**Diabetes Data SQL Analysis**

1. Display any 10 random DM patients.

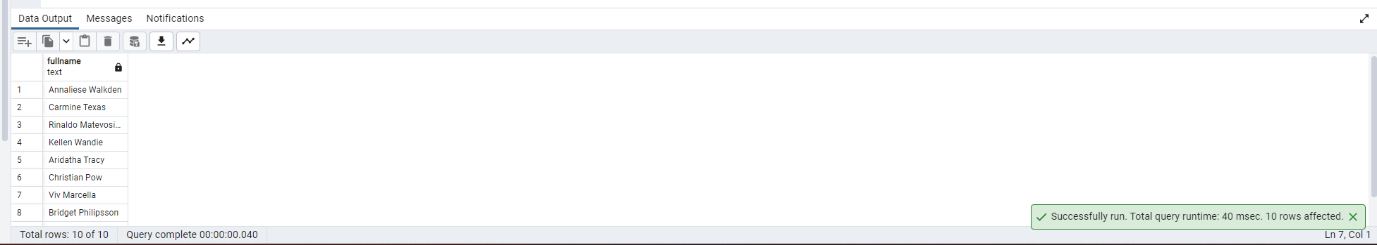
Query:

select "Firstname"||' '||"Lastname" as fullname

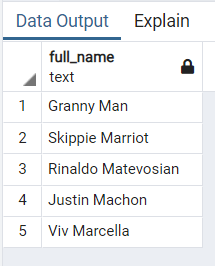
from "Patients"

order by random()

limit 10;



1. Please go through the below screenshot and create the exact output.

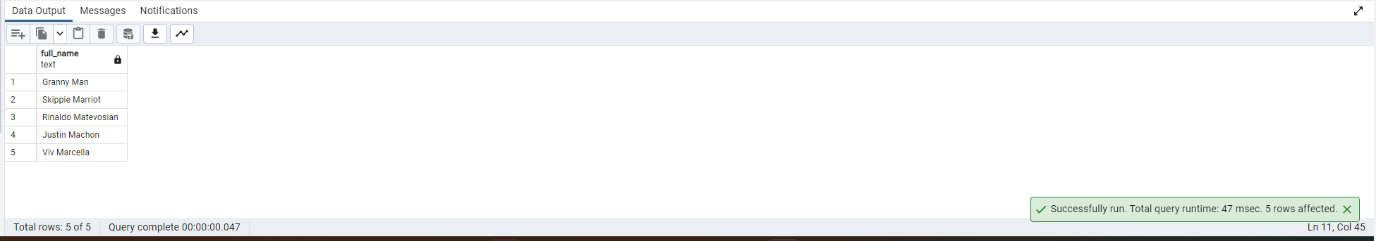


Query:

select "Firstname"||' '||"Lastname" as full\_name

from "Patients"

where "Lastname" LIKE 'Ma%';

****

1. Write a query to get a list of patients whose RPE start is at moderate intensity.

Query:

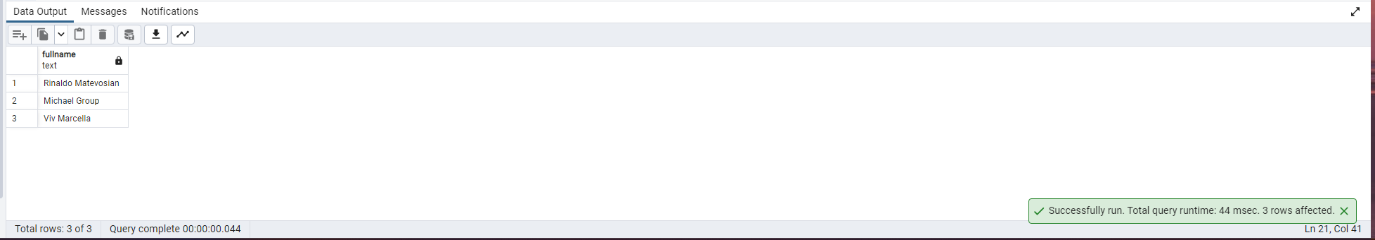
select "Firstname"||' '||"Lastname" as fullname

from public."Patients" as p

join public."Walking\_Test" as w

on p."WalkTest\_ID" = w."WalkTest\_ID"

where "Gait\_RPE\_Start " between 4 and 6;



1. Write a query by using common table expressions and case statements to display birthyear ranges.

Query:

WITH birthyear\_range as (

select "Patient\_ID", (DATE\_PART('year',"Visit\_Date")-"Age") as Birthyear,

(CASE

WHEN (DATE\_PART('year',"Visit\_Date")-"Age") <= 1930 THEN '1920-1930'

WHEN (DATE\_PART('year',"Visit\_Date")-"Age") <= 1940 THEN '1931-1940'

WHEN (DATE\_PART('year',"Visit\_Date")-"Age") <= 1950 THEN '1941-1950'

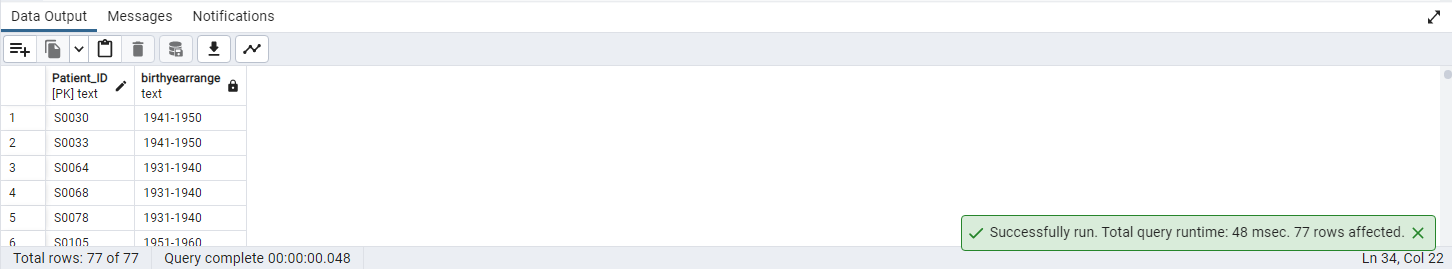
ELSE '1951-1960'

END) BirthyearRange

from public."Patients")

select "Patient\_ID", BirthyearRange

from birthyear\_range;



1. Display DM patient names with highest day MAP and night MAP (without using limit).

Query:

WITH MAP\_table AS (

SELECT p."Firstname"||' '||p."Lastname" as Patientname,

(b."24Hr\_Day\_DBP" + (b."24Hr\_Day\_SBP" - b."24Hr\_Day\_DBP")/3) AS dayMAP,

(b."24Hr\_Day\_DBP" + (b."24Hr\_Day\_SBP" - b."24Hr\_Day\_DBP")/3) AS nightMAP

FROM public."Patients" AS p

JOIN public."Blood\_Pressure" AS b

ON b."BP\_ID" = p."BP\_ID"

)

SELECT Patientname

FROM MAP\_table

WHERE dayMAP = (

SELECT MAX(dayMAP)

FROM MAP\_table

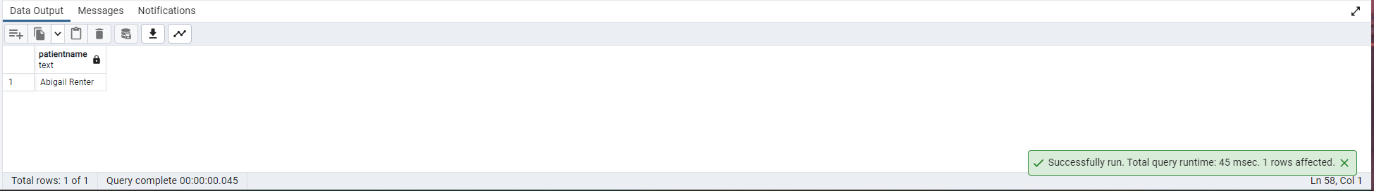
) OR

nightMAP = (

SELECT MAX(nightMAP)

FROM MAP\_table

);



1. Create view on table Lab Test by selecting some columns and filter data using Where condition.

Query:

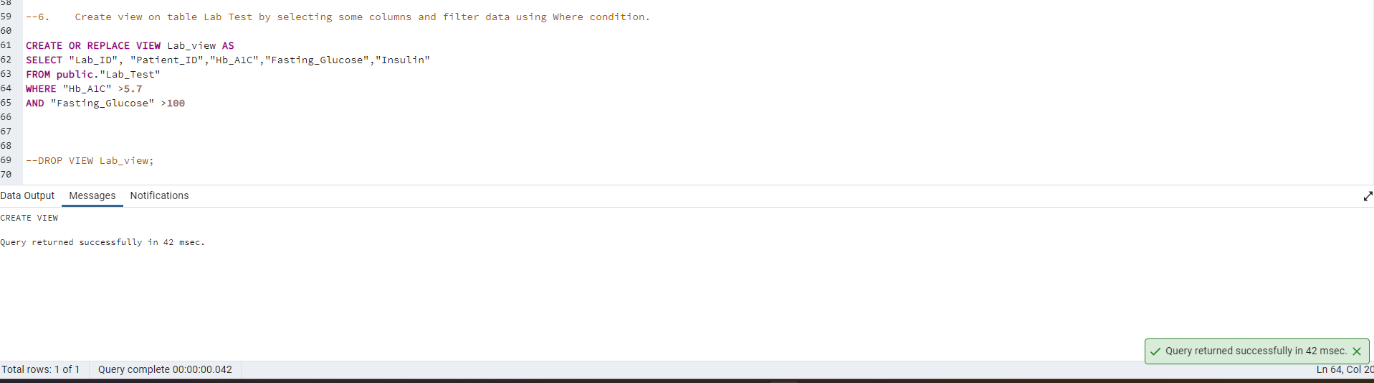
CREATE OR REPLACE VIEW Lab\_view AS

SELECT "Lab\_ID", "Patient\_ID","Hb\_A1C","Fasting\_Glucose","Insulin"

