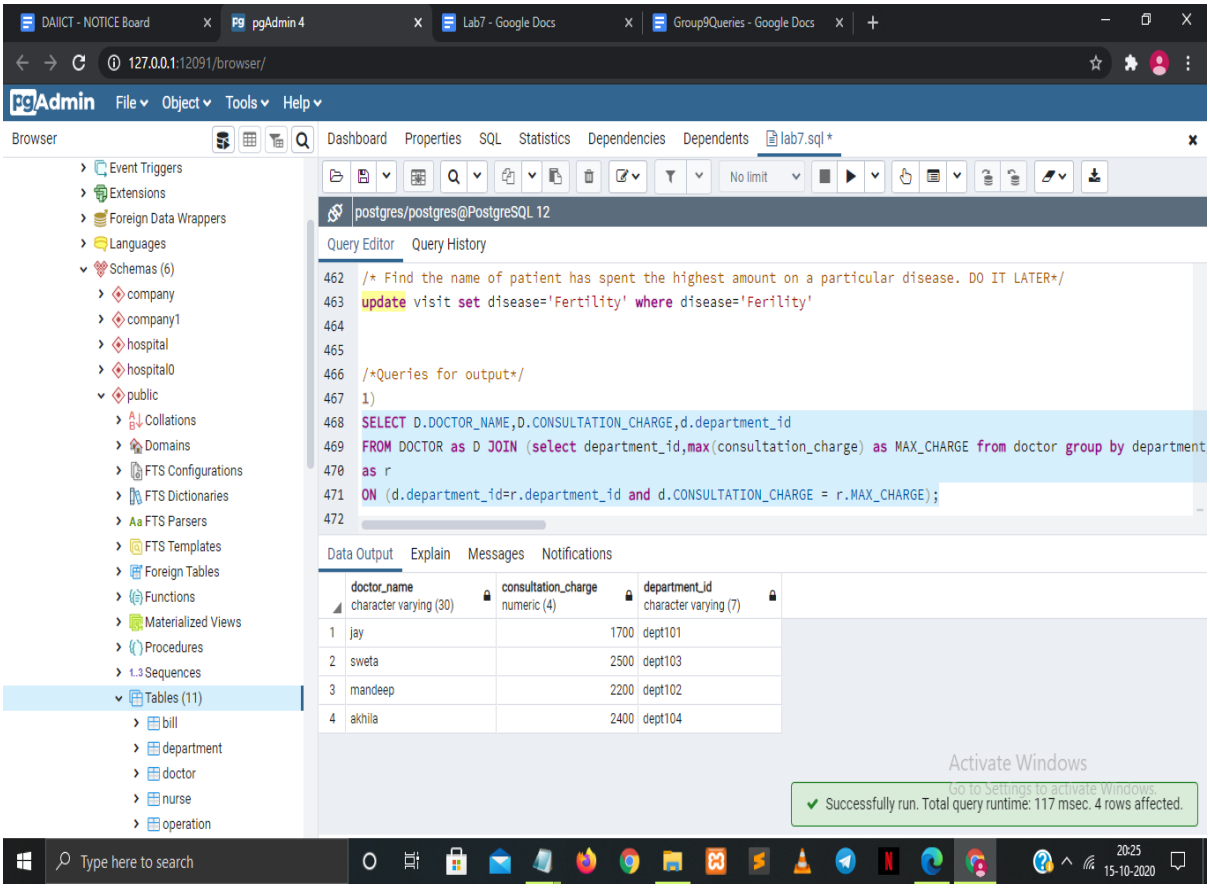
Relational Algebra:

$\pi_{D.DOCTOR\_NAME,\ D.CONSULTATION\_CHARGE,\ D.DEPARTMENT\_ID}(\rho(D,DOCTOR) \bowtie_{D.DEPARTMENT\_ID=R.DEPARTMENT\_ID\ AND\ D.CONSULTATION\_CHARGE=R.MAX\_CHARGE}(\rho(DEPARTMENT\_ID\mathcal{F}MAX(CONSULTATION\_CHARGE) \rightarrow MAX\_CHARGE(DOCTOR),R))$

SQL:

```
SELECT D.DOCTOR_NAME,D.CONSULTATION_CHARGE,d.department_id

FROM DOCTOR as D JOIN (select department_id,max(consultation_charge) as
MAX_CHARGE from doctor group by department_id)

as r

ON (d.department_id=r.department_id and d.CONSULTATION_CHARGE =
r.MAX_CHARGE);
```

OUTPUT:



2) Count the number of doctors in each department.

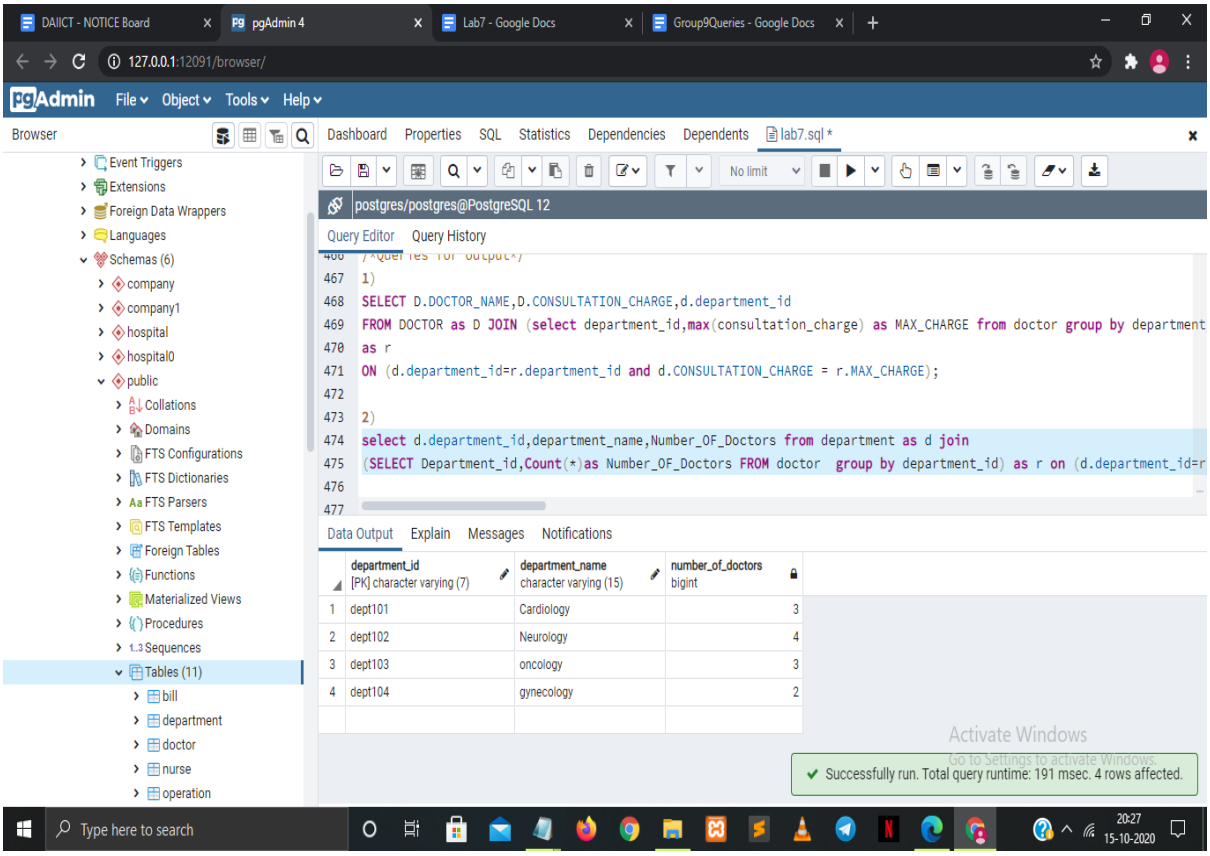
Relational Algebra:

π D.DEPARTMENT_ID, DEPARTMENT_NAME, NUMBER_OF_DOCTORS(ρ(D,DEPARTMENT)$\bowtie$D.DEPARTMENT_ID=R.DEPARTMENT_ID(ρ(DEPARTMENT_ID $\mathcal{F}$COUNT(*)$\rightarrow$NUMBER_OF_DOCTORS(DOCTOR),R))

SQL:

```
select d.department_id,department_name,Number_OF_Doctors from department as d join

(SELECT Department_id,Count(*)as Number_OF_Doctors FROM doctor  group by
department_id) as r on (d.department_id=r.department_id);
```

OUTPUT



3) List the details of nurses working in a particular shift.

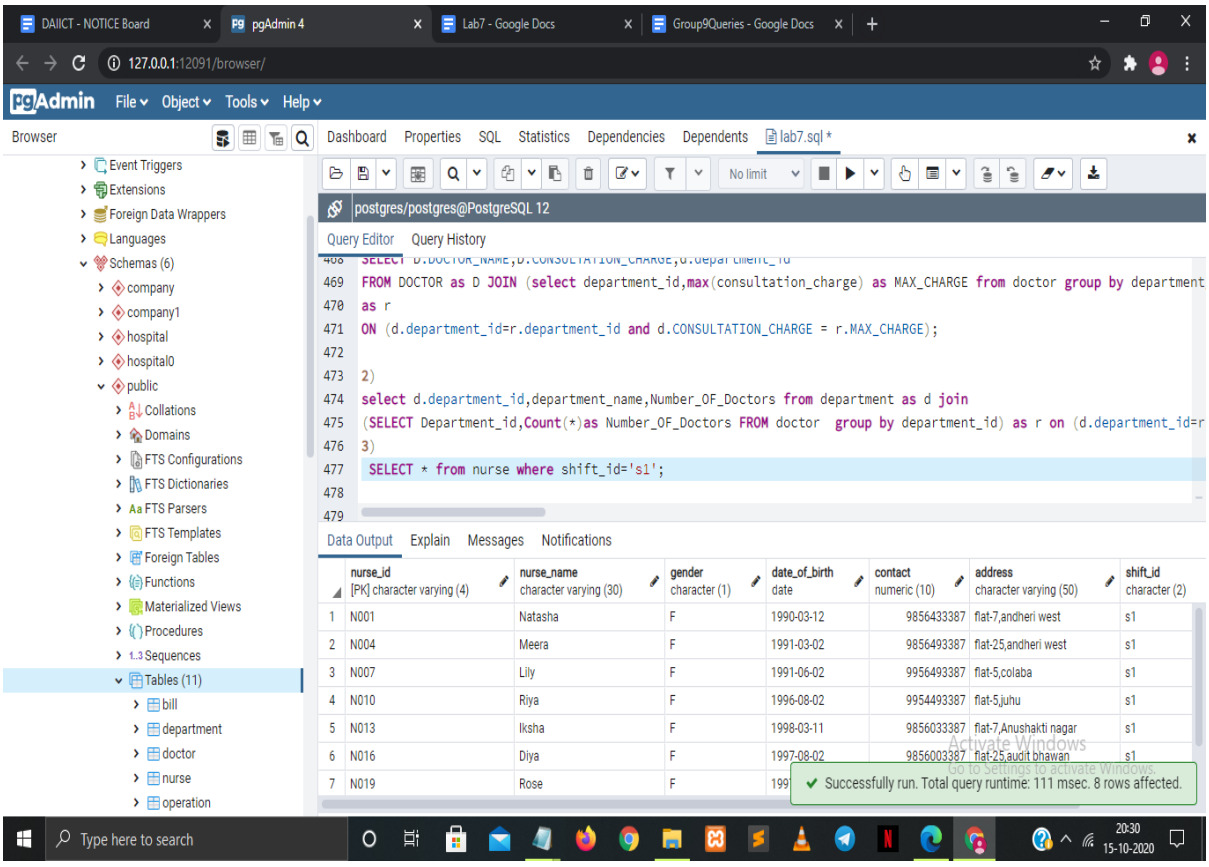
Relational Algebra:

$\Pi_{*}(\sigma_{SHIFT\_ID='S1'}(NURSE))$

SQL:

```
SELECT * from nurse where shift_id='s1';
```

OUTPUT:



4) List the name of doctors who have done more than 2 operations in a month.

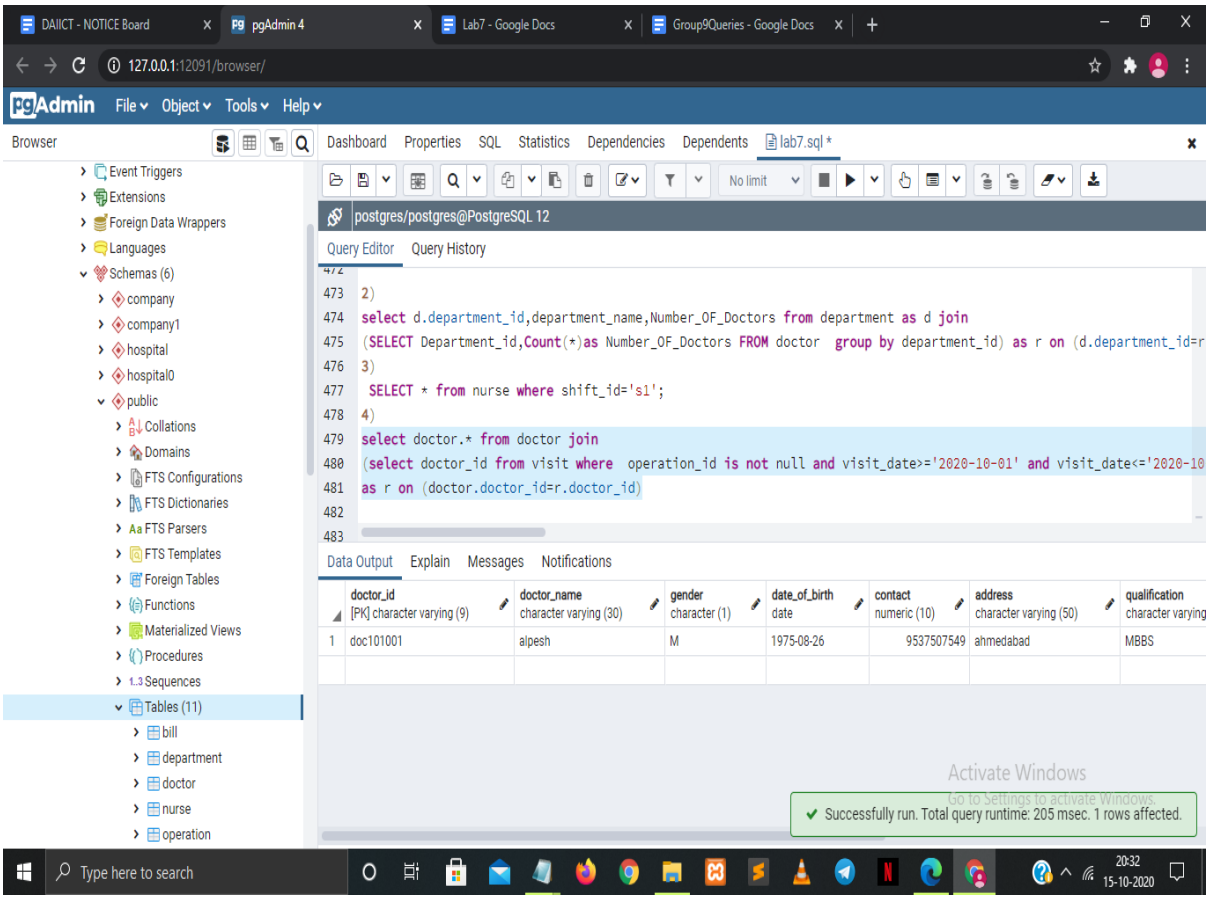
Relational Algebra:

$\Pi_{doctor.*}(doctor \bowtie_{doctor.doctor\_id=r.doctor\_id}(\rho(r,\sigma_{count(*)>2 \text{ and operation\_id is not null and visit\_date>='2020-10-01' and visit\_date<='2020-10-31'}}(doctor\_id\mathcal{F}_{count(*)}(visit)))$

SQL:

```
select doctor.* from doctor join

(select doctor_id from visit where  operation_id is not null and visit_date>='2020-10-01' and visit_date<='2020-10-31' group by doctor_id having count(*)>2)

as r on (doctor.doctor_id=r.doctor_id);
```

OUTPUT:



5) List the name of doctors working on a particular shift.