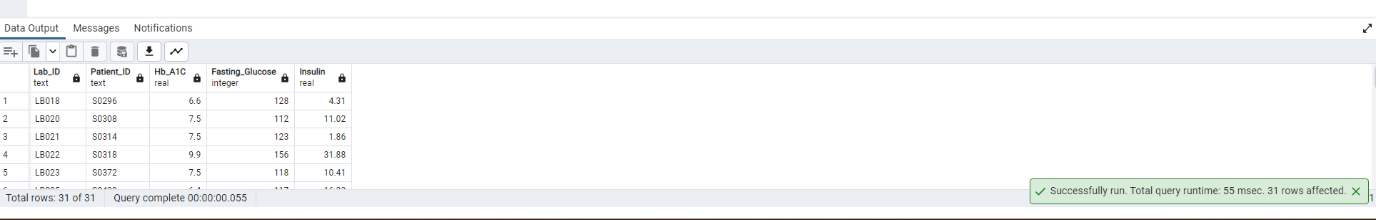
FROM public."Lab\_Test"

WHERE "Hb\_A1C" >5.7

AND "Fasting\_Glucose" >100



SELECT \* FROM public.lab\_view



DROP VIEW Lab\_view;

1. Display a list of Patient IDs and their Group whose diabetes duration is greater than 10 years.

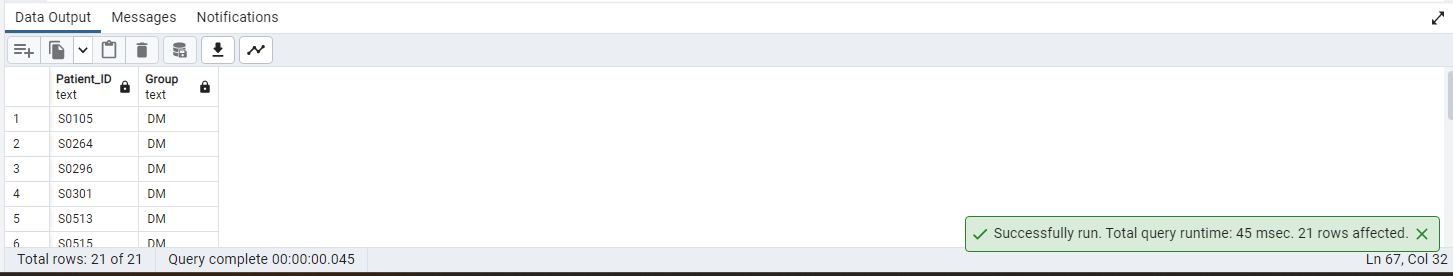
Query:

SELECT "Patient\_ID","Group"

FROM public."Patients",public."Group"

WHERE public."Patients"."Group\_ID"=public."Group"."Group\_ID"

AND "Diabetes\_Duration" >10;



1. Write a query to list male patient ids and their names who are above 40 years of age and less than 60 years and have Day BloodPressureSystolic above 120 and Day BloodPressureDiastolic above 80.

Query:

SELECT p."Patient\_ID", p."Firstname"||' '||p."Lastname" as Fullname

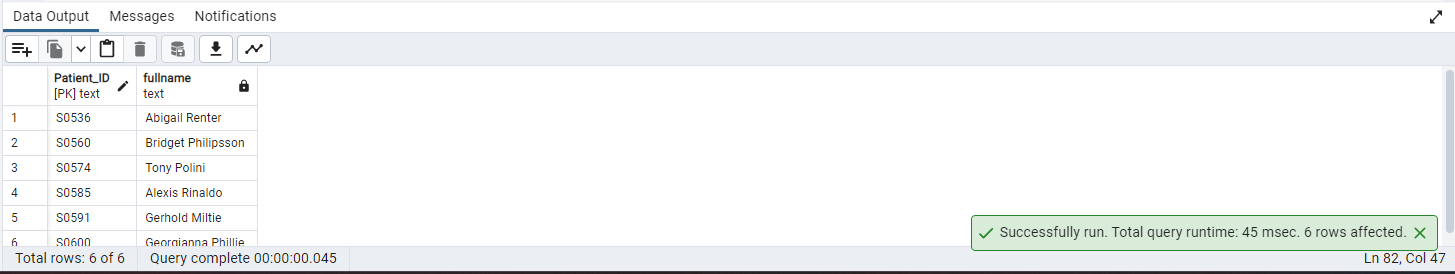
FROM public."Patients" as p,public."Blood\_Pressure" as b

WHERE p."BP\_ID" = b."BP\_ID"

AND p."Age" BETWEEN 40 AND 60

AND b."24Hr\_Day\_SBP" > 120

AND b."24Hr\_Day\_DBP" > 80



1. Use a function to calculate the percentage of patients according to the lab visited per month.

Query:

CREATE OR REPLACE FUNCTION lab\_pct(pt\_count real) returns numeric AS

$$ select pt\_count \* 100/77;

$$ Language SQL;

SELECT "Lab\_names",

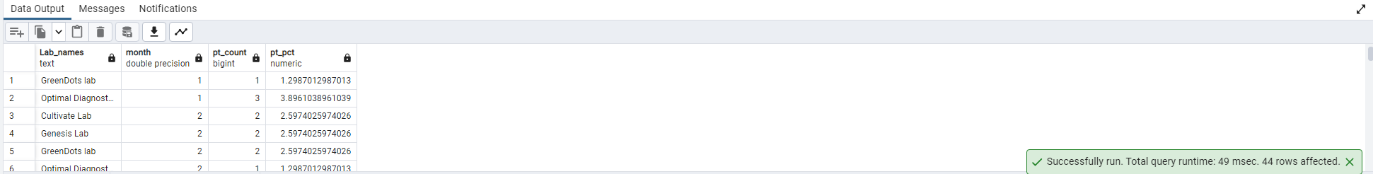
DATE\_PART('month',"Lab\_Visit\_Date") AS MONTH,

COUNT(\*) AS pt\_count,

lab\_pct(COUNT(\*)) AS pt\_pct

FROM public."Lab\_Visit"

GROUP BY 1,2

ORDER BY 2;

1. Count of patients by first letter of firstname.

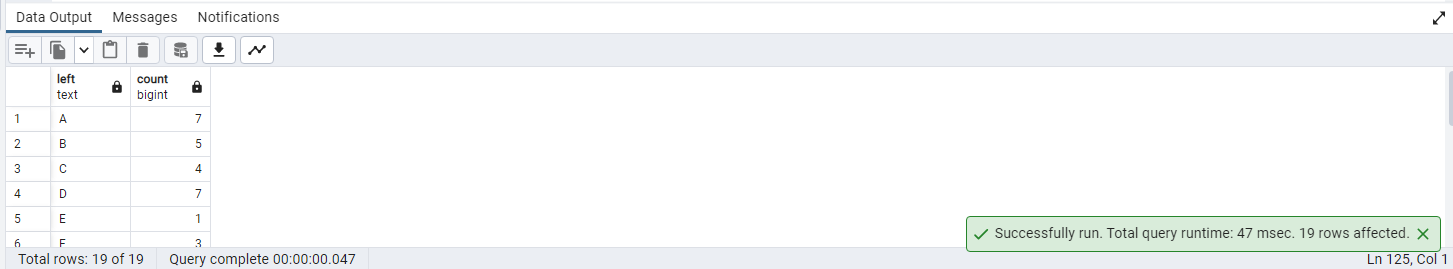
Query:

SELECT LEFT("Firstname",1),COUNT(\*)

FROM public."Patients"

GROUP BY 1

ORDER BY 1;



1. write a query to get the list of patients whose lipid test value is null.

Query:

SELECT p."Firstname"||' '||p."Lastname" AS patientname

FROM public."Patients" p

LEFT JOIN public."Link\_Reference" l

ON p."Link\_Reference\_ID"= l."Link\_Reference\_ID"

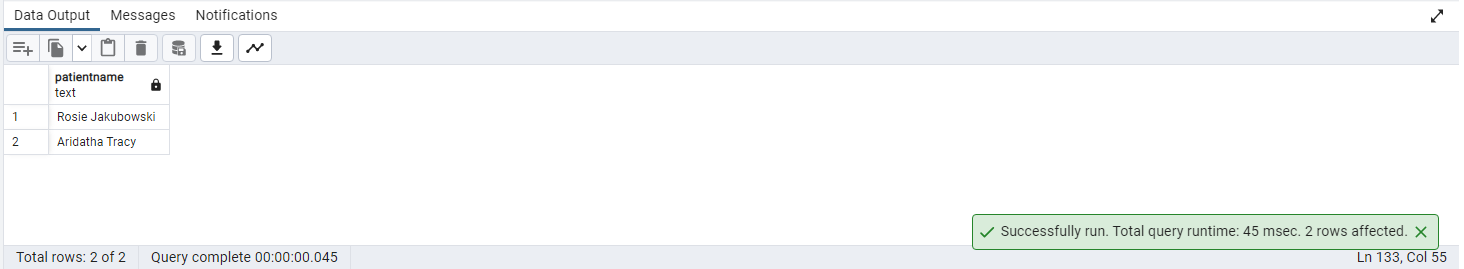
LEFT JOIN public."Lipid\_Lab\_Test" ll

ON l."Lipid\_ID" = ll."Lipid\_ID"

WHERE "Fasting\_Cholestrol" IS NULL

OR "Fasting\_Triglyc" IS NULL

OR "Fasting\_HDL" IS NULL

OR "Fasting\_LDL" IS NULL

1. Create a stored procedure to make user ids for the given patient id.

Query:

CREATE OR REPLACE PROCEDURE patient\_userid(patientid text, INOUT val refcursor)

LANGUAGE plpgsql

AS $BODY$

BEGIN

OPEN val FOR

SELECT LOWER(LEFT("Firstname",1)||"Lastname") AS username

FROM public."Patients"

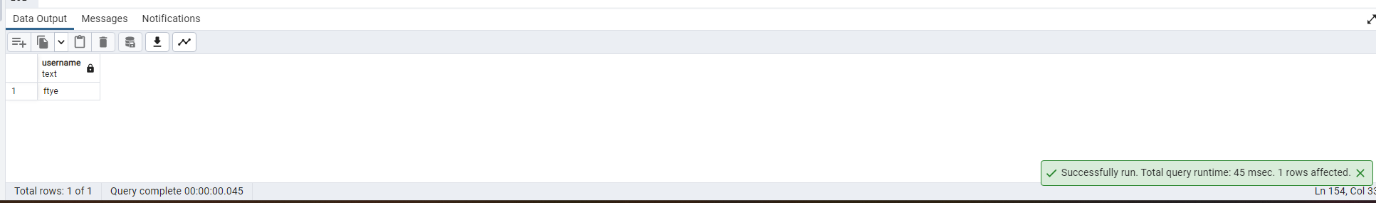
WHERE "Patient\_ID" = patientid;

END

$BODY$;

CALL patient\_userid('S0531','ref');

FETCH ALL IN "ref";



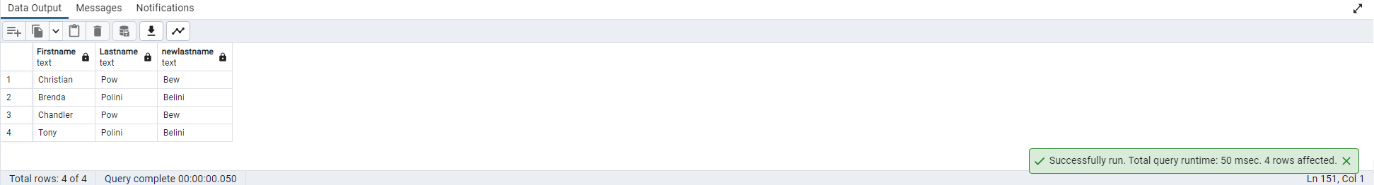
1. Display Patients With LastName contains 'Po' and replace it with 'Be'.