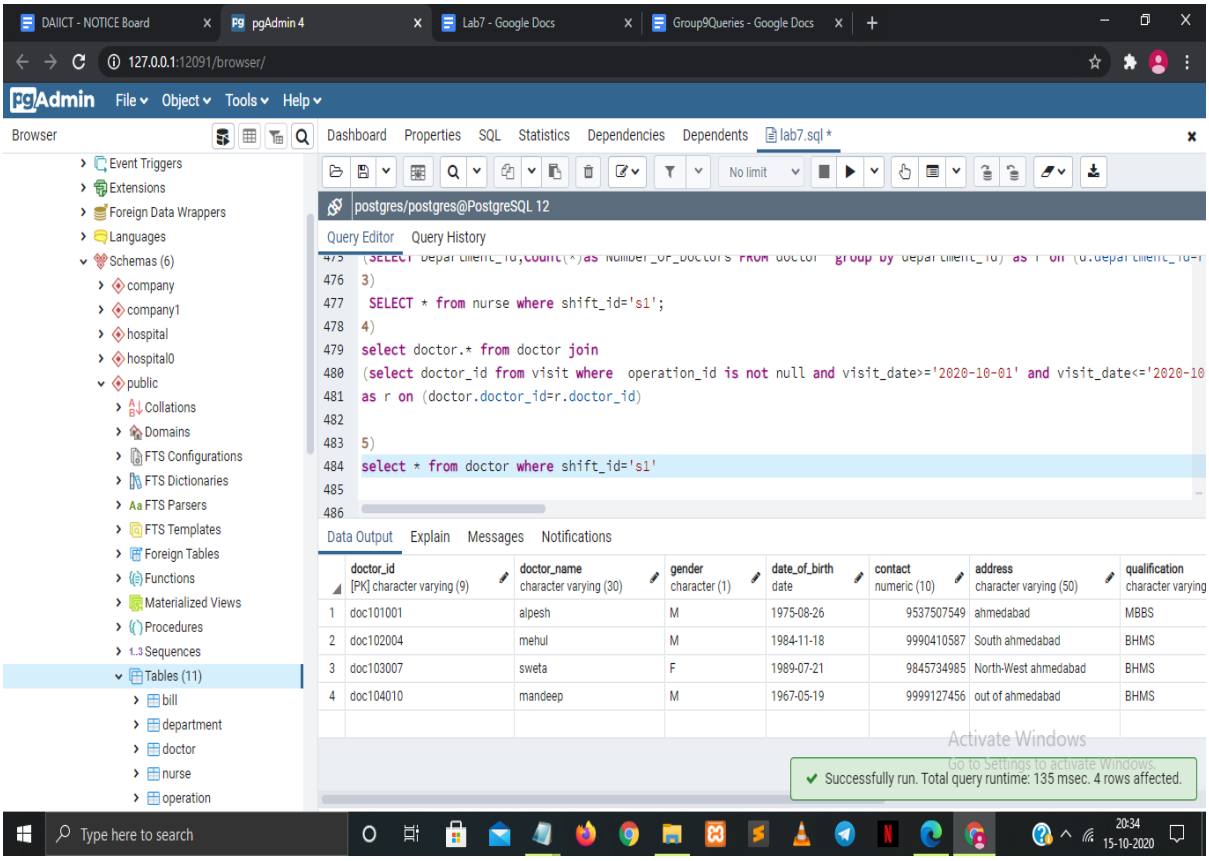
Relational Algebra:

$\sigma_{shift\_id=s1}(doctor)$

SQL:

```
select * from doctor where shift_id='s1';
```

OUTPUT:



6) Given the doctor's id, retrieve the details of patients he has treated so far.

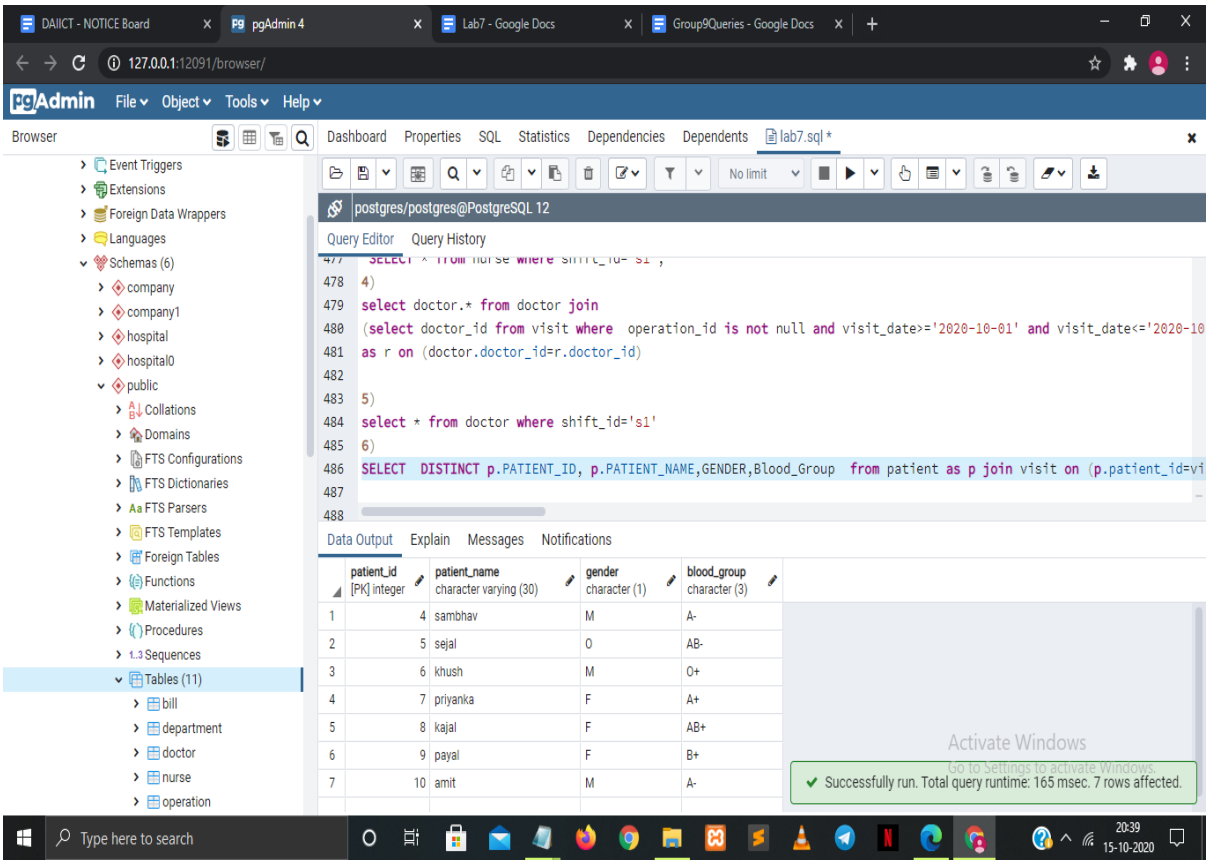
Relational Algebra:

$\pi_{\text{distinct P.PATIENT\_ID, P.PATIENT\_NAME,GENDER,Blood\_Group}}(\sigma_{\text{DOCTOR\_ID='DOC101001'}}(\rho(P,\text{PATIENT}) \bowtie_{\text{P.PATIENT\_ID=VISIT.PATIENT\_ID}} (\text{VISIT}))$

SQL:

```
SELECT  DISTINCT p.PATIENT_ID, p.PATIENT_NAME,GENDER,Blood_Group  from patient as p join visit on (p.patient_id=visit.patient_id) where doctor_id='doc101001';
```

OUTPUT:



7) Retrieve details of nurses assigned to a particular room.

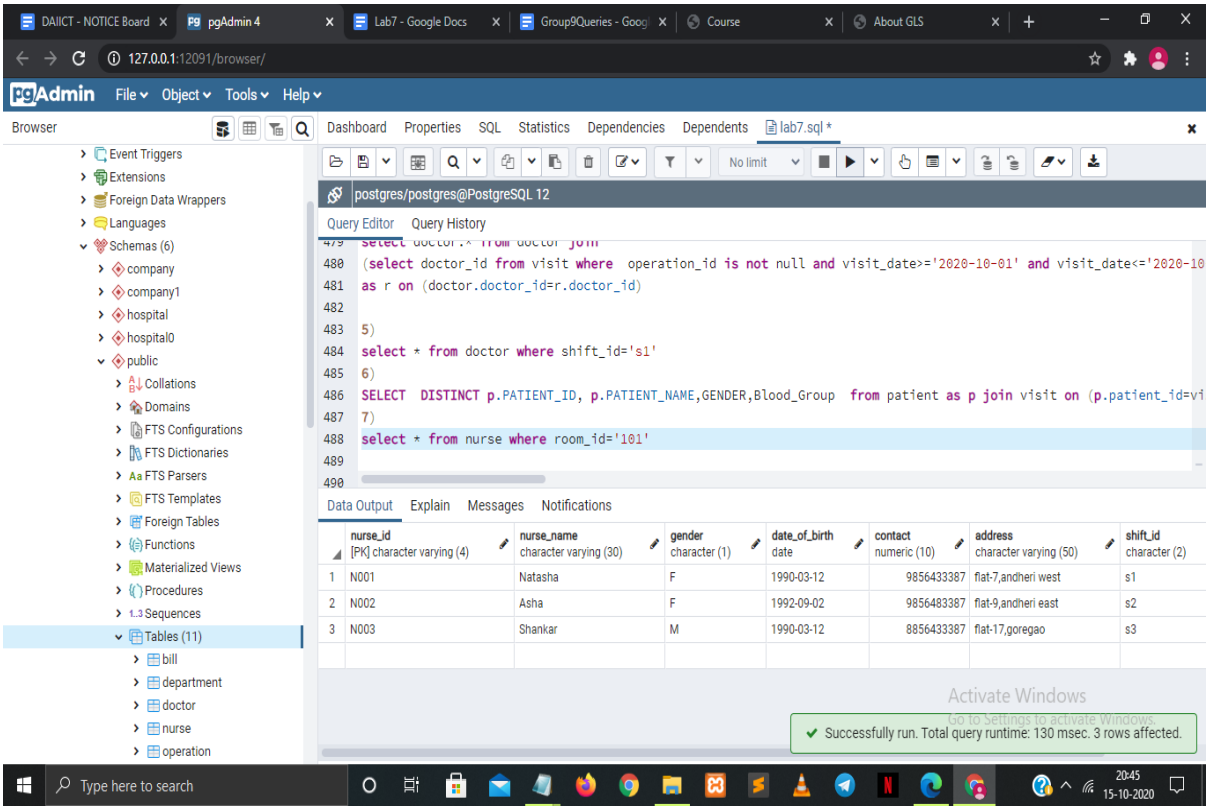
Relational Algebra:

$\sigma_{room\_id=101}(nurse)$

SQL:

```
select * from nurse where room_id='101';
```

OUTPUT:



8) Count the number of patients preferring a particular category of room.(to understand which room category is preferred so the infrastructure of rooms in hospital is adjusted accordingly)

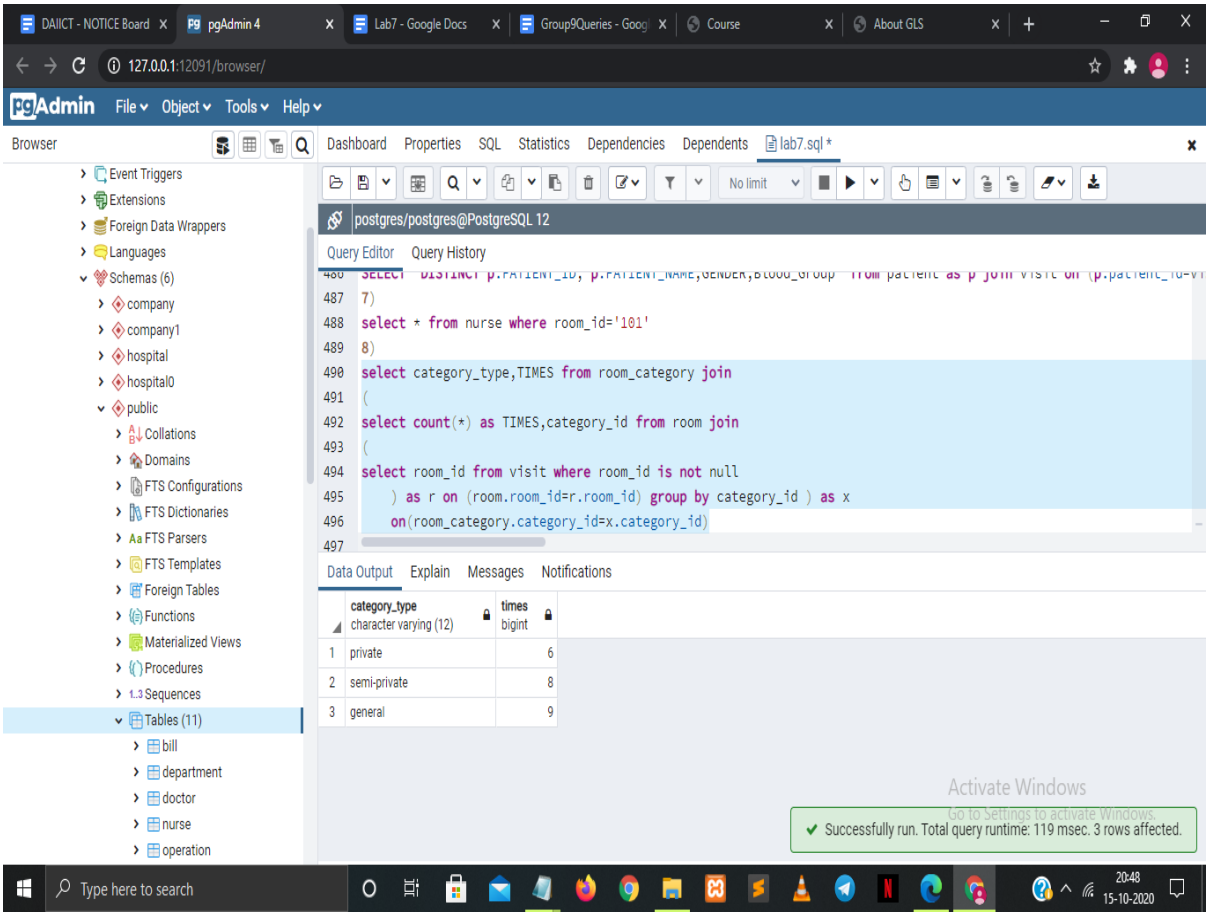
Relational Algebra:

$\pi_{category\_type,times}(\rho(room\_category) \bowtie_{room\_category.category\_id=x.category\_id} (\rho(\mathcal{F}_{COUNT(*)\rightarrow times,category\_id}(x,room) \bowtie_{room.room\_id=r.room\_id} (\rho(\sigma_{room\_id\ not\ null}(r,visit))))$

SQL:

```
select category_type,TIMES from room_category join
(
select count(*) as TIMES,category_id from room join
(
select room_id from visit where room_id is not null
        ) as r on (room.room_id=r.room_id) group by category_id ) as x
        on(room_category.category_id=x.category_id);
```

OUTPUT:



9) From date A to B retrieve the data of visited patients.