Query:

SELECT "Firstname","Lastname",REGEXP\_REPLACE("Lastname",'Po','Be') AS newLastname

FROM public."Patients"

WHERE "Lastname" LIKE '%Po%'



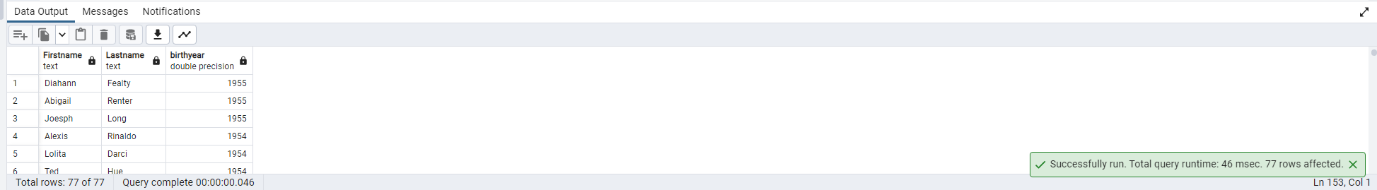
1. Calculate the patient's birth year in descending order.

Query:

SELECT "Firstname","Lastname",(DATE\_PART('year',"Visit\_Date")-"Age") AS Birthyear

FROM public."Patients"

ORDER BY 3 DESC;



1. Find the patients that have eye damage due to diabetes

Query:

SELECT

p."Firstname",

p."Lastname",

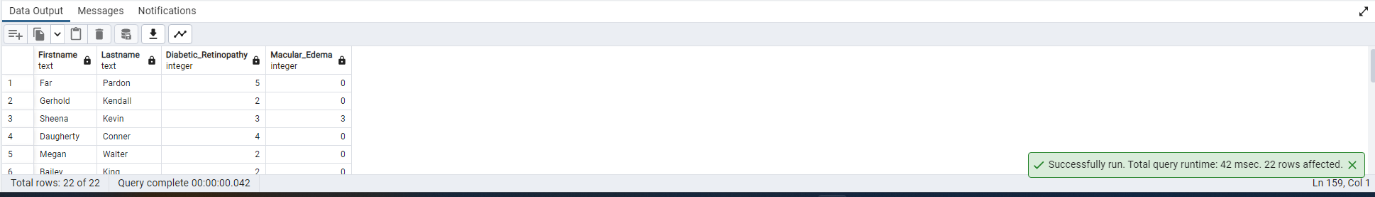
o."Diabetic\_Retinopathy",

o."Macular\_Edema"

FROM public."Patients" AS p,public."Opthalmology" AS o

WHERE p."Opthal\_ID" = o."Opthal\_ID"

AND (o. "Diabetic\_Retinopathy">0 OR o."Macular\_Edema" >0);



1. Query to classify Gait RPE End into 5 categories as per the intensity. (Hint: Use of CASE statement)

Query:

SELECT

p."Firstname",

p."Lastname",

w."Gait\_RPE\_End ",

(CASE

WHEN w."Gait\_RPE\_End " = 0 THEN 'Rest'

WHEN w."Gait\_RPE\_End " BETWEEN 1 AND 3 THEN 'Easy Intensity'

WHEN w."Gait\_RPE\_End " BETWEEN 4 AND 6 THEN 'Moderate Intensity'

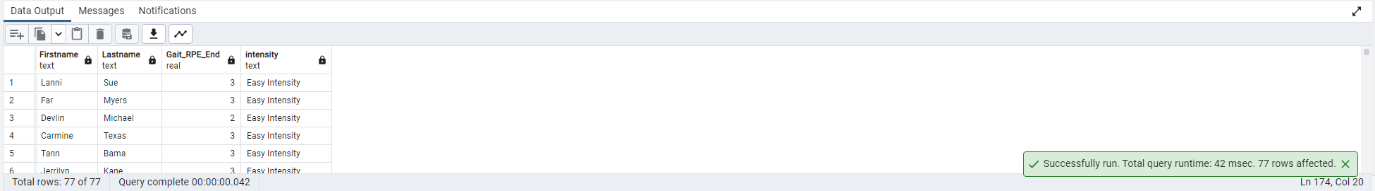
WHEN w."Gait\_RPE\_End " BETWEEN 7 AND 9 THEN 'Hard Intensity'

ELSE 'Max Effort intensity'

END) AS Intensity

FROM public."Patients" AS p,public."Walking\_Test" AS w

WHERE p."WalkTest\_ID" = w."WalkTest\_ID";

****

17. Create view on patient table with check constraint condition.

Query:

create or replace view patients\_view

as select "Patient\_ID","Age" from "Patients" with check option;

update patients\_view set "Patient\_ID" =1;

A picture containing text, screenshot

Description automatically generated

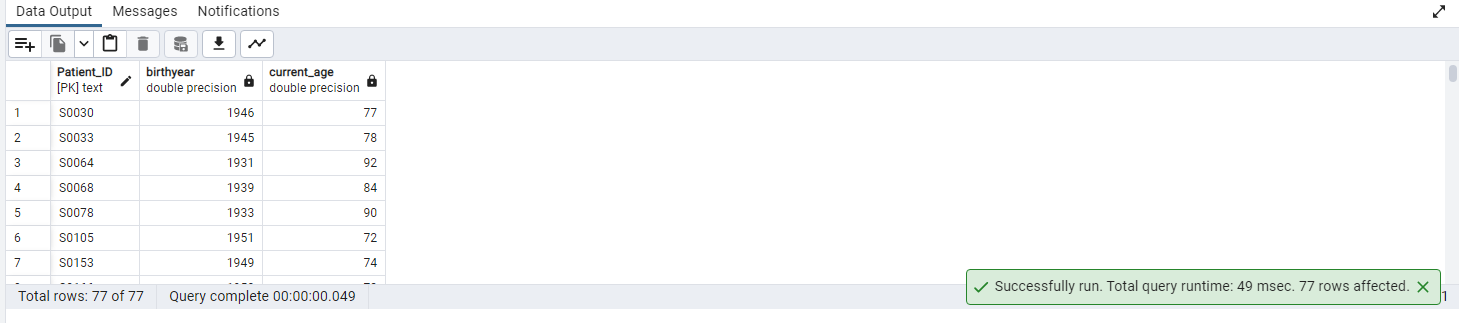
1. Calculate the patient's current age.

Query:

select "Patient\_ID", (DATE\_PART('year',"Visit\_Date")-"Age") as Birthyear,

(date\_part('year',current\_date)-(DATE\_PART('year',"Visit\_Date")-"Age")) as current\_Age

from public."Patients";



1. Write a query to display Mr. or Ms. as prefix to patients’ names with respect to gender.

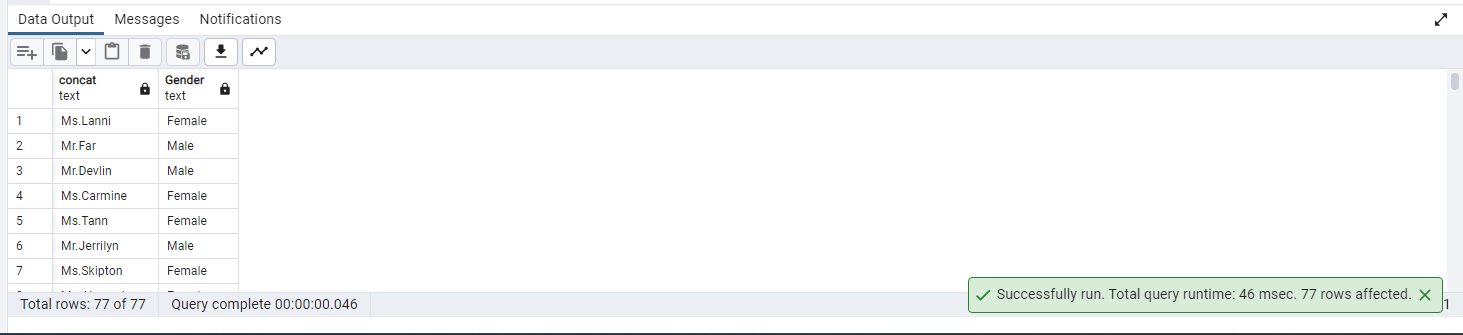
Query:

select concat(case

when "Gender"."Gender"='Male' then 'Mr.'

when "Gender"."Gender"='Female' then 'Ms.' end , "Patients"."Firstname" ),"Gender"."Gender"

from "Patients","Gender" where "Patients"."Gender\_ID"="Gender"."Gender\_ID";