Relational Algebra:

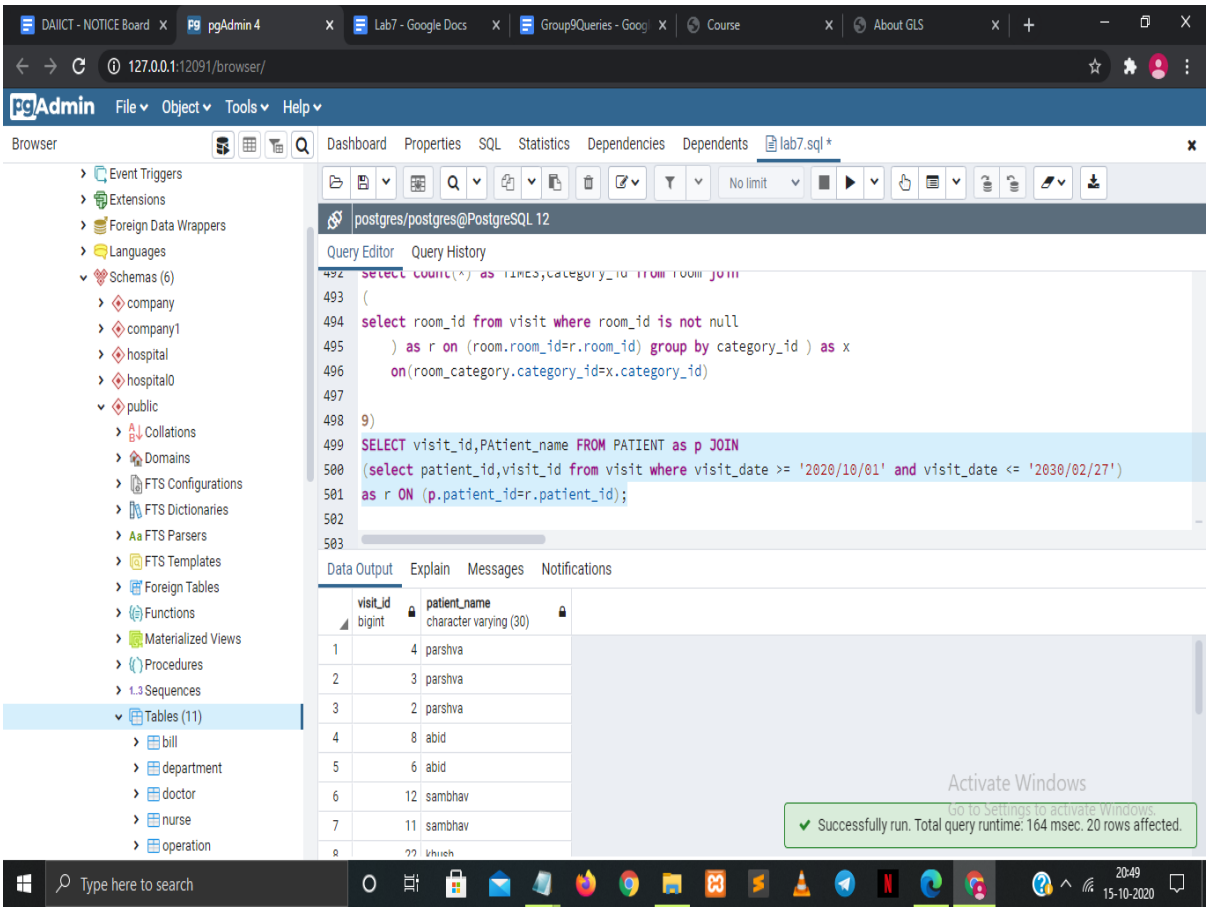
$\pi_{VISIT\_ID,\ PATIENT\_NAME}(\rho(P,PATIENT) \bowtie$

$P.PATIENT\_ID=R.PATIENT\_ID\ \rho(R,\pi_{VISIT\_ID,\ PATIENT\_NAME}(\sigma_{visit\_date >= '2020/10/01' and visit\_date <= '2030/02/27'}(VISIT)))$

SQL:

```
SELECT visit_id,PAtient_name FROM PATIENT as p JOIN

(select patient_id,visit_id from visit where visit_date >= '2020/10/01' and visit_date <= '2030/02/27')

as  r ON (p.patient_id=r.patient_id);
```

OUTPUT:



10) Get the admit history of a particular patient.

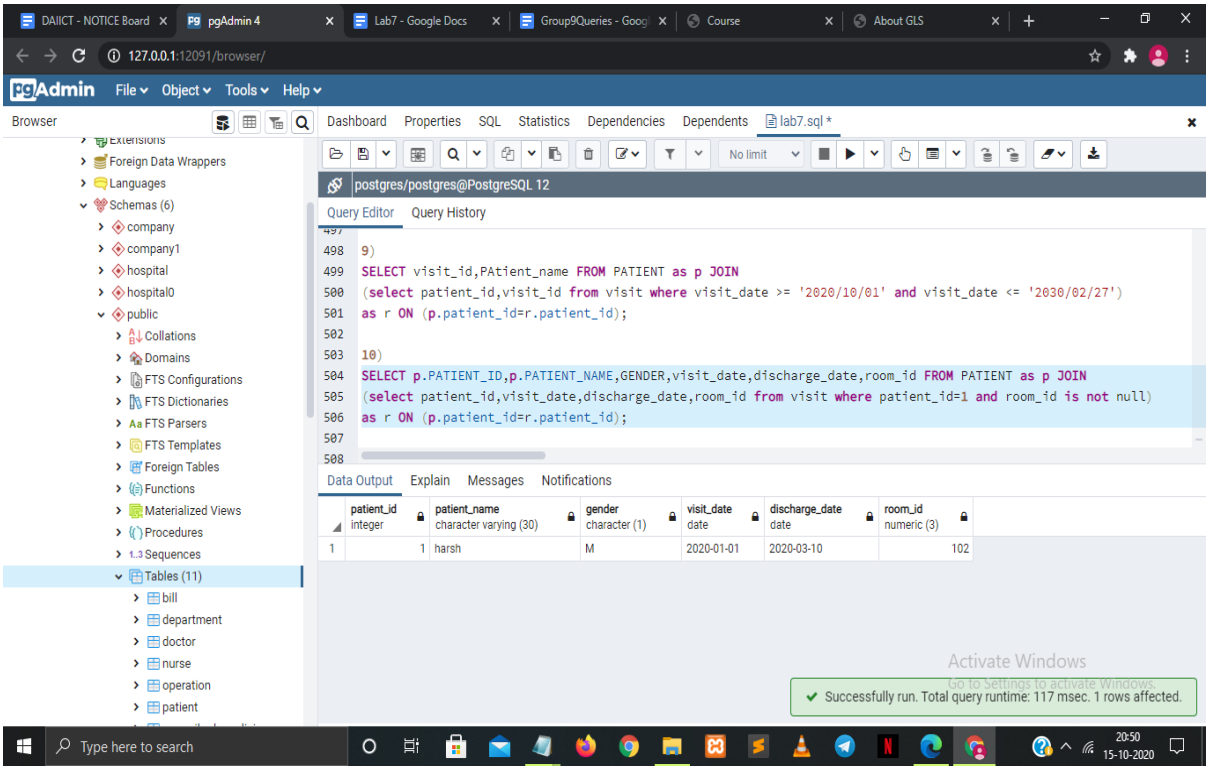
Relational Algebra:

$\pi_{P.PATIENT\_ID,\ P.PATIENT\_NAME,\ GENDER,\ VISIT\_DATE,\ DISCHARGE\_DATE,\ ROOM\_ID}(\rho(P,PATIENT) \bowtie_{P.PATIENT\_ID=R.PATIENT\_ID} \rho(R,\sigma_{PATIENT\_ID=XXX\ AND\ ROOM\ ID\ IS\ NOT\ NULL}(VISIT))$

SQL:

```
SELECT p.PATIENT_ID,p.PATIENT_NAME,GENDER,visit_date,discharge_date,room_id
FROM PATIENT as p JOIN

(select patient_id,visit_date,discharge_date,room_id from visit where patient_id=1 and
room_id is not null)

as r ON (p.patient_id=r.patient_id);
```

OUTPUT:



11) List of medicines prescribed to a patient on a particular visit.

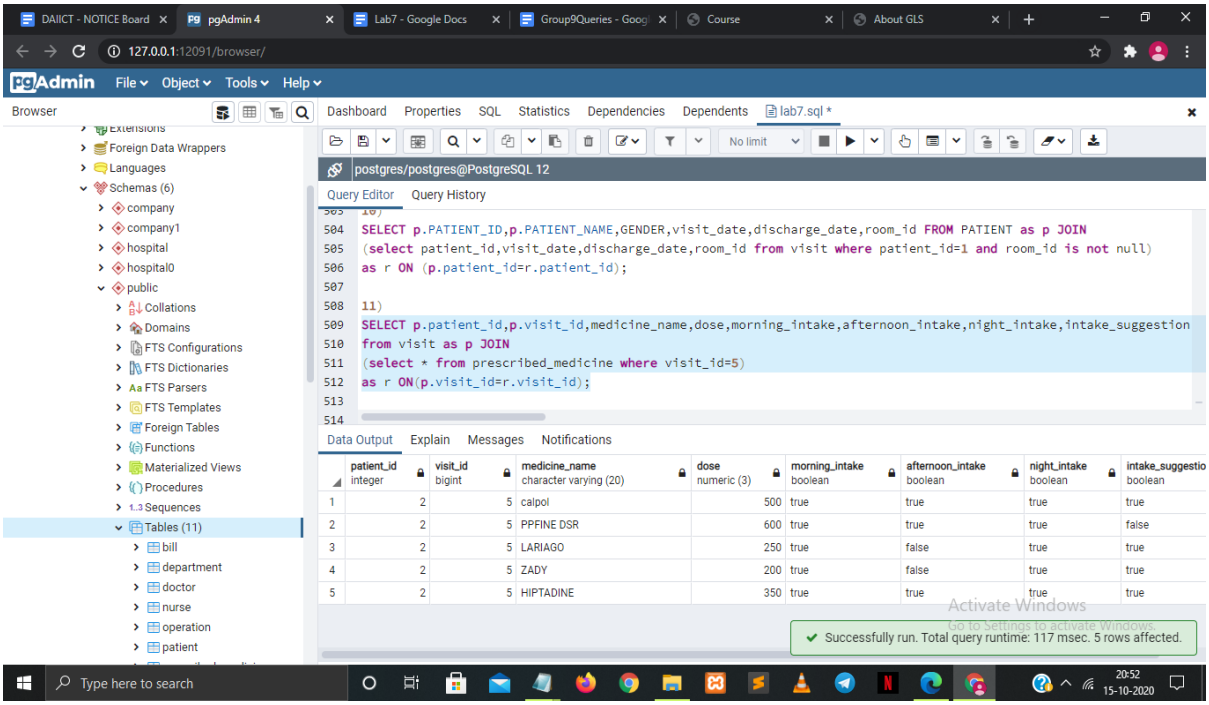
Relational Algebra:

$\pi_{P.PATIENT\_ID,\ P.VISIT\_ID,\ MEDICINE\_NAME,\ DOSE,MORNING\_INTAKE,\ AFTERNOON\_INTAKE,\ NIGHT\_INTAKE,\ INTAKE\_SUGGESTION}\ (\rho(P,VISIT) \bowtie_{P.VISIT\_ID=R.VISIT\_ID} \rho(R,\sigma_{VISIT\_ID=X}(PRESCRIBED\_MEDICINE)))$

SQL:

```
SELECT
p.patient_id,p.visit_id,medicine_name,dose,morning_intake,afternoon_intake,night_intake,intake_suggestion

from visit as p JOIN

(select * from prescribed_medicine where visit_id=1)

as r on (p.visit_id=r.visit_id);
```

OUTPUT:



12) List of the disease of a particular patient in previous visits.