1. Write a query to get DM patient names whose distance is greater than 400 and speed is greater than 1.

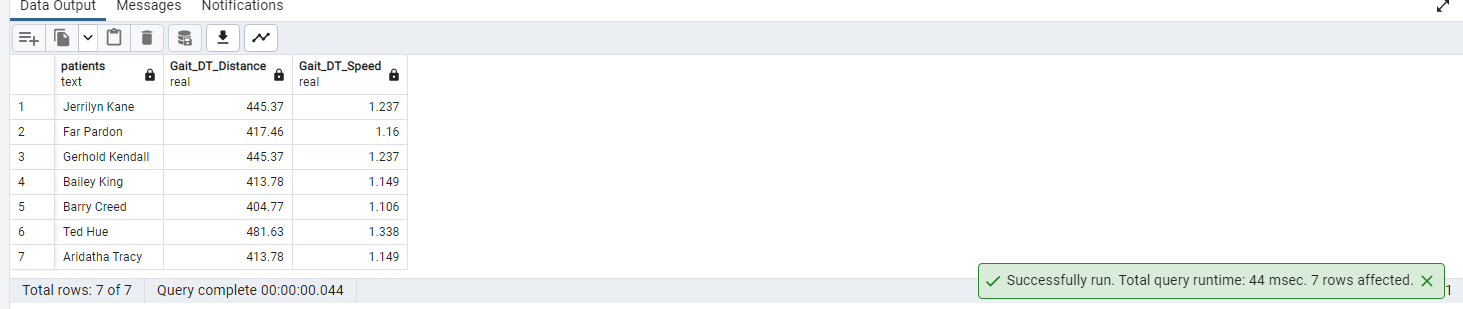
Query:

select concat(pat."Firstname",' ',pat."Lastname") Patients,w."Gait\_DT\_Distance",w."Gait\_DT\_Speed"

from "Patients" pat join "Walking\_Test" w on pat."WalkTest\_ID"=w."WalkTest\_ID"

join "Group" grp on pat."Group\_ID"=grp."Group\_ID"

where grp."Group"='DM' and w."Gait\_DT\_Distance">400 and w."Gait\_DT\_Speed">1;



21.Create a trigger to raise notice and prevent the deletion of a record from a view.

Query:

create view race\_view as

select \* from "Race";

---------------------------------------------------

create or replace function delete\_race\_view()returns trigger as $DonotDelete$

begin

if (TG\_OP = 'DELETE') then

Raise exception 'STOP AND THINK ABOUT WHAT YOURE DOING';

end if;

return null;

end;

$DonotDelete$ language plpgsql;

----trigger---

create trigger DonotDelete

instead of delete on public.race\_view

for each row

execute function delete\_race\_view();

-------------------

delete from public.race\_view;

A screen shot of a computer

Description automatically generated with medium confidence

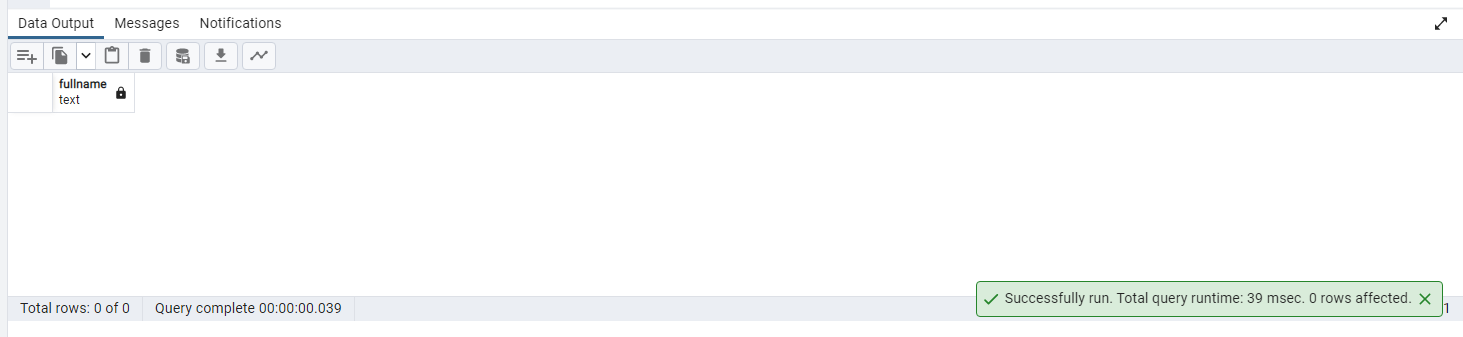
22. Select the patient's full name with a name starting with 's'

followed by any character, followed by 'r', followed by any character, followed by b.

Query:

select concat("Firstname","Lastname") as "fullname"

from "Patients" where concat("Firstname","Lastname") like 'S\_r\_b%';



23. Write a query to get which race has the maximum number of Diabetic patients.

Query:

--(Patients having Fasting\_Glucose >=120 are diabetic)

select ra."Race" ,count(pat."Patient\_ID") Diabetic

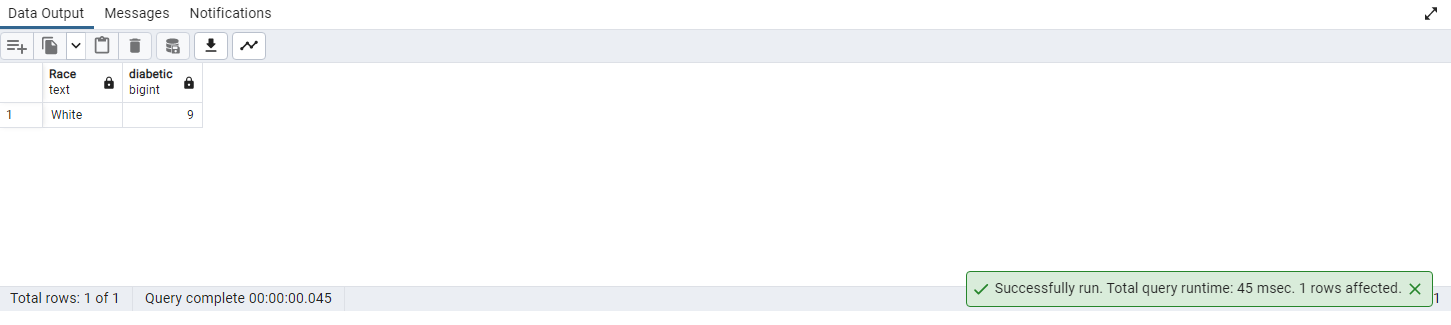
from "Patients" pat join "Race" ra on ra."Race\_ID"=pat."Race\_ID"

join "Link\_Reference" lr on pat."Link\_Reference\_ID"=lr."Link\_Reference\_ID"

join "Lab\_Test" lt on lt."Lab\_ID"=lr."Lab\_ID"

where lt."Fasting\_Glucose">=120

group by ra."Race" order by Diabetic desc limit 1;



24. Find the list of Patients who has leukopenia.

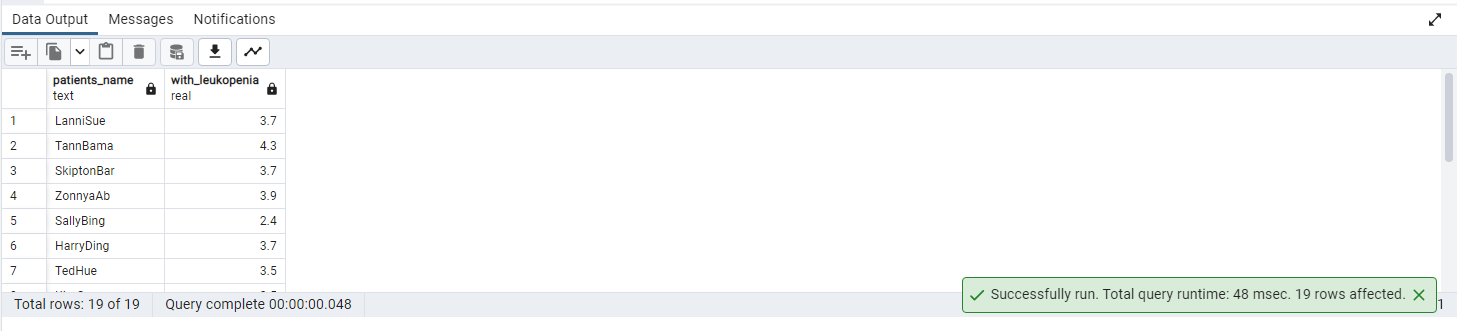
Query:

select concat(pat."Firstname",pat."Lastname") as Patients\_Name,lt."WBC" as with\_leukopenia from "Patients" pat

join "Link\_Reference" lr on pat."Link\_Reference\_ID"=lr."Link\_Reference\_ID"

join "Lab\_Test" lt on lt."Lab\_ID"=lr."Lab\_ID"

where lt."WBC" in (select "WBC" from "Lab\_Test" where "WBC" not between '4.5' and '11');



25. Get the number of patients in the year 2005 in each of the Genesis and Cultivate labs.

Query:

select lv."Lab\_names",count(pat."Patient\_ID")as no\_of\_Patients,

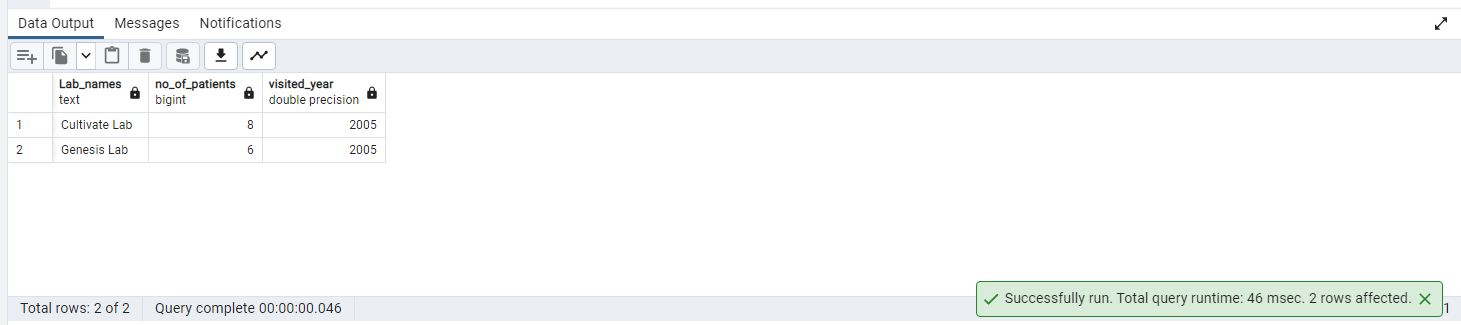
date\_part('year',pat."Visit\_Date") as visited\_year from "Patients" pat

join "Link\_Reference" lr on pat."Link\_Reference\_ID"=lr."Link\_Reference\_ID"

join "Lab\_Visit" lv on lv."Lab\_visit\_ID"=lr."Lab\_visit\_ID"

where lv."Lab\_names" in ('Genesis Lab','Cultivate Lab') and date\_part('year',pat."Visit\_Date")=2005

group by date\_part('year',pat."Visit\_Date"),lv."Lab\_names";