Relational Algebra:

$\pi_{patient.patient\_id,patient\_name,disease}(\rho\ (r,\ \sigma_{patient\_id=1}(visit))$

$\bowtie_{<r.patient\_id=patient.patient\_id>}$ patient)

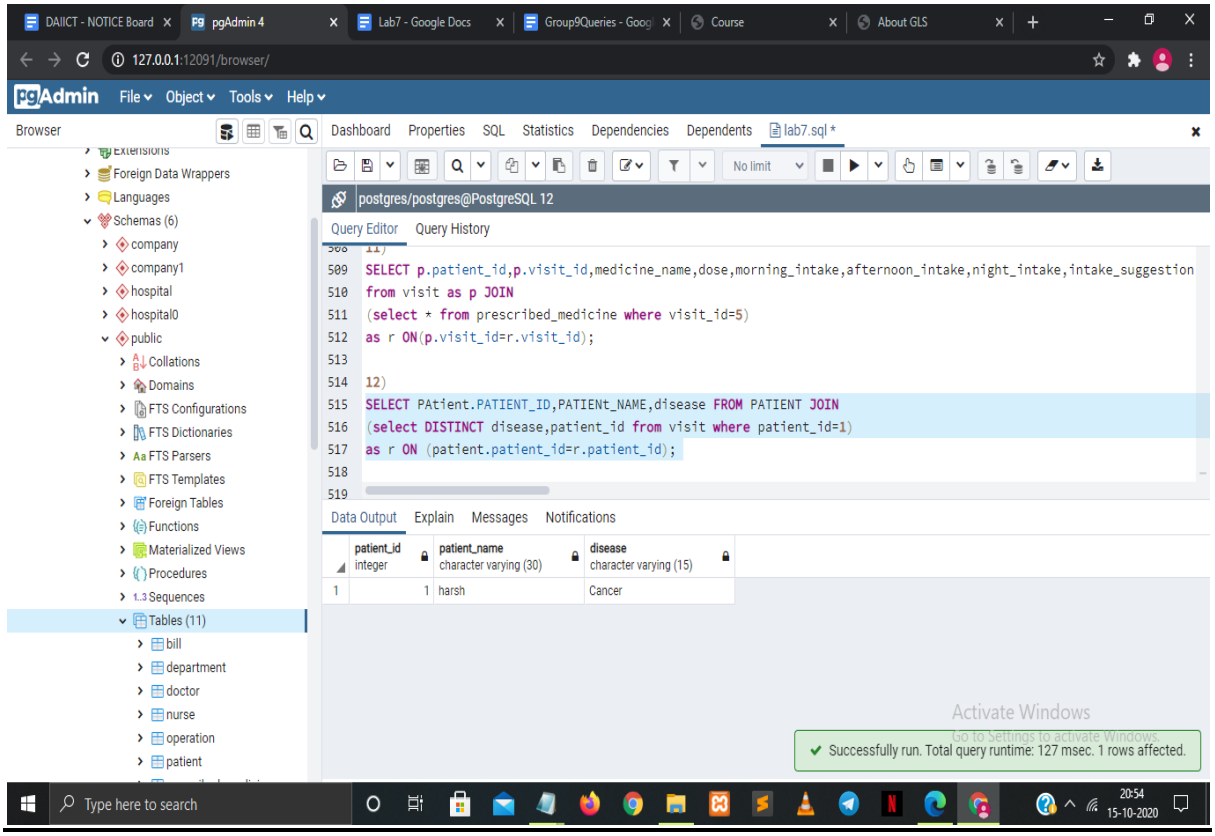
SQL:

SELECT PAtient.PATIENT_ID,PATIENt_NAME,disease FROM PATIENT JOIN

(select DISTINCT disease,patient_id from visit where patient_id=1)

as r ON (patient.patient_id=r.patient_id);

OUTPUT:



13) Show the details of patients treated by a particular doctor.

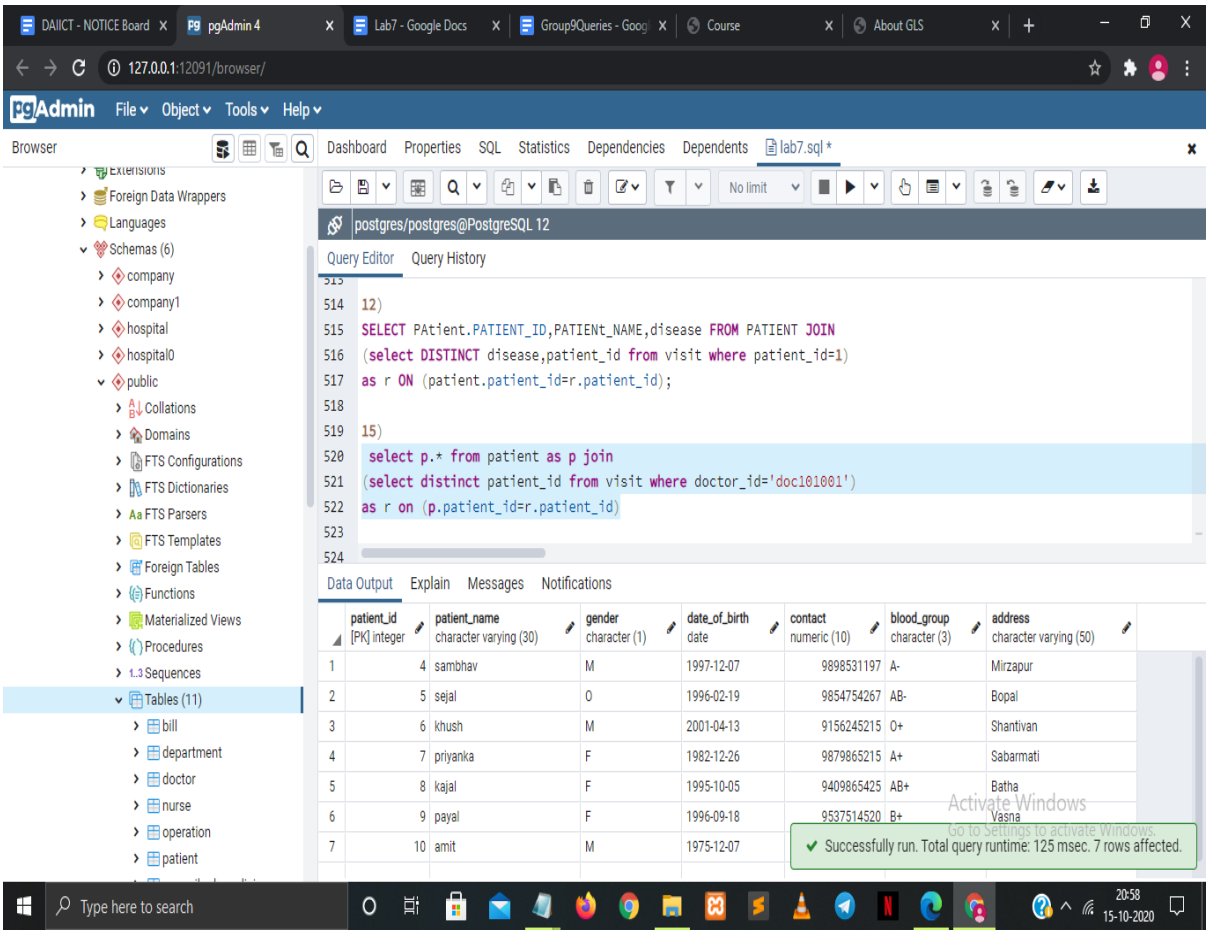
Relational Algebra:

$\Pi_{*}(\rho(p,patient) \bowtie_{p.patient\_id=r.patient\_id} \rho(r, \Pi_{distinct\ patient\_id}(\sigma_{doctor\_id='doc101001'}(visit)))$

SQL:

```
select p.* from patient as p join
(select distinct patient_id from visit where doctor_id='doc101001')
as r on (p.patient_id=r.patient_id);
```

OUTPUT:



14) Show the details of patients operated by a particular doctor.

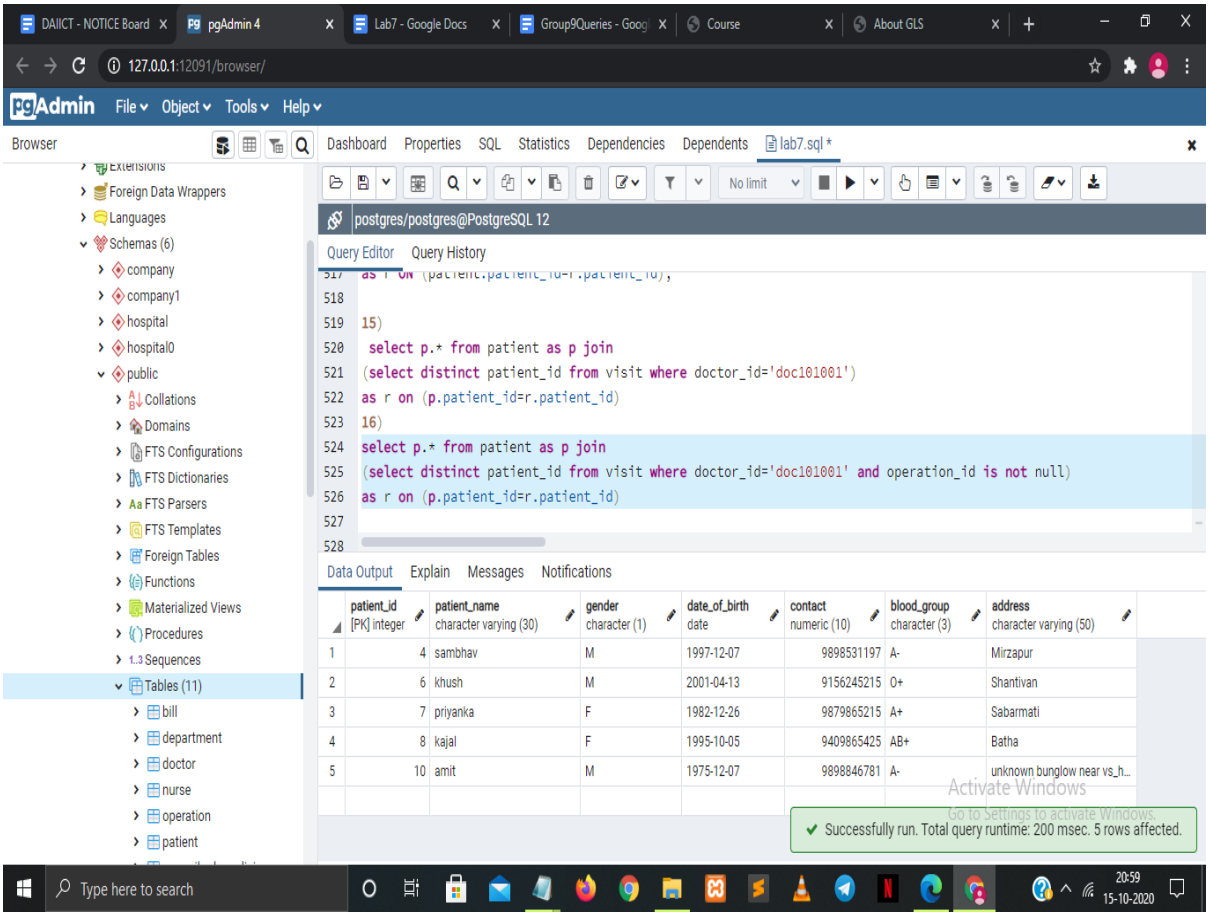
Relational Algebra:

$\Pi_{p.*}(\rho(p,patient) \bowtie_{p.patient\_id=r.patient\_id} \rho(r, \Pi_{distinct\ patient\_id} \sigma_{doctor\_id='doc101001' \text{ and } operation\_id \text{ is not } null}(visit)))$

SQL:

```
SELECT  p.* FROM patient as p JOIN

(SELECT DISTINCT patient_id FROM visit where doctor_id='doc101001' and operation_id
is not null)

as r ON (p.patient_id=r.patient_id);
```

OUTPUT:



15) Count the number of visits of a particular patient as per his disease.

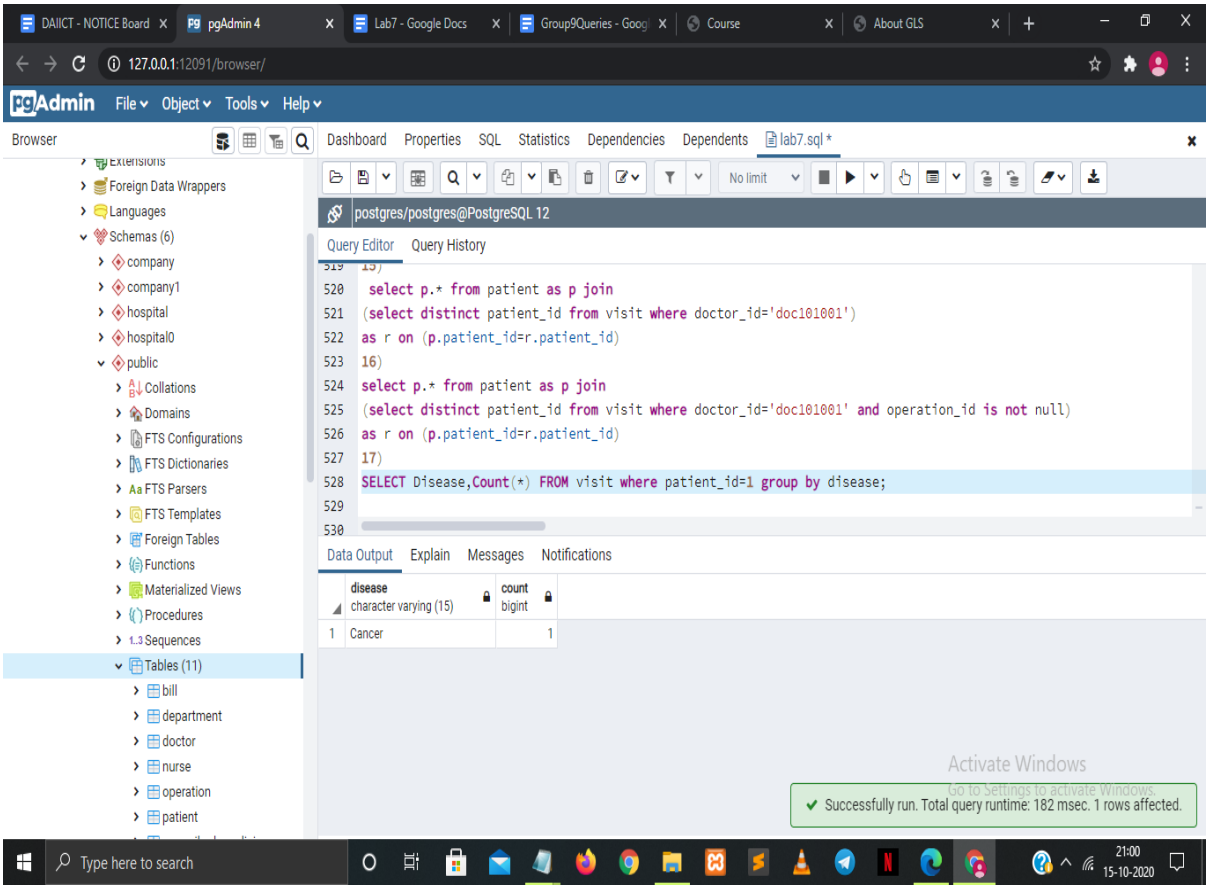
Relational Algebra:

$_{\text{disease}}\mathcal{F}_{\text{count}(*)}\ (\sigma_{\text{patient\_id}=1}(\text{visit}))$

SQL:

SELECT Disease,Count(*) FROM visit where patient_id=1 group by disease;

OUTPUT:



16) List the details of the doctor working in a particular department and shift.

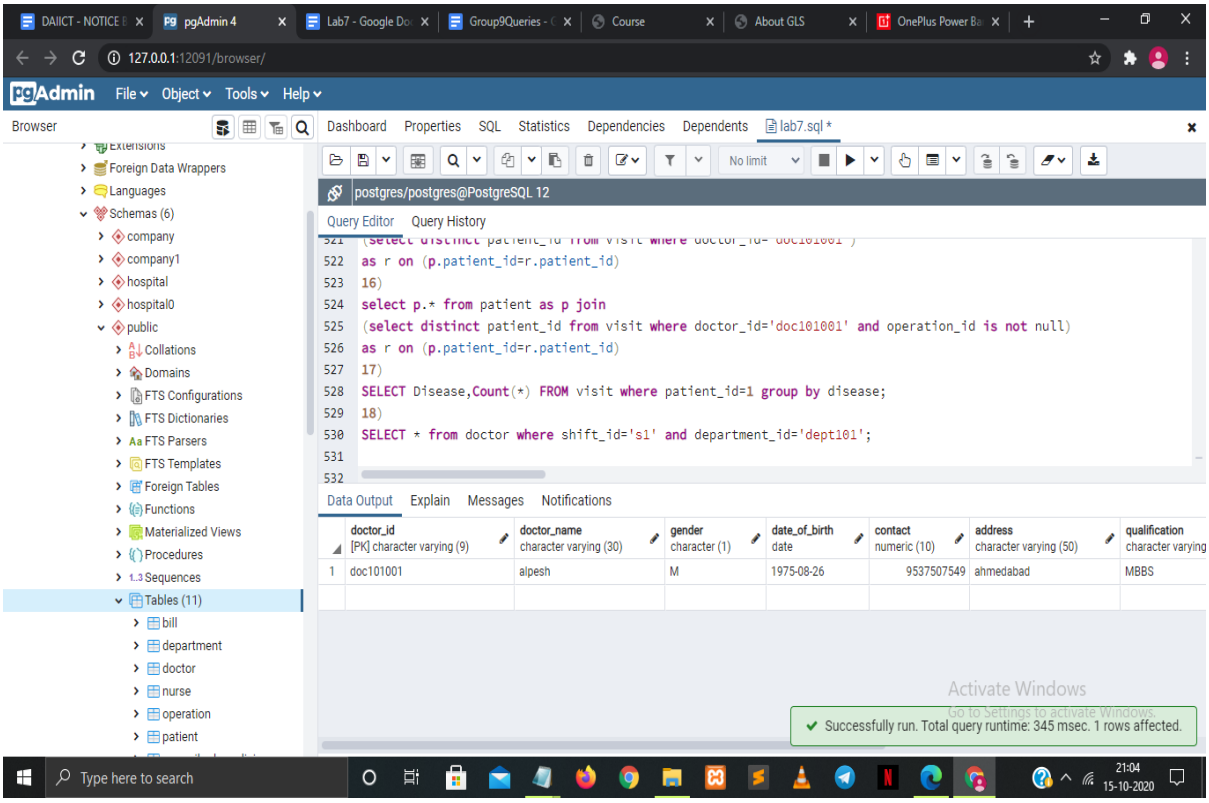
Relational Algebra:

$\Pi_{*}(\sigma_{SHIFT\_ID='S1' \text{ AND } DEPARTMENT\_ID='DEPT101}(DOCTOR))$

SQL:

```
SELECT * from doctor where shift_id='s1' and department_id='dept101';
```

OUTPUT:



17) Get the revenue generated in a particular month.

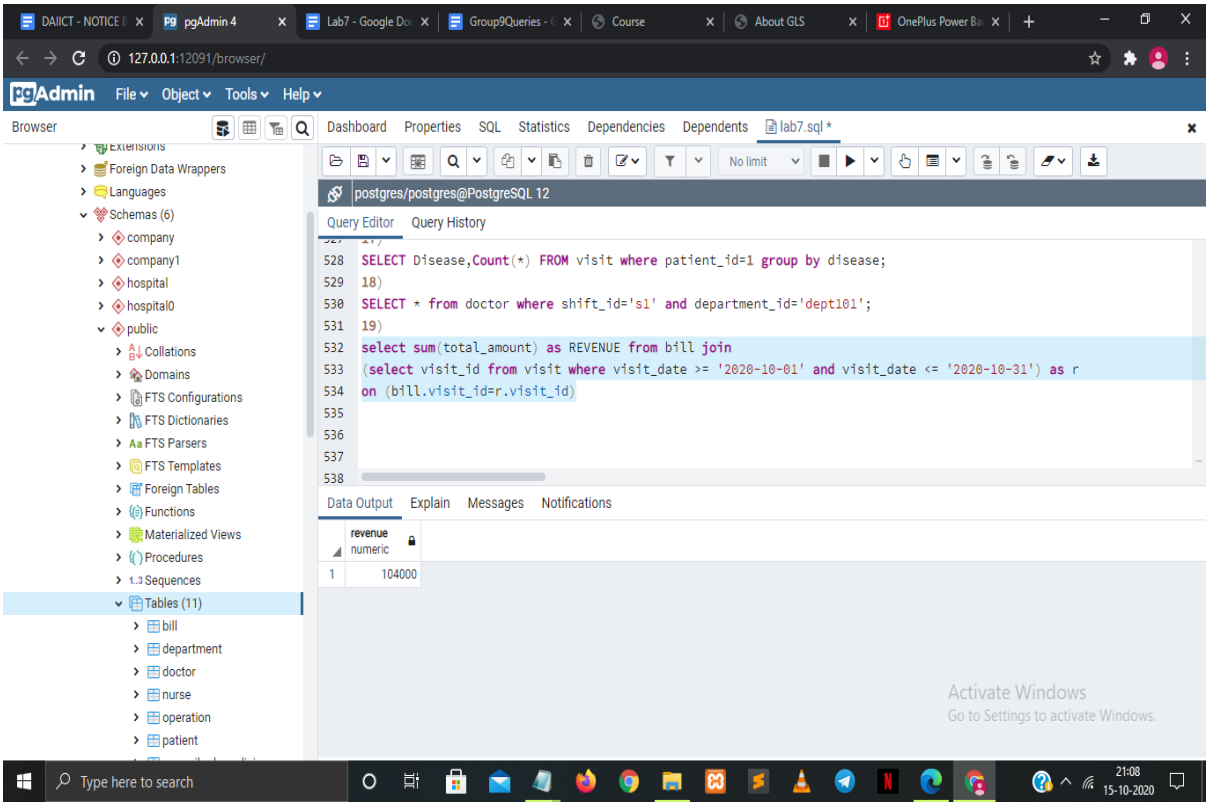
Relational Algebra:

$\mathcal{F}_{sum(total\_amount)\rightarrow revenue}(bill \bowtie_{bill.visit\_id=r.visit\_id} \rho(r, \Pi_{visit\_id}(\sigma_{visit\_date>=2020\text{-}10\text{-}01 \text{ and } visit\_date<=2020\text{-}10\text{-}31}(visit)))$

SQL:

```
SELECT SUM(total_amount) as REVENUE FROM bill JOIN
(SELECT visit_id FROM visit where visit_date >= '2020-10-01' and visit_date <= '2020-10-31') as r
ON (bill.visit_id=r.visit_id);
```

OUTPUT:



18) Get the last prescription of a particular patient.

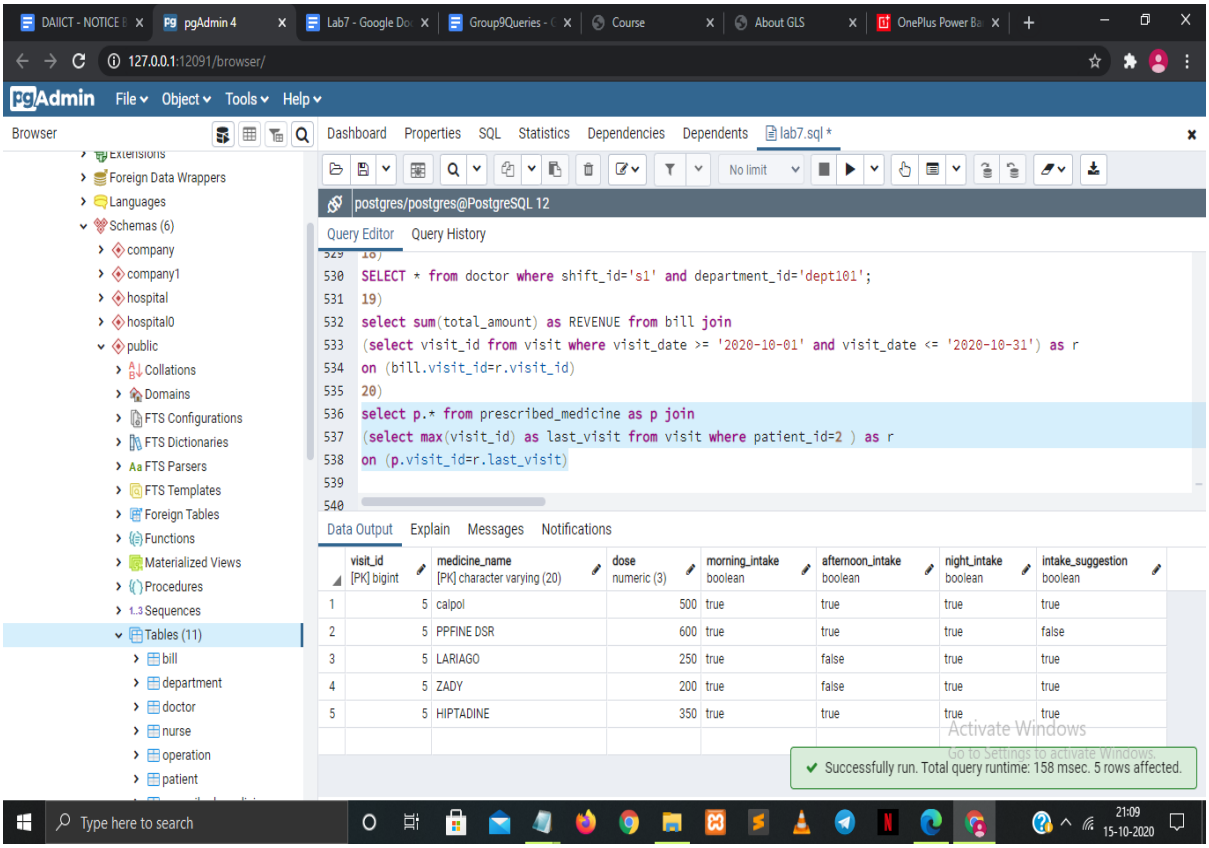
Relational Algebra:

$\Pi_{p.*}(\rho(p,prescribed\_medicine) \bowtie_{p.visit\_id=r.last\_visit} \rho(r,\sigma_{patient\_id=2}(\mathcal{F}_{max(visit\_id)\rightarrow last\_visit}(visit))))$

SQL:

```
SELECT p.* FROM prescribed_medicine as p JOIN
(SELECT MAX(visit_id) as last_visit FROM visit where patient_id=2 ) as r
ON (p.visit_id=r.last_visit);
```

OUTPUT:



19) Total revenue contributed by a doctor in a financial year.

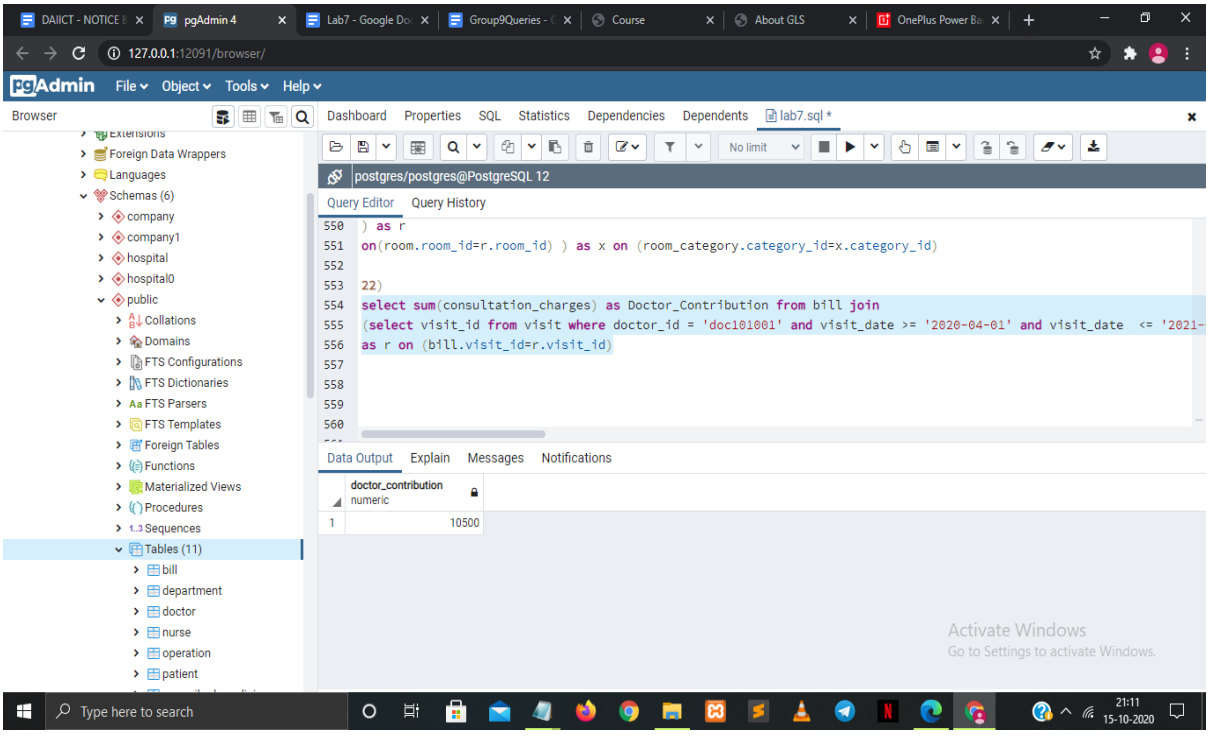
Relational Algebra:

$\mathcal{F}$ sum(consultation_charges)→doctor_contribution(bill) ⋈ bill.visit_id=r.visit_id ρ(r, πvisit_id(σ doctor_id = 'doc101001' and visit_date >= '2020-04-01' and visit_date <= '2021-03-31'(visit)))

SQL:

```
SELECT SUM(consultation_charges) as Doctor_Contribution FROM bill JOIN
(SELECT visit_id FROM visit where doctor_id = 'doc101001' and visit_date >= '2020-04-01'
and visit_date  <= '2021-03-31')as r
ON (bill.visit_id=r.visit_id);
```

OUTPUT:



20) Count the number of particular operations in a year.