26. Write a query to get a list of patient IDs' and their Fasting Cholesterol in February 2006

Query:

select "Lipid\_Lab\_Test"."Patient\_ID","Lipid\_Lab\_Test"."Fasting\_Cholestrol",

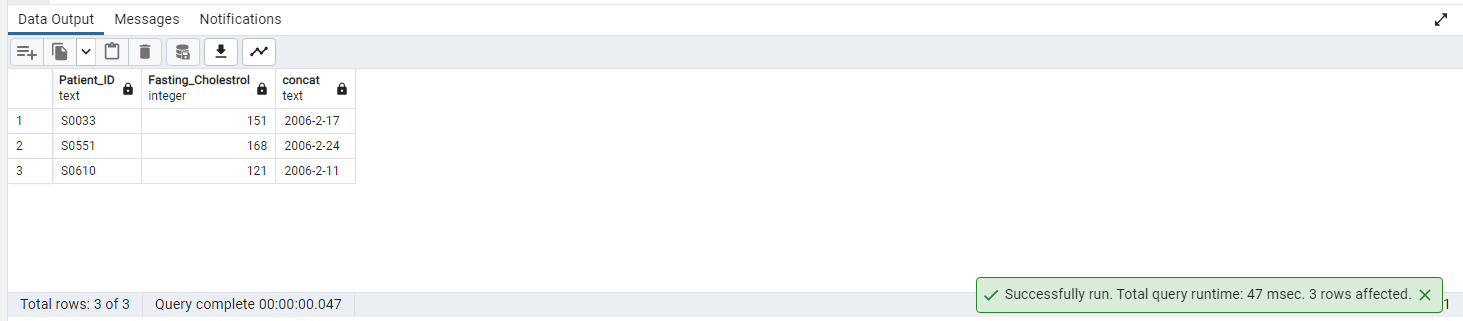
concat(date\_part('year',"Visit\_Date"),'-',date\_part('month',"Visit\_Date"),'-',date\_part('day',"Visit\_Date"))

from public."Lipid\_Lab\_Test"

join "Link\_Reference" on "Lipid\_Lab\_Test"."Lipid\_ID"=public."Link\_Reference"."Lipid\_ID"

join "Patients" on "Link\_Reference"."Link\_Reference\_ID"="Patients"."Link\_Reference\_ID"

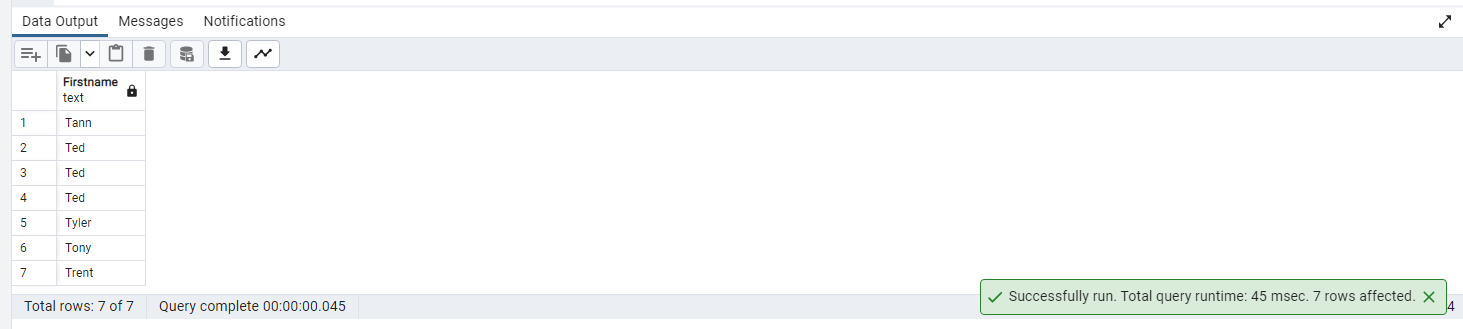
where "Patients"."Visit\_Date" between '2006-2-1' and '2006-2-28';



27. Write a query to get a list of patients whose first names is starting with the letter T.

Query:

select "Firstname" from "Patients" where "Firstname" like 'T%';



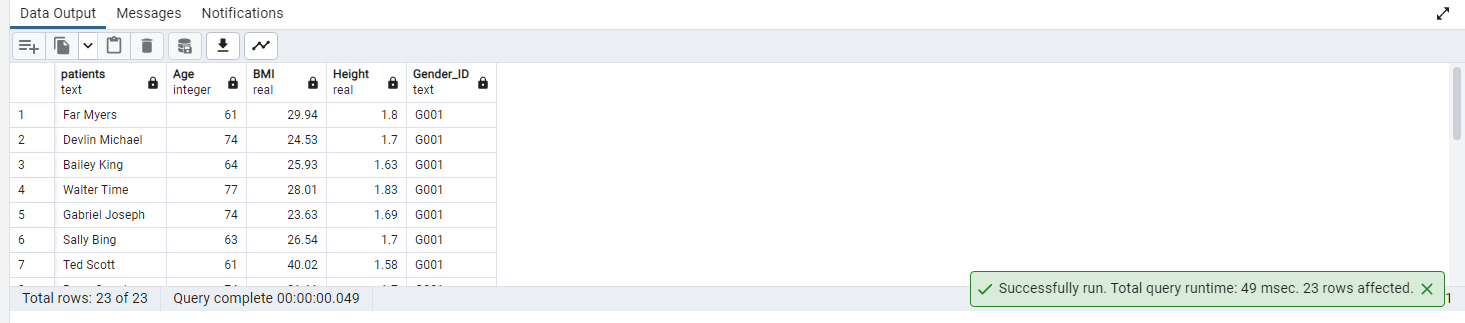
28. Find a list of Male patients whose age is more than 60 whose, BMI is more than 18.5, and whose height is more than e 1.5 M.

Query:

select concat("Firstname",' ',"Lastname") Patients,"Age","BMI","Height","Gender\_ID" from "Patients"

where "Age">60 and "BMI">18.5 and "Height" >1.5

and "Gender\_ID" in (select "Gender\_ID" from "Gender" where "Gender"='Male') ;



29. Write a query to get ceiled creatinine levels for male who age is greater than 35 and levels are abnormal.

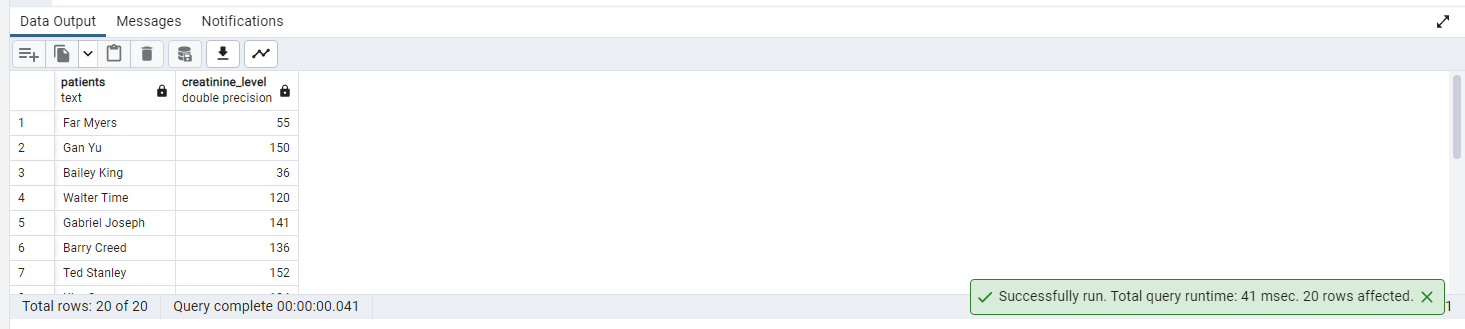
Query:

select concat(pat."Firstname",' ',pat."Lastname") Patients,ceil(ut."Creatinine") creatinine\_level from "Patients" pat join

"Link\_Reference" lr on lr."Link\_Reference\_ID"=pat."Link\_Reference\_ID"

join "Urine\_Test" ut on ut."Urine\_ID"=lr."Urine\_ID"

where ut."Creatinine" not between '65.4' and '119.3' and pat."Gender\_ID"='G001' and pat."Age" >35;



30. Write a query to get the number of patients who visited the Lab between 9 am to 12 am.

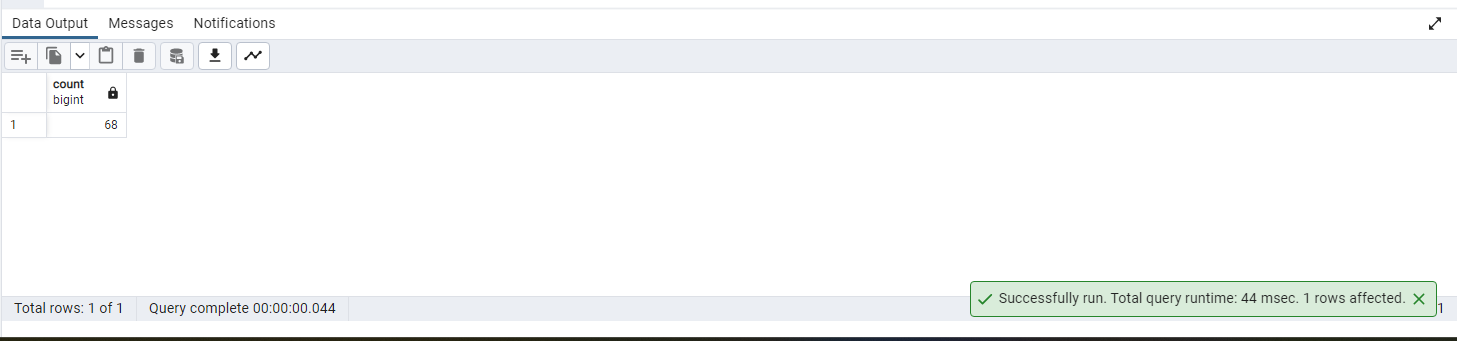
Query:

select count(\*) from "Patients" pat

join "Link\_Reference" lr on pat."Link\_Reference\_ID"=lr."Link\_Reference\_ID"

join "Lab\_Visit" lv on lv."Lab\_visit\_ID"=lr."Lab\_visit\_ID"

where date\_part('hour',lv."Lab\_Visit\_Date") between 9 and 24 ;



31. Write a trigger that calls a function, for checking space and case for

two columns or more before you add new data to a table.

Query:

create or replace function checking\_space\_case() returns trigger as $tgr\_insert$

begin

-- checking for spaces and case while adding a row

if substring(new."Race\_ID",' ') is not null then

raise exception 'cannot have spaces ';

end if;

if initcap(new."Race\_ID") <> new."Race\_ID" then

raise exception 'Please check Title case';

end if;

if substring(new."Race",' ') is not null then

raise exception 'cannot have spaces ';

end if;

if initcap(new."Race") <> new."Race" then

raise exception 'Please check Title case';

end if;

return new;

end;

$tgr\_insert$ language plpgsql;

----trigger---

CREATE TRIGGER tgr\_insert

BEFORE INSERT ON "Race"

FOR EACH ROW

EXECUTE function checking\_space\_case();

------------------------------------------

INSERT INTO "Race" ("Race\_ID","Race")

VALUES(' R053','abcD');

---------------------------- A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with low confidence

32. Write a query to calculate the running moving averages of diabetes\_duration

for Group 2 using the moving windows/sliding dynamic average windows.

Query:

select pat."Visit\_Date"::date,

avg(pat."Diabetes\_Duration") over

(order by pat."Visit\_Date"::date rows between 3 preceding and current row ) moving\_Avg

from "Patients" pat

where pat."Group\_ID"='GRP\_02';

A picture containing text, screenshot, software

Description automatically generated

33. Write a query to create a table to get patients’ demographic details whose birth year is 1939. Name the table as “Patient\_Detail

Query:

create table patient\_detail(patient\_id varchar,first\_name varchar,last\_name varchar,age int, gender varchar,race varchar);

insert into patient\_detail  
(select "Patients"."Patient\_ID" , "Patients"."Firstname" ,"Patients"."Lastname", "Patients"."Age", "Gender"."Gender","Race"."Race"  
from "Patients" inner join "Gender" on "Patients"."Gender\_ID"= "Gender"."Gender\_ID"  
inner join "Race" ON "Race"."Race\_ID" = "Patients"."Race\_ID"  
WHERE EXTRACT(YEAR FROM "Patients"."Visit\_Date") - "Patients"."Age" = 1939);

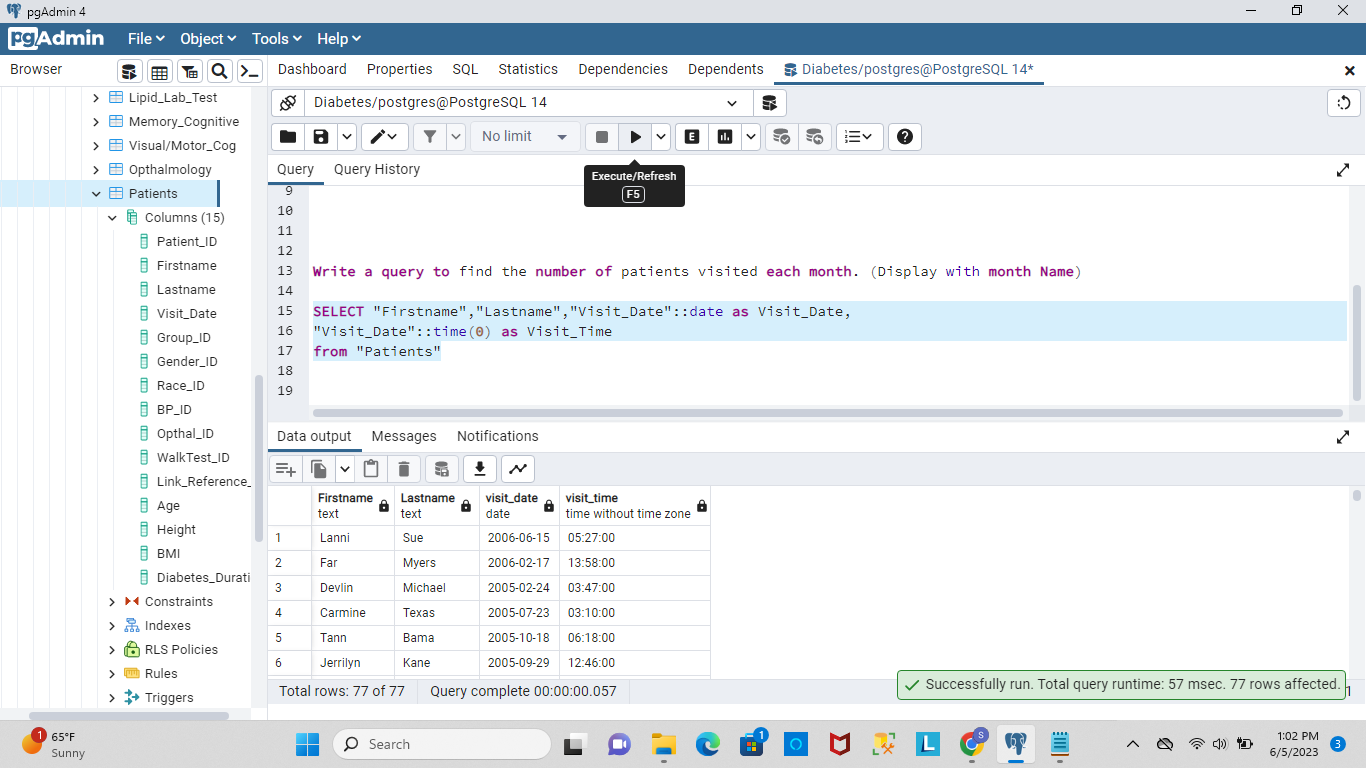
A screenshot of a computer

Description automatically generated

# 34. Convert and display Timestamp and date into visit time from Patient table on visit dates

Query:

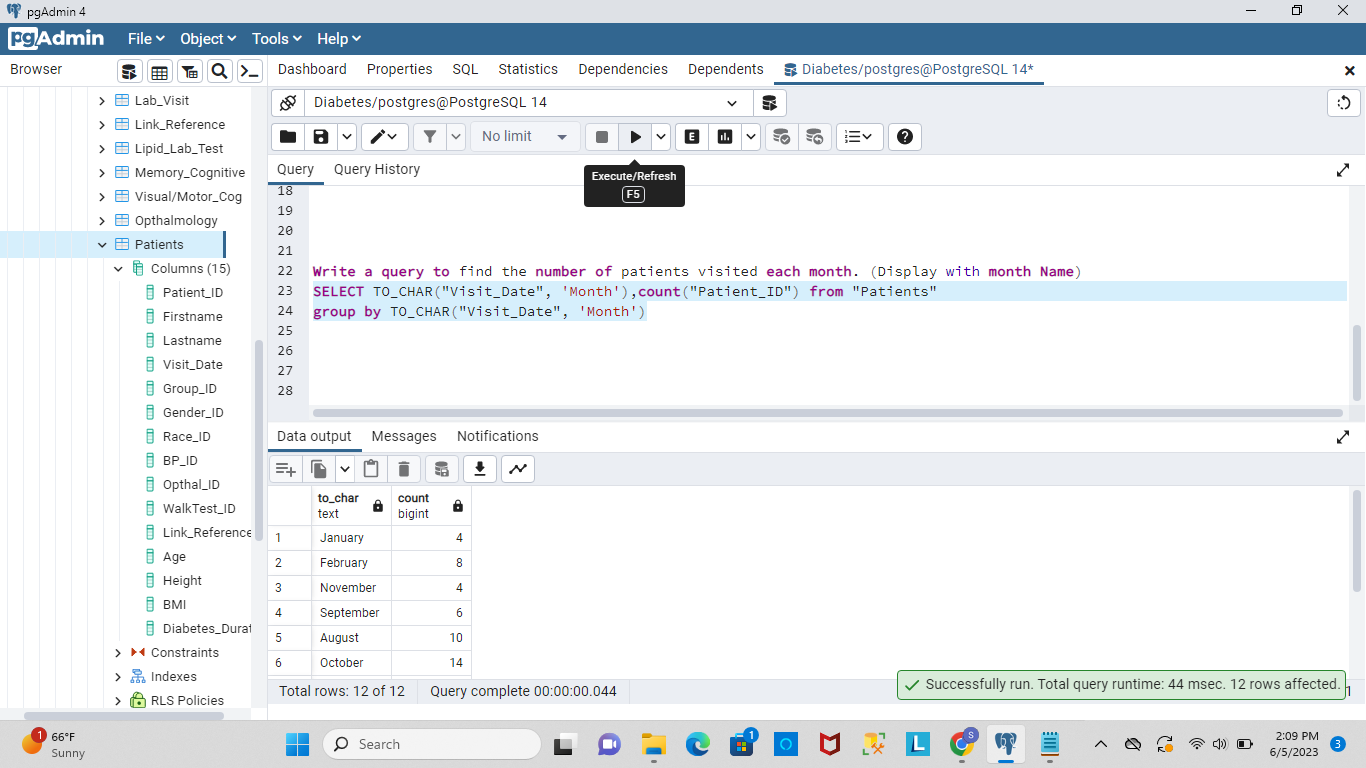
SELECT "Firstname","Lastname","Visit\_Date"::date as Visit\_Date,  
"Visit\_Date"::time(0) as Visit\_Time  
from "Patients";



35. Write a query to find the number of patients visited each month. (Display with month Name)

Query:

SELECT TO\_CHAR("Visit\_Date", 'Month'),count("Patient\_ID") from "Patients"  
group by TO\_CHAR("Visit\_Date", 'Month');



36. Write a query to get a number of visual/motor dementia patients who have any 2 abnormal conditions. (Display with condition name). (dementia/cognitive impairment: any patient who has any two abnormal test results).