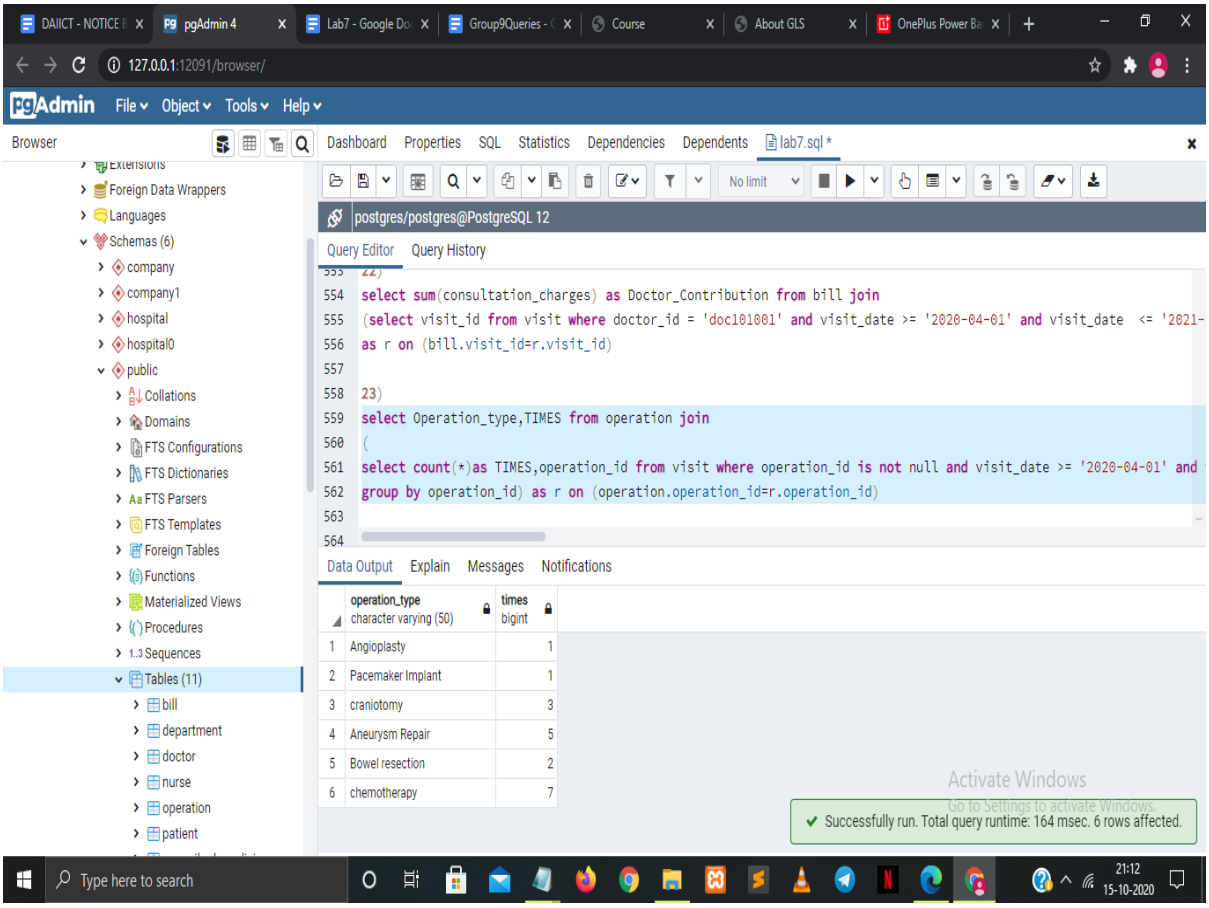
Relational Algebra:

$\pi_{operation\_type,times}$ (operation$\bowtie_{operation.operation_id=r.operation_id}$ $\rho(r,\sigma$ operation_id not null and visit_date>='2020-04-01' and visit_date <= '2021-03-31' (operation_id $\mathcal{F}$ count(*) $\rightarrow$ times (visit))))

SQL:

```
SELECT Operation_type,TIMES FROM operation JOIN
(SELECT count(*)as TIMES,operation_id FROM visit where operation_id is not null and
visit_date >= '2020-04-01' and visit_date  <= '2021-03-31'group by operation_id) as r
ON (operation.operation_id=r.operation_id);
```

OUTPUT:



21) Count number of operations done by doctors in a year.

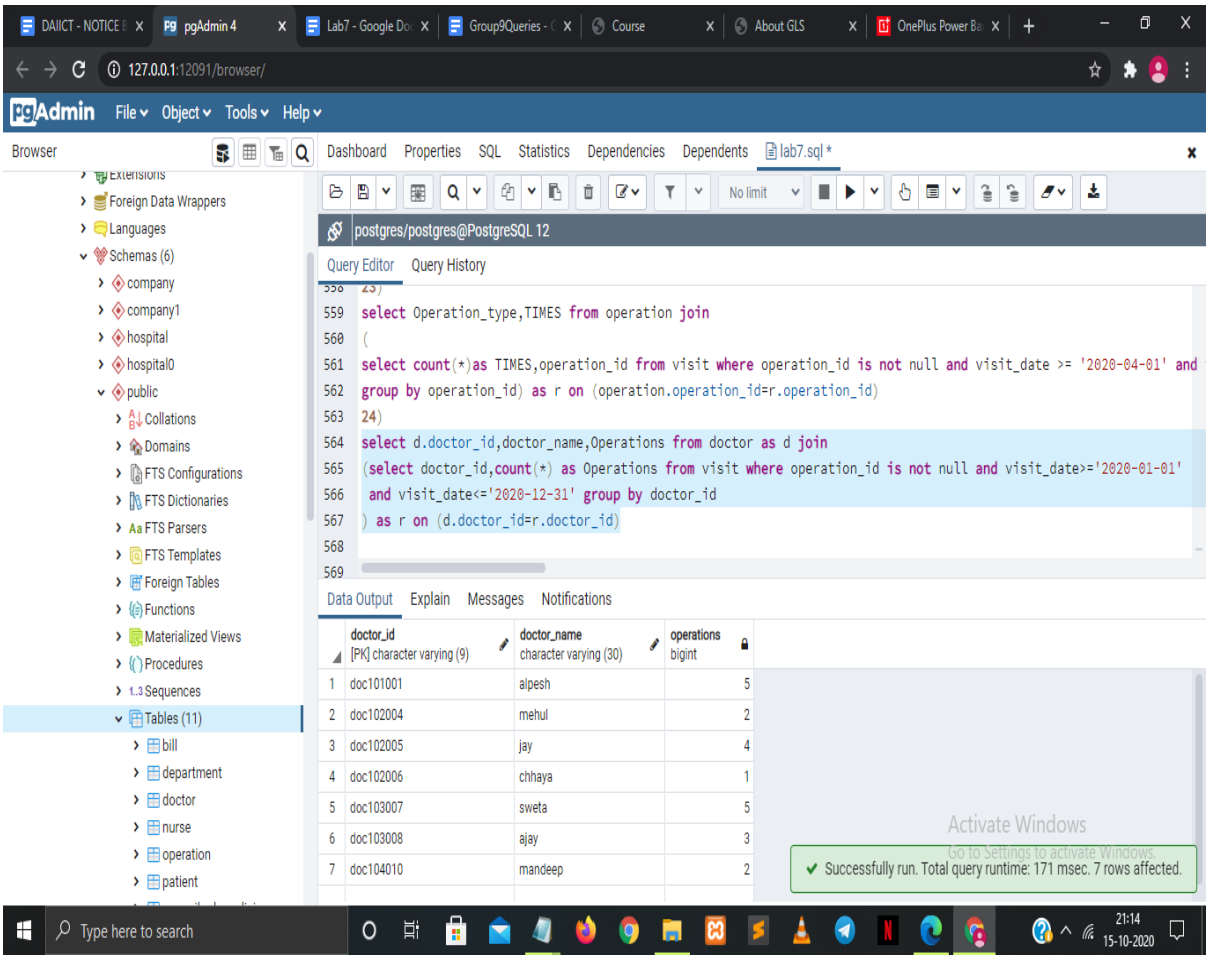
Relational Algebra:

$\Pi_{d.doctor\_id,doctor\_name,Operations}(\rho(d,doctor) \bowtie_{d.doctor\_id=r.doctor\_id} \rho(r,\sigma_{operation\_id\ not\ null\ and\ visit\_date>=2020\text{-}01\text{-}01\ and\ visit\_date<=2020\text{-}12\text{-}31}(\ doctor\_id\mathcal{F}_{count(*)\rightarrow operations}(visit)))$

SQL:

```
select d.doctor_id,doctor_name,Operations from doctor as d join
(select doctor_id,count(*) as Operations from visit where operation_id is not null and
visit_date>='2020-01-01' and visit_date<='2020-12-31' group by doctor_id ) as r
on (d.doctor_id=r.doctor_id);
```

OUTPUT:



22) Count the number of patients who took appointments with at least two different physicians.

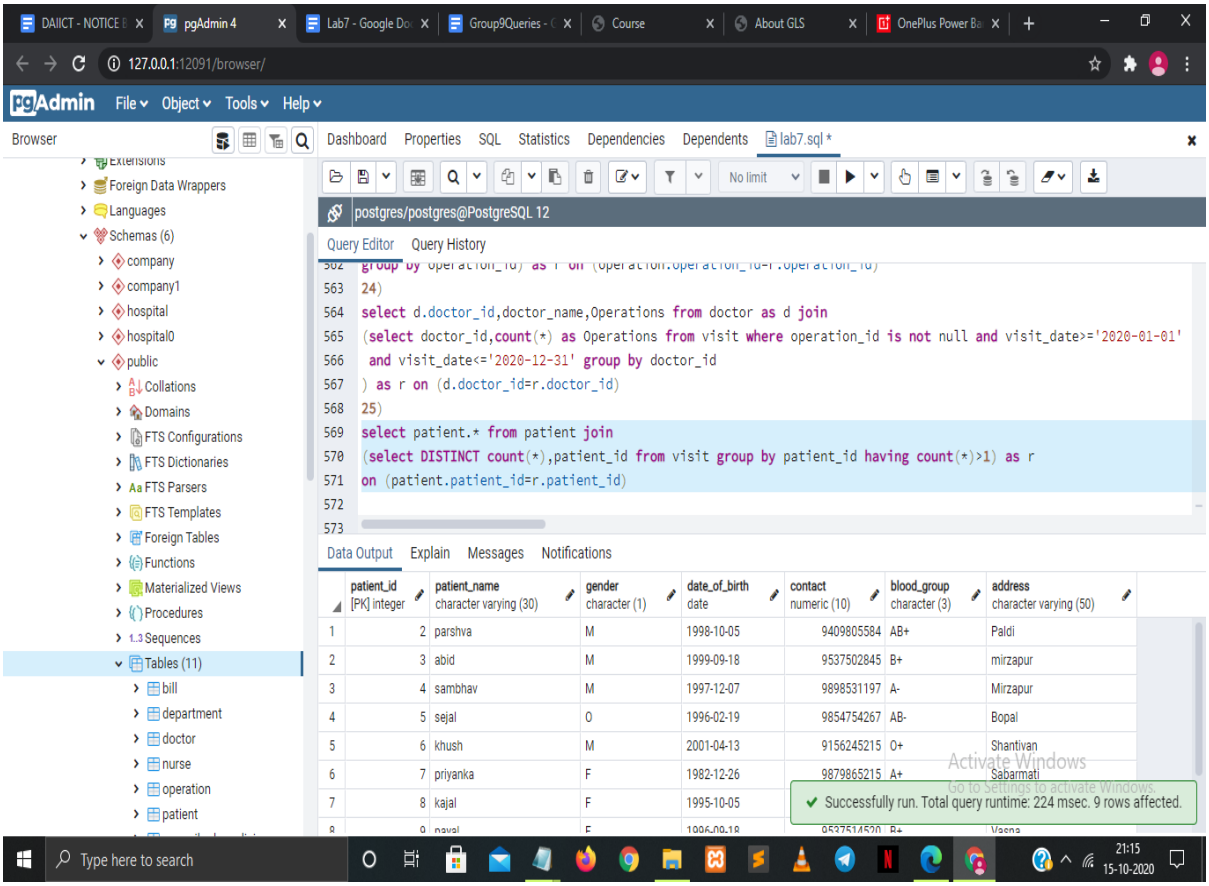
Relational Algebra:

$\Pi_{patient.*}(patient) \bowtie_{patient.patient\_id=r.patient\_id} \rho(r, \sigma_{count(*)>1}(patient\_id \mathcal{F}_{distinct\ count(*)} (visit)))$

SQL:

```
SELECT patient.* FROM patient JOIN
(SELECT  DISTINCT count(*),patient_id FROM visit GROUP BY patient_id having
count(*)>1) as r
ON (patient.patient_id=r.patient_id) ;
```

OUTPUT:



23) Obtain the names of all patients whose operation is done by a doctor who is head of a department.

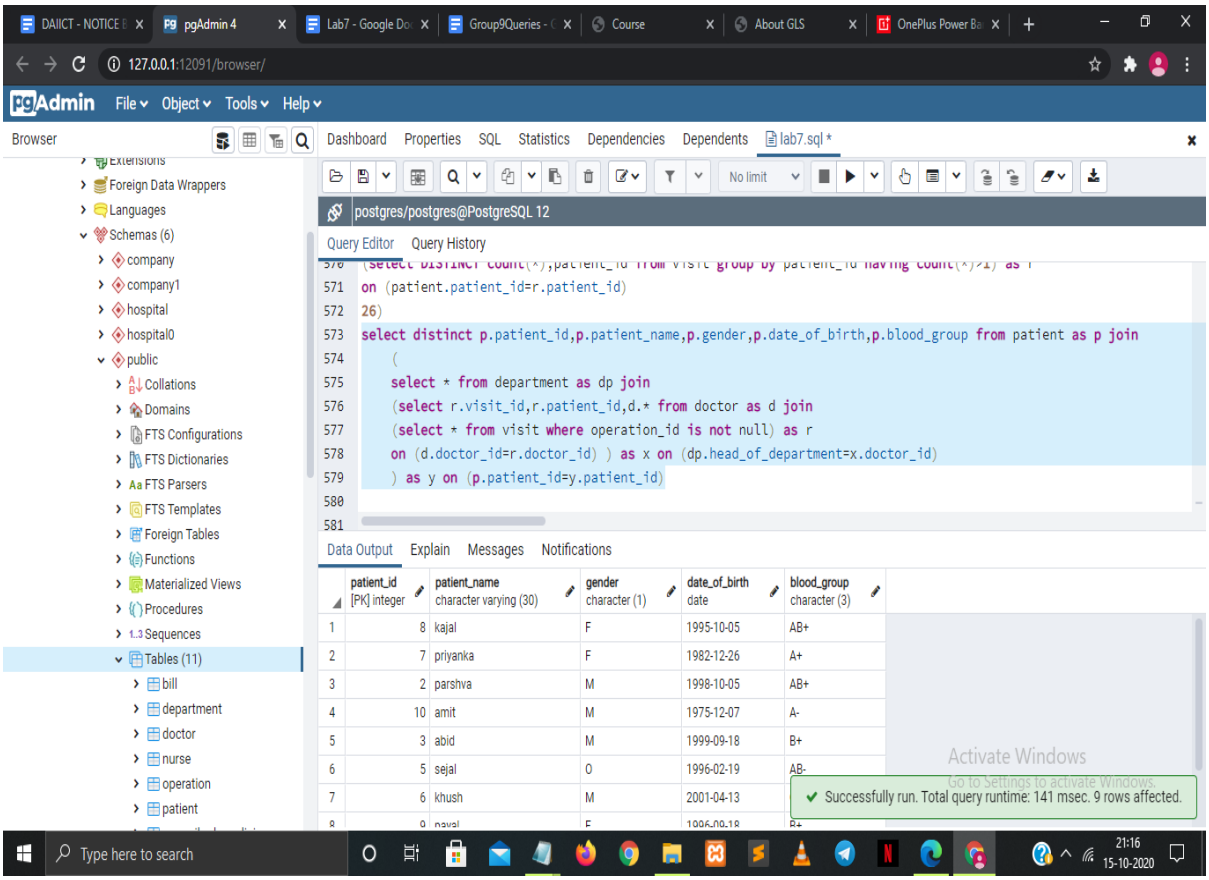
Relational Algebra:

$\Pi_{\text{distinct p.patient\_id,p.patient\_name,p.gender,p.date\_of\_birth,p.blood\_group}}(\rho(p,\text{patient}) \bowtie_{\text{p.patient\_id=y.patient\_id}} \rho(dp,\Pi_{*}(\text{department})) \bowtie_{\text{dp.head\_of\_department=x.doctor\_id}} \rho(d, \Pi_{\text{r.visit\_id,r.patient\_id,d.*}}(\text{doctor}) \bowtie_{\text{d.doctor\_id=r.doctor\_id}} \Pi_{*} \rho(r, \sigma_{\text{operation\_id not null}}(\text{visit}))))$

SQL:

```
SELECT distinct p.patient_id,p.patient_name,p.gender,p.date_of_birth,p.blood_group FROM
patient as p JOIN
(SELECT * FROM department as dp JOIN
(SELECT r.visit_id,r.patient_id,d.* FROM doctor as d JOIN
(SELECT * FROM visit where operation_id is not null) as r
ON (d.doctor_id=r.doctor_id) ) as x
ON (dp.head_of_department=x.doctor_id)) as y
ON (p.patient_id=y.patient_id);
```

OUTPUT:



24) List the Nurses who have taken leave on a particular date.