Query:

select "Patient\_ID",  
case  
when "TM">=42 and "Clock"<2 then  'Cognitive Impairment and Dementia'  
when "TM">=42 and "RCFT\_IR"<71 then  'Cognitive Impairment and Rey Complex'  
when  "Clock"<2 and "RCFT\_IR"<71 then  'Dementia and Rey Complex'  
END AS abnormal\_conditions  
from "Visual/Motor\_Cog"  
where ("TM">=42 and "Clock"<2) or ("TM">=42 and "RCFT\_IR"<71) or  
("Clock"<2 and "RCFT\_IR"<71);

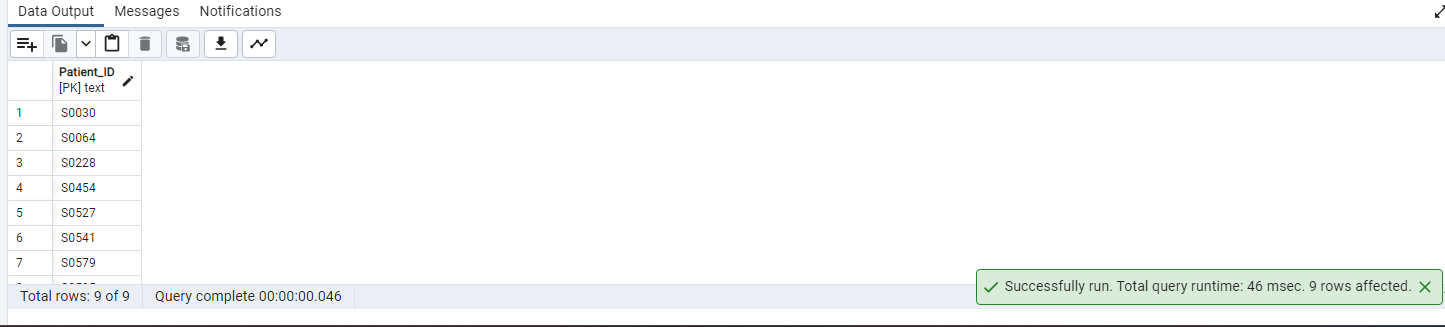
A screenshot of a computer

Description automatically generated

37. Write a query to get a list of patient IDs whose fasting glucose is 80, 85, and 89.

Query:

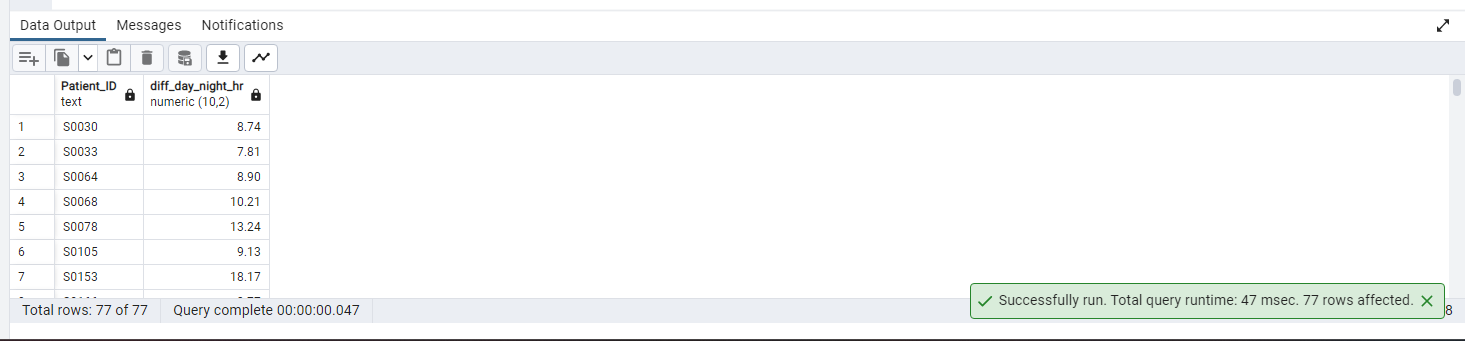
select "Patients"."Patient\_ID" from "Patients" inner join "Lab\_Test" on "Patients"."Patient\_ID"= "Lab\_Test"."Patient\_ID"  
where "Lab\_Test"."Fasting\_Glucose" IN(80,85,89);

****

# 38. calculate the difference between Day and night HR. (Display 2 decimal only)

Query:

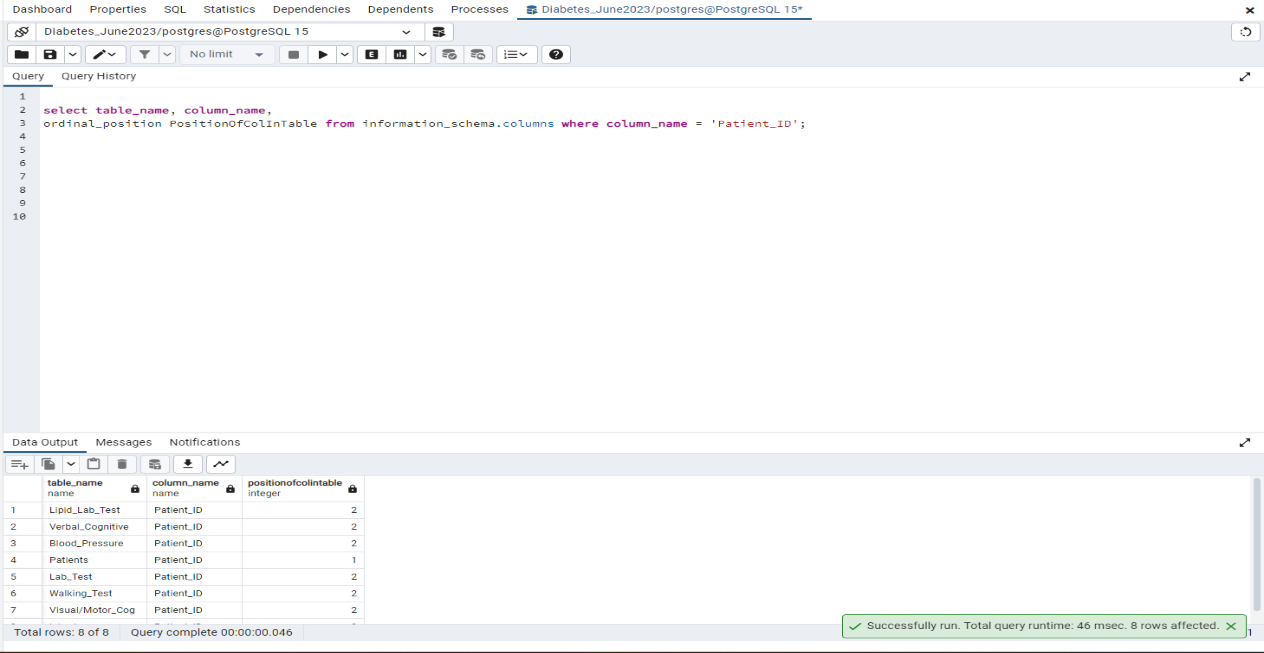
select "Patient\_ID",cast(("24Hr\_Day\_HR"-"24Hr\_Night\_HR")as numeric(10,2)) as diff\_day\_night\_HR from "Blood\_Pressure";



39. Find out the tables where column Patient\_ID is present. (Display column position number with respective table also)

Query:

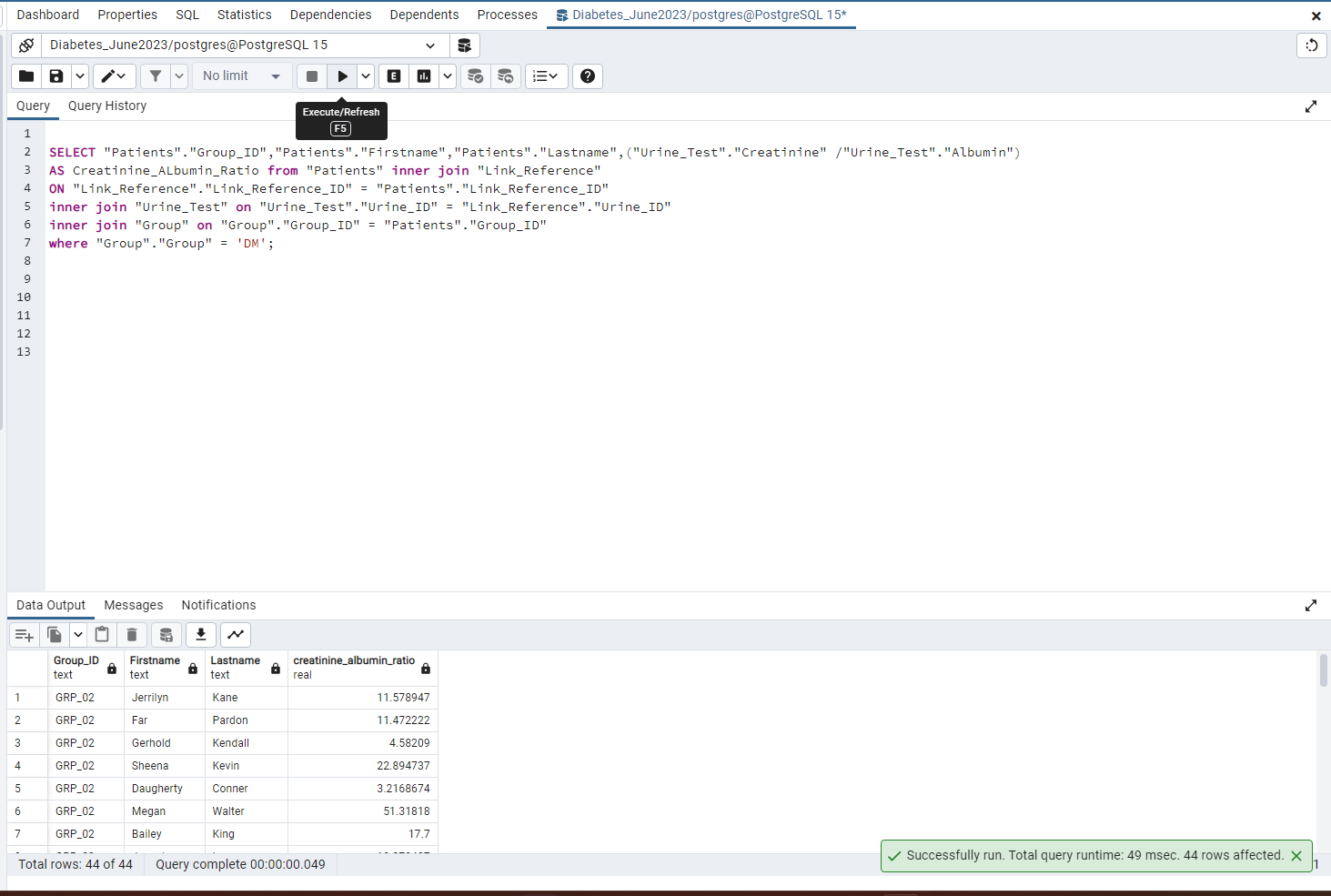
select table\_name, column\_name,  
ordinal\_position PositionOfColInTable from information\_schema.columns where column\_name = 'Patient\_ID';