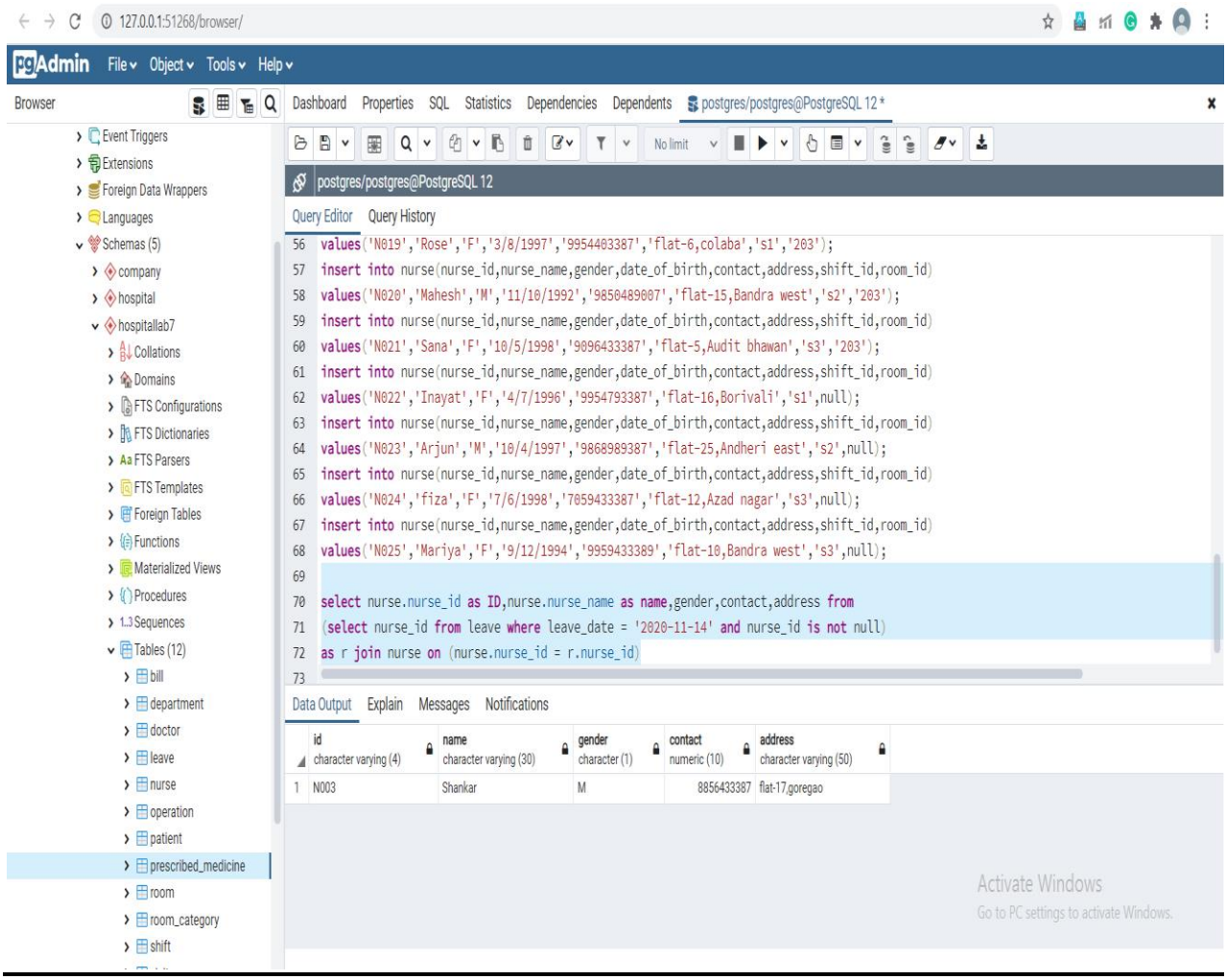
Relational Algebra:

$\Pi_{\text{nurse\_id -> ID,nurse\_name->name,gender,contact,address}}\ \rho(\Pi_{\text{nurse\_id}}\ (\sigma_{\text{leave\_date}} = \text{'2020-11-14' and nurse\_id is not null Leave}), r) \bowtie_{\text{nurse.nurse\_id = r.nurse\_id}} \text{Nurse}$

SQL:

```
select nurse.nurse_id as ID,nurse.nurse_name as name,gender,contact,address from (select nurse_id from leave where leave_date = '2020-11-14' and nurse_id is not null) as r join nurse on (nurse.nurse_id = r.nurse_id)
```

OUTPUT:



25) Get the revenue generated in a particular month by rooms (operation).

Relational Algebra:

$\Pi_{.*}$ (room_category) $\bowtie$ room_category.category_id=x.category_id

ρ($\Pi_{.*}$ (room) $\bowtie$ room.room_id=r.room_id

ρ(Πvisit_id,room_id,(discharge_date-visit_date -> days) (σ visit_date >= '2020-10-01' and visit_date <= '2020-10-31' and discharge_date <='2020-10-31' visit)

U

Πvisit_id,room_id,(discharge_date-visit_date -> days) (σ visit_date<'2020-10-01' and discharge_date <= '2020-10-31' and discharge_date >= '2020-10-01' and discharge_date is not null

visit)

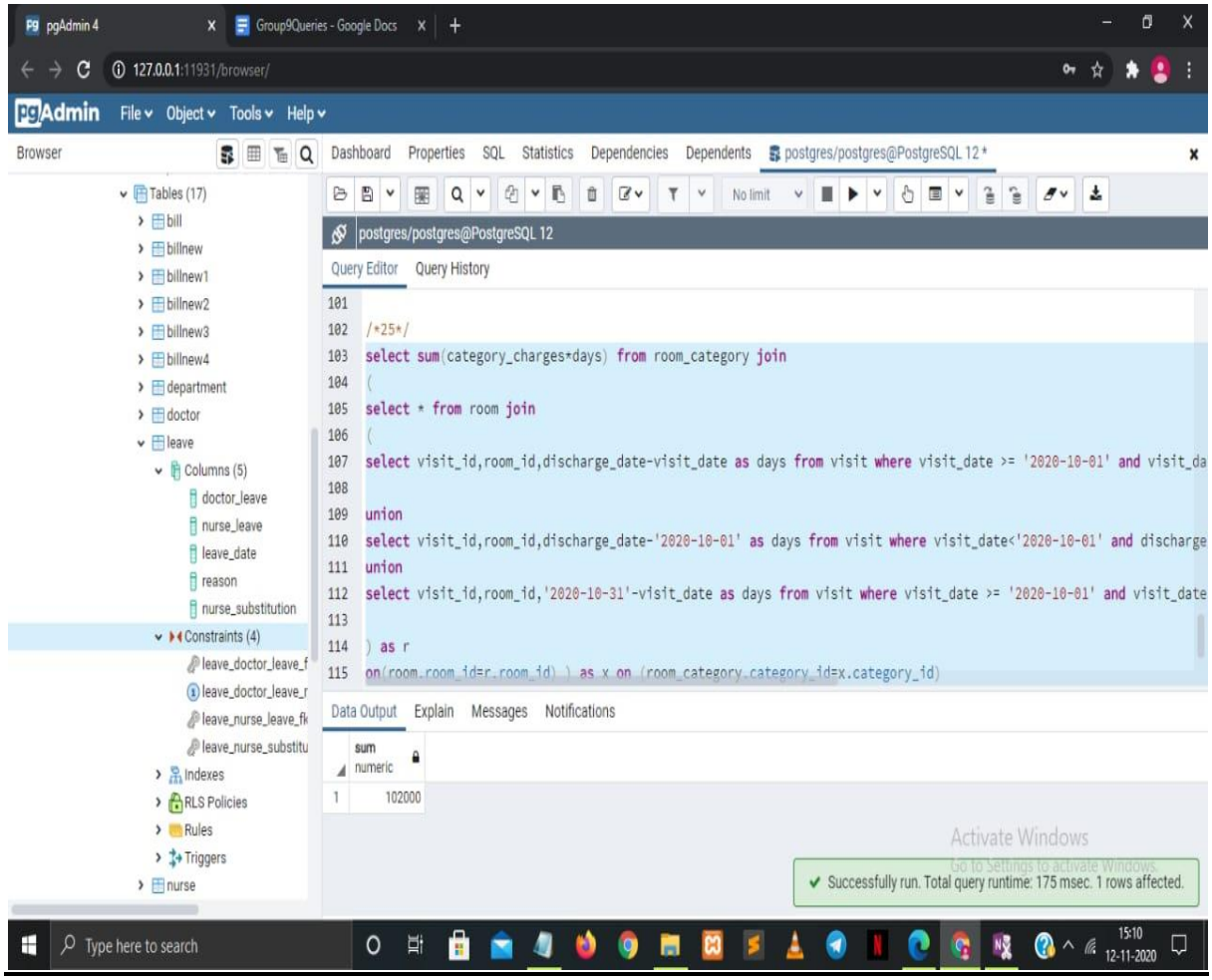
U

Πvisit_id,room_id,(discharge_date-visit_date -> days) (σ visit_date >= '2020-10-01' and visit_date <= '2020-10-31' and discharge_date >'2020-10-31'visit)

,r),x)

SQL:

```
select sum(category_charges*days) from room_category join

(select * from room join

(select visit_id,room_id,discharge_date-visit_date as days from visit where visit_date >= '2020-10-01' and visit_date  <= '2020-10-31' and discharge_date <='2020-10-31'

union

select visit_id,room_id,discharge_date-'2020-10-01' as days from visit where visit_date<'2020-10-01' and discharge_date  <= '2020-10-31' and discharge_date  >= '2020-10-01' and discharge_date is not nullunion

select visit_id,room_id,'2020-10-31'-visit_date as days from visit where visit_date >= '2020-10-01' and visit_date  <= '2020-10-31' and discharge_date >'2020-10-31') as r

on(room.room_id=r.room_id) ) as x on (room_category.category_id=x.category_id)
```


pgAdmin 4
Group9Queries - Google Docs
127.0.0.1:11931/browser/
pgAdmin
File
Object
Tools
Help
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
postgres/postgres@PostgreSQL 12 *
Tables (17)
bill
billnew
billnew1
billnew2
billnew3
billnew4
department
doctor
leave
Columns (5)
doctor_leave
nurse_leave
leave_date
reason
nurse_substitution
Constraints (4)
leave_doctor_leave_f
leave_doctor_leave_r
leave_nurse_leave_fk
leave_nurse_substitu
Indexes
RLS Policies
Rules
Triggers
nurse
No limit
postgres/postgres@PostgreSQL 12
Query Editor
Query History
101
102 /*25*/
103 select sum(category_charges*days) from room_category join
104 (
105 select * from room join
106 (
107 select visit_id,room_id,discharge_date-visit_date as days from visit where visit_date >= '2020-10-01' and visit_da
108
109 union
110 select visit_id,room_id,discharge_date-'2020-10-01' as days from visit where visit_date<'2020-10-01' and discharge
111 union
112 select visit_id,room_id,'2020-10-31'-visit_date as days from visit where visit_date >= '2020-10-01' and visit_date
113
114) as r
115 on(room.room_id=r.room_id)) as x on (room_category.category_id=x.category_id)
Data Output
Explain
Messages
Notifications
sum
numeric
1
102000
Activate Windows
Go to Settings to activate Windows.
Successfully run. Total query runtime: 175 msec. 1 rows affected.
Type here to search
15:10
12-11-2020

26) List the substitution of nurses on a particular date.

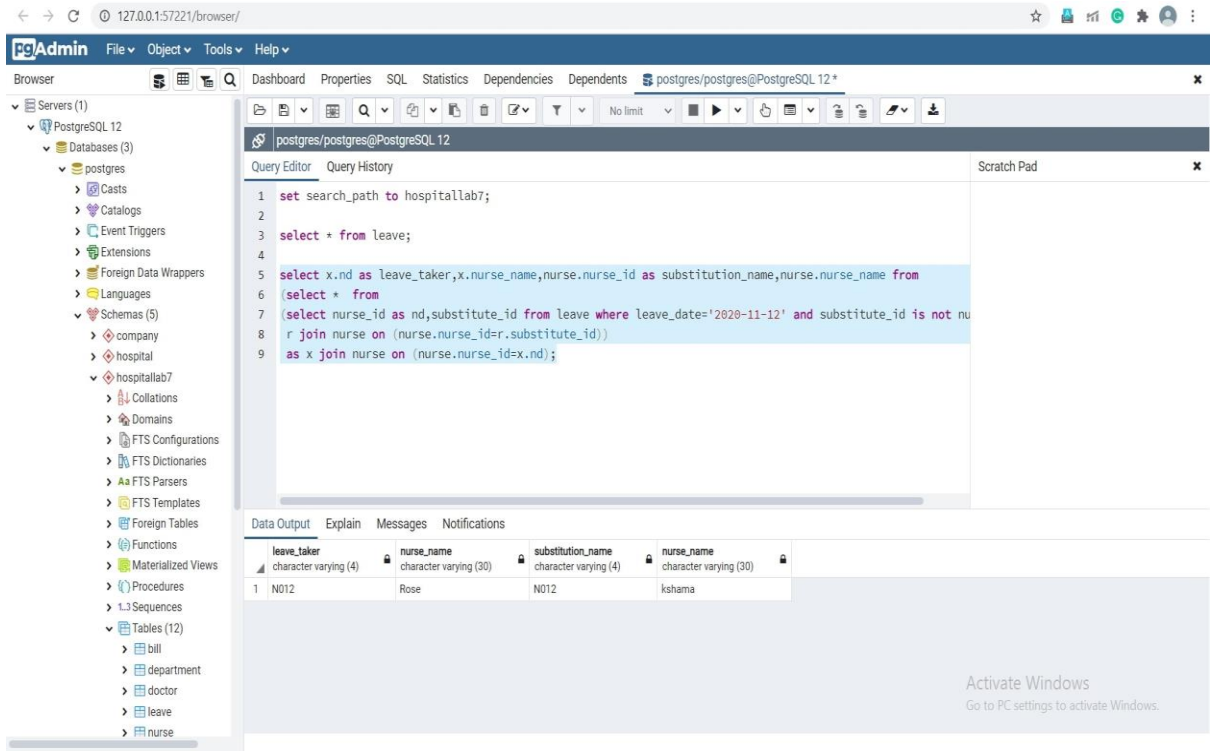
Relational Algebra:

$\Pi_{\text{nurse.nurse\_id -> Leave\_Taker,nurse.nurse\_name,x.sid-> Substitute,name,x.nurse\_name}}(\boxtimes(\pi_{.*}(\boxtimes(\pi_{\text{nurse\_id->nid , substitute\_id->sid}}(\sigma_{\text{leave\_date = '2020-11-14' and substitute\_id is not null}}\text{Leave}),R)\bowtie_{\text{nurse.nurse\_id=r.sid}}\text{Nurse}),X)\bowtie_{\text{nurse.nurse\_id=x.nid}}\text{Nurse})$

SQL:

```
select nurse.nurse_id as Leave_Taker,nurse.nurse_name, x.sid as Substitute,x.nurse_name
from(

select * from

(

select nurse_id as nid,substitute_id as sid from leave where leave_date='2020-11-12' and
substitute_id is not null) as

 r join nurse on (nurse.nurse_id=r.sid) ) as x join nurse on (nurse.nurse_id=x.nid)
```

OUTPUT:



27) Count the number of leaves taken by particular Nurse in a given time.