40. Write a query to calculate Creatinine ALbumin Ratio (uCAR) For DM Patients.

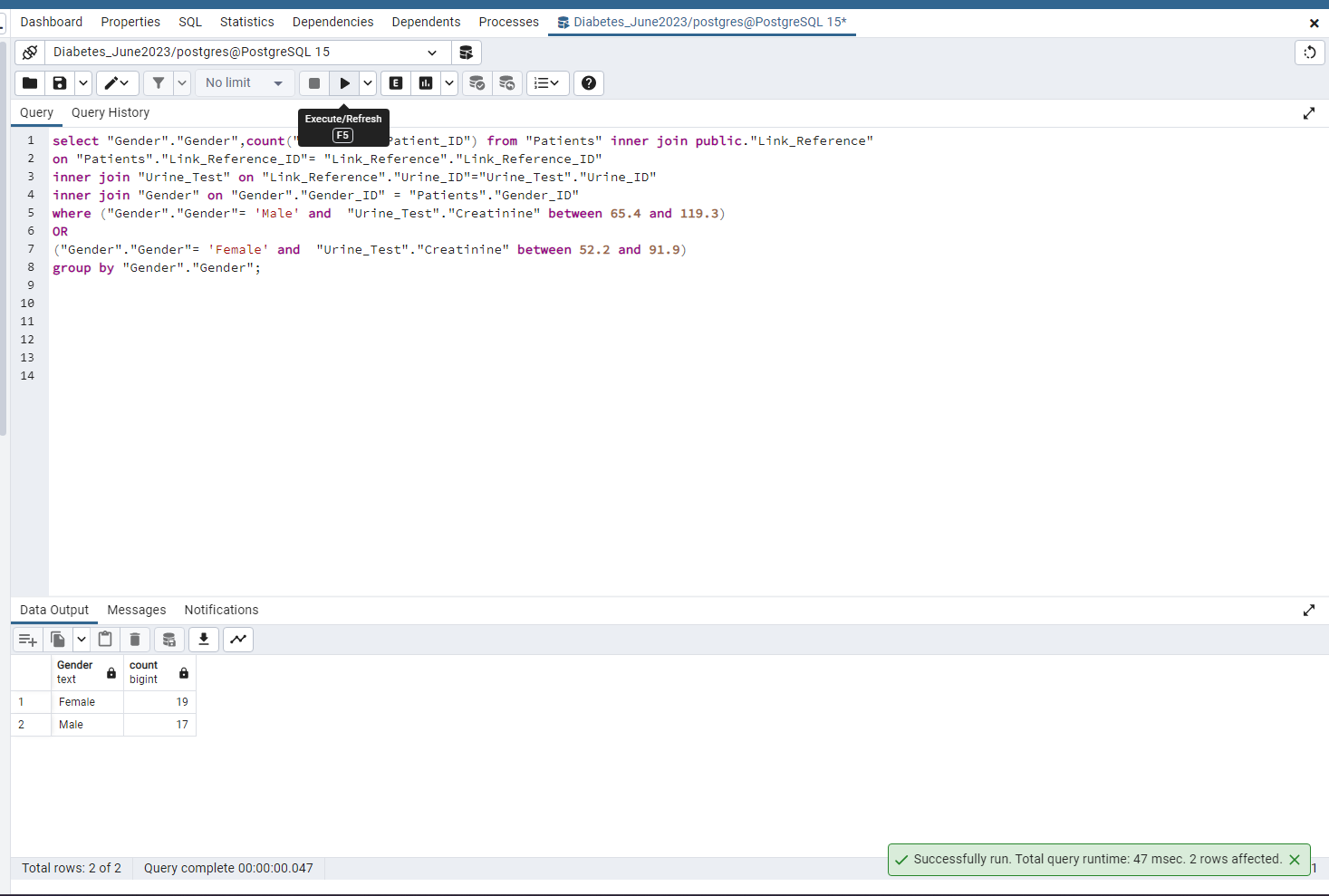
Query:

SELECT "Patients"."Group\_ID","Patients"."Firstname","Patients"."Lastname",("Urine\_Test"."Creatinine" /"Urine\_Test"."Albumin")  
AS Creatinine\_ALbumin\_Ratio from "Patients" inner join "Link\_Reference"  
ON "Link\_Reference"."Link\_Reference\_ID" = "Patients"."Link\_Reference\_ID"  
inner join "Urine\_Test" on "Urine\_Test"."Urine\_ID" = "Link\_Reference"."Urine\_ID"  
inner join "Group" on "Group"."Group\_ID" = "Patients"."Group\_ID"  
where "Group"."Group" = 'DM';



41. write a query to get the number of patients whose urine creatinine is in a normal range (Gender wise).  
  
Query:

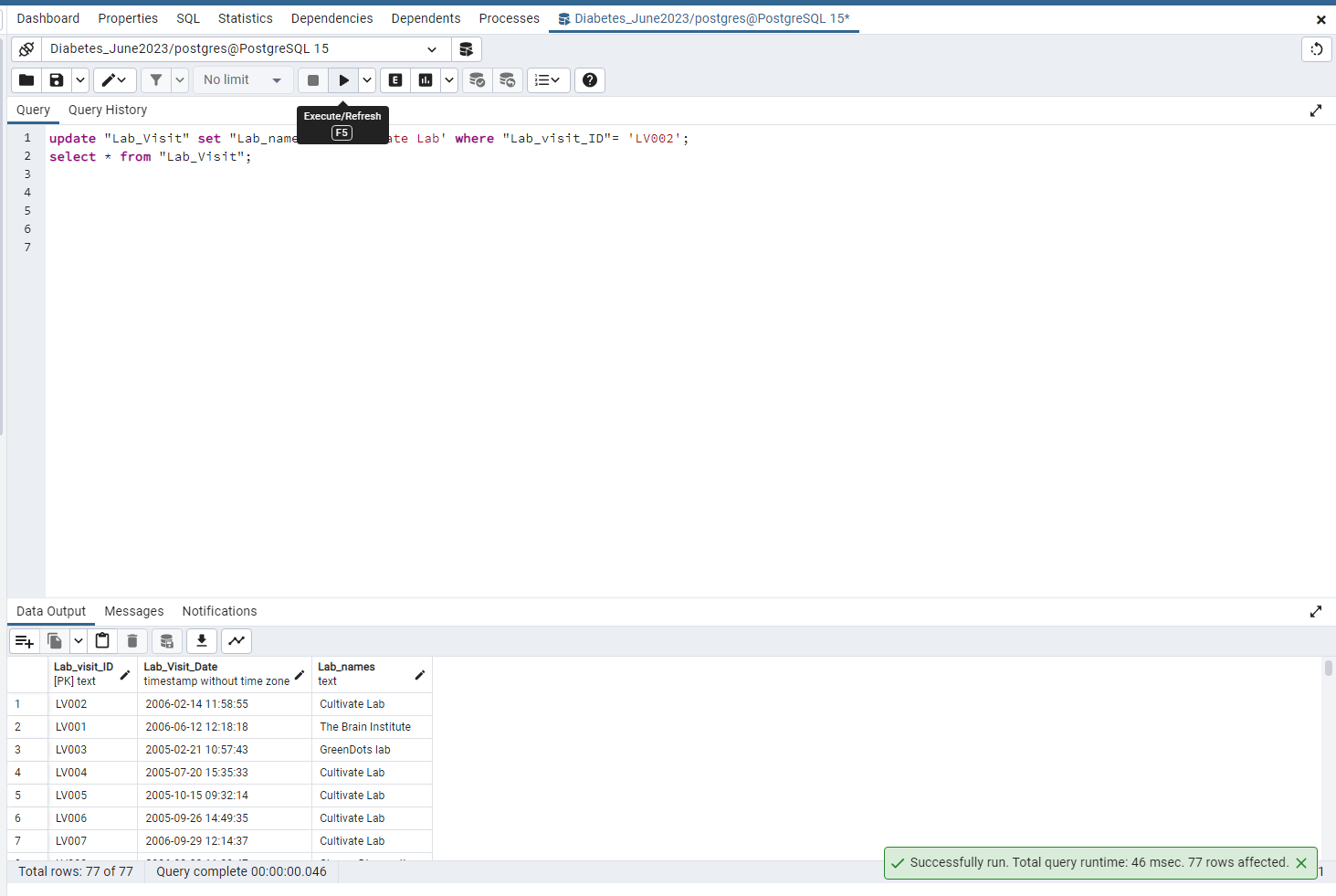
select "Gender"."Gender",count("Patients"."Patient\_ID") from "Patients" inner join public."Link\_Reference"  
on "Patients"."Link\_Reference\_ID"= "Link\_Reference"."Link\_Reference\_ID"  
inner join "Urine\_Test" on "Link\_Reference"."Urine\_ID"="Urine\_Test"."Urine\_ID"  
inner join "Gender" on "Gender"."Gender\_ID" = "Patients"."Gender\_ID"  
where ("Gender"."Gender"= 'Male' and  "Urine\_Test"."Creatinine" between 65.4 and 119.3)  
OR  
("Gender"."Gender"= 'Female' and  "Urine\_Test"."Creatinine" between 52.2 and 91.9)  
group by "Gender"."Gender";



# 42. Write a query to update id LB002 with the lab name Cultivate Lab.

Query:

update "Lab\_Visit" set "Lab\_names" ='Cultivate Lab' where "Lab\_visit\_ID"= 'LV002';  
select \* from "Lab\_Visit";



# 43. Create an index on any table and use explain analyze to show differences if any.