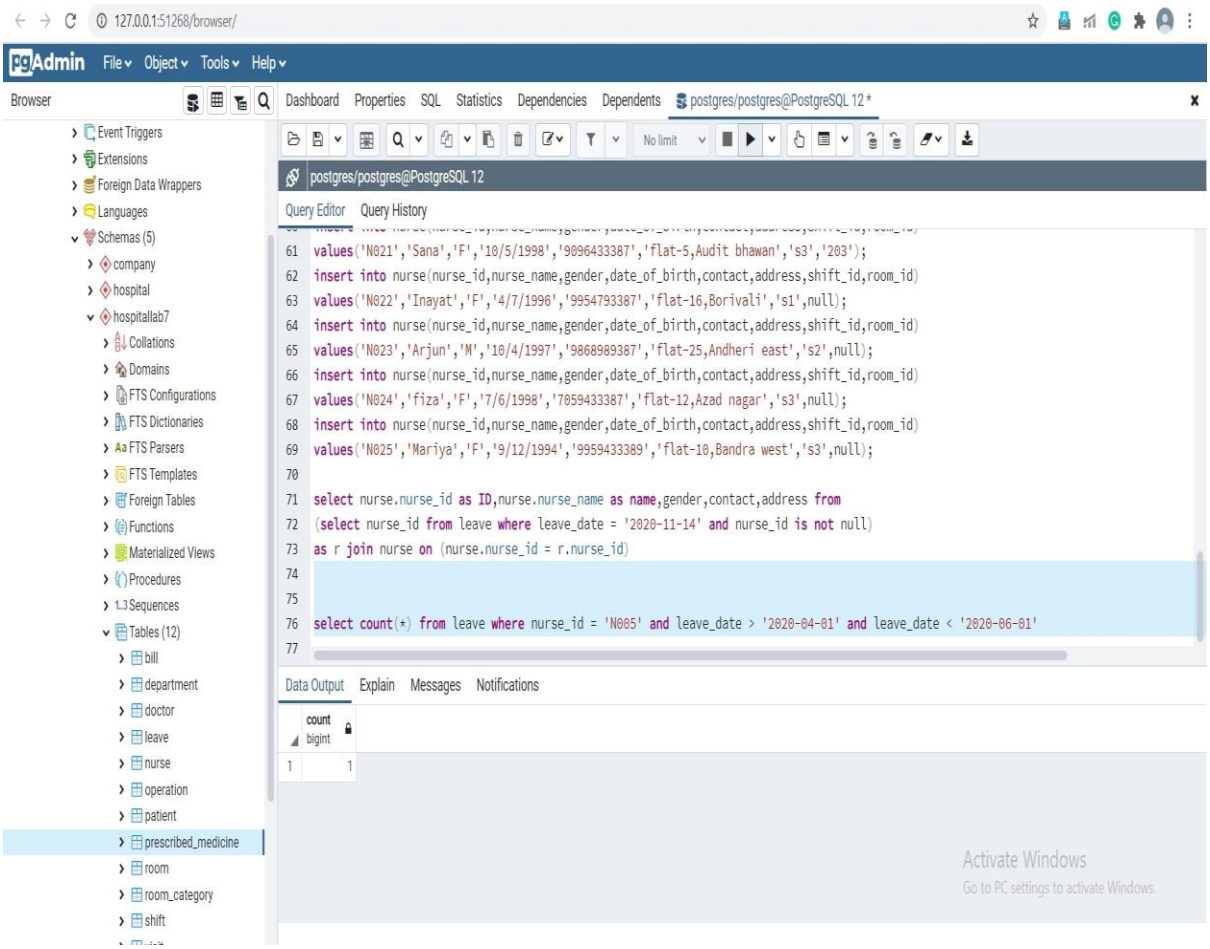
Relational Algebra:

$\mathcal{F}_{count*}(\sigma_{nurse\_id}$ = 'N005' and leave_date > '2020-04-01' and leave_date < '2020-06-01' Leave)

SQL:

select count(*) from leave where nurse_id = 'N005' and leave_date > '2020-04-01' and leave_date < '2020-06-01'

OUTPUT:



28) Count the number of leaves taken by each nurse in a year

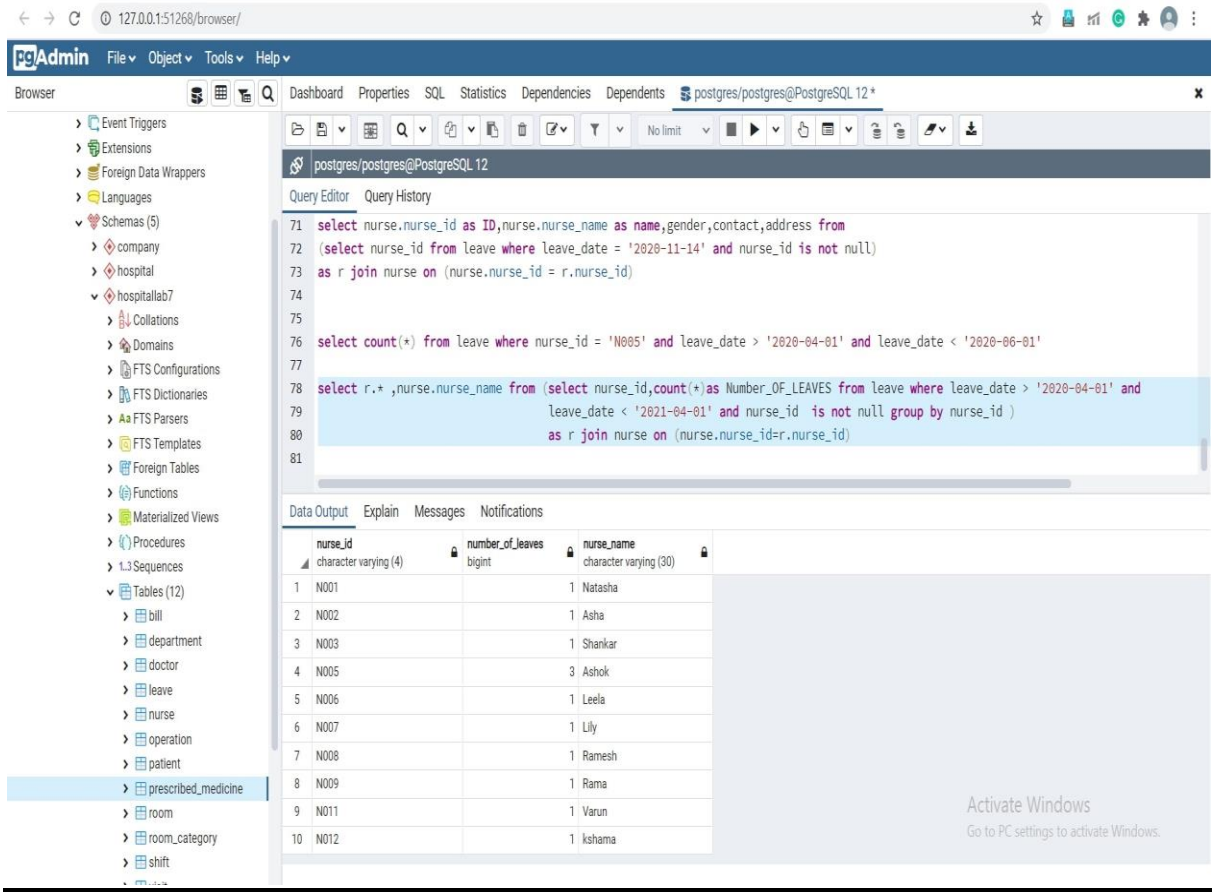
Relational Algebra:

$\Pi_{r.*,nurse.nurse\_name}\ \rho(_{nurse\_id}\mathcal{F}_{count^* \rightarrow number\_of\_leaves,nurse\_id}\ (\sigma_{leave\_date} > '2020\text{-}11\text{-}01'$ and leave_date < '2020-12-01' and nurse_id is not null Leave),r) $\bowtie_{nurse.nurse\_id=r.nurse\_id}$ Nurse

SQL:

select r.* ,nurse.nurse_name from (select nurse_id,count(*)as Number_OF_LEAVES from leave where leave_date > '2020-04-01' and leave_date < '2021-04-01' and nurse_id is not null group by nurse_id) as r join nurse on (nurse.nurse_id=r.nurse_id)

OUTPUT:



29) Select the details of doctors (department wise) whose consultation charge is greater than the average consultation charge.

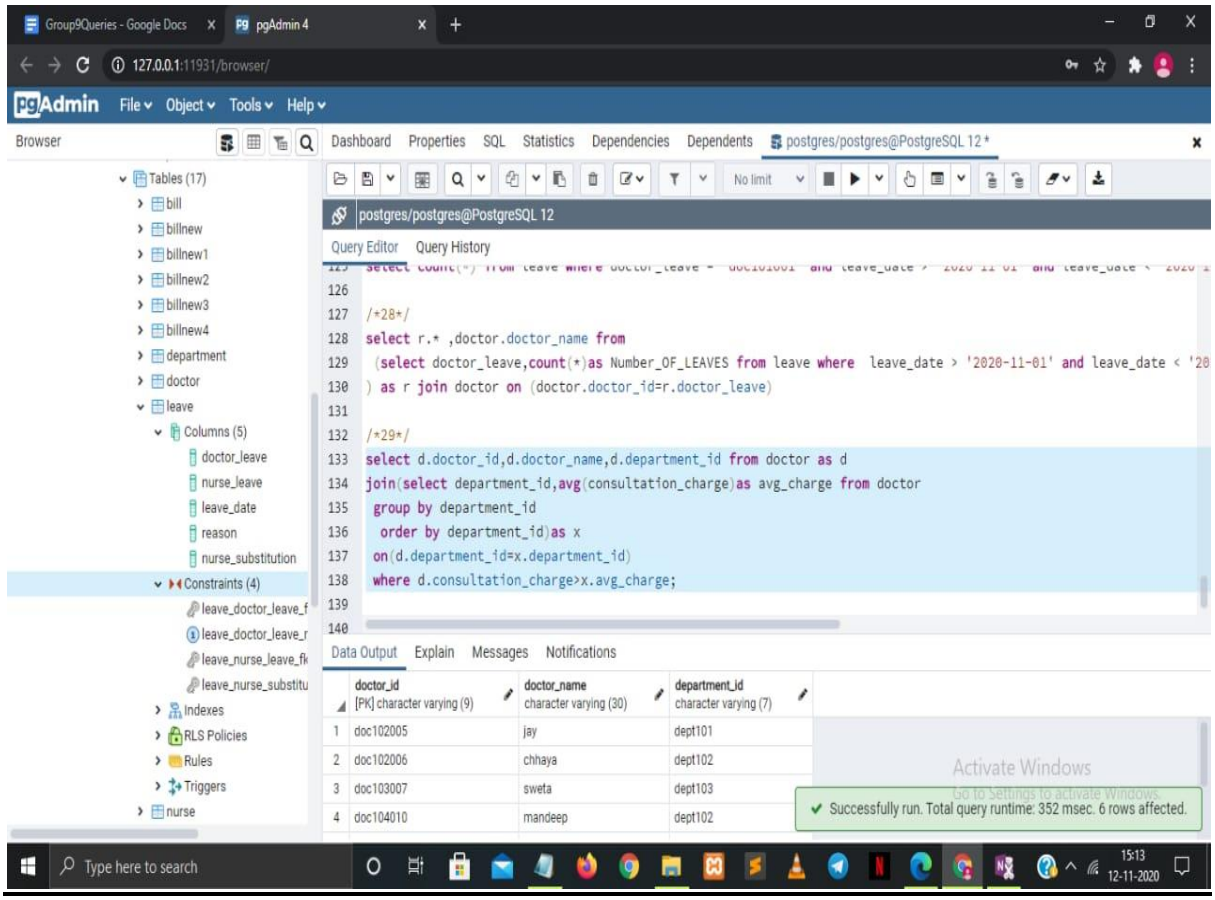
Relational Algebra:

$\sigma_{d.consultation\_charge>x.avg\_charge}(doctor \bowtie_{<d.department\_id=x.department\_id>}(\rho(_{department\_id}f_{avg(consultation\_charge)\rightarrow avg\_charge}(doctor),x)))$

SQL:

```
select d.doctor_id,d.doctor_name,d.department_id from doctor as d
join(select department_id,avg(consultation_charge)as avg_charge from doctor
 group by department_id
  order by department_id)as x
 on(d.department_id=x.department_id)
 where d.consultation_charge>x.avg_charge;
```

OUTPUT:



30) Number of new patient registered with hospital in a year

Relational Algebra:

$f_{count(*)\rightarrow new\_patients}(\rho(r, _{patient\_id}f_{min(visit\_date)\rightarrow d}(\sigma_{r.d > '2019\text{-}04\text{-}01' \text{ and } r.d < '2020\text{-}03\text{-}31'} visit)))$

SQL:

```
select count(*) as New_Patients from

(select min(visit_date) as d,patient_id from visit group by patient_id ) as r where r.d > '2019-04-01' and r.d <'2020-03-31'
```

OUTPUT: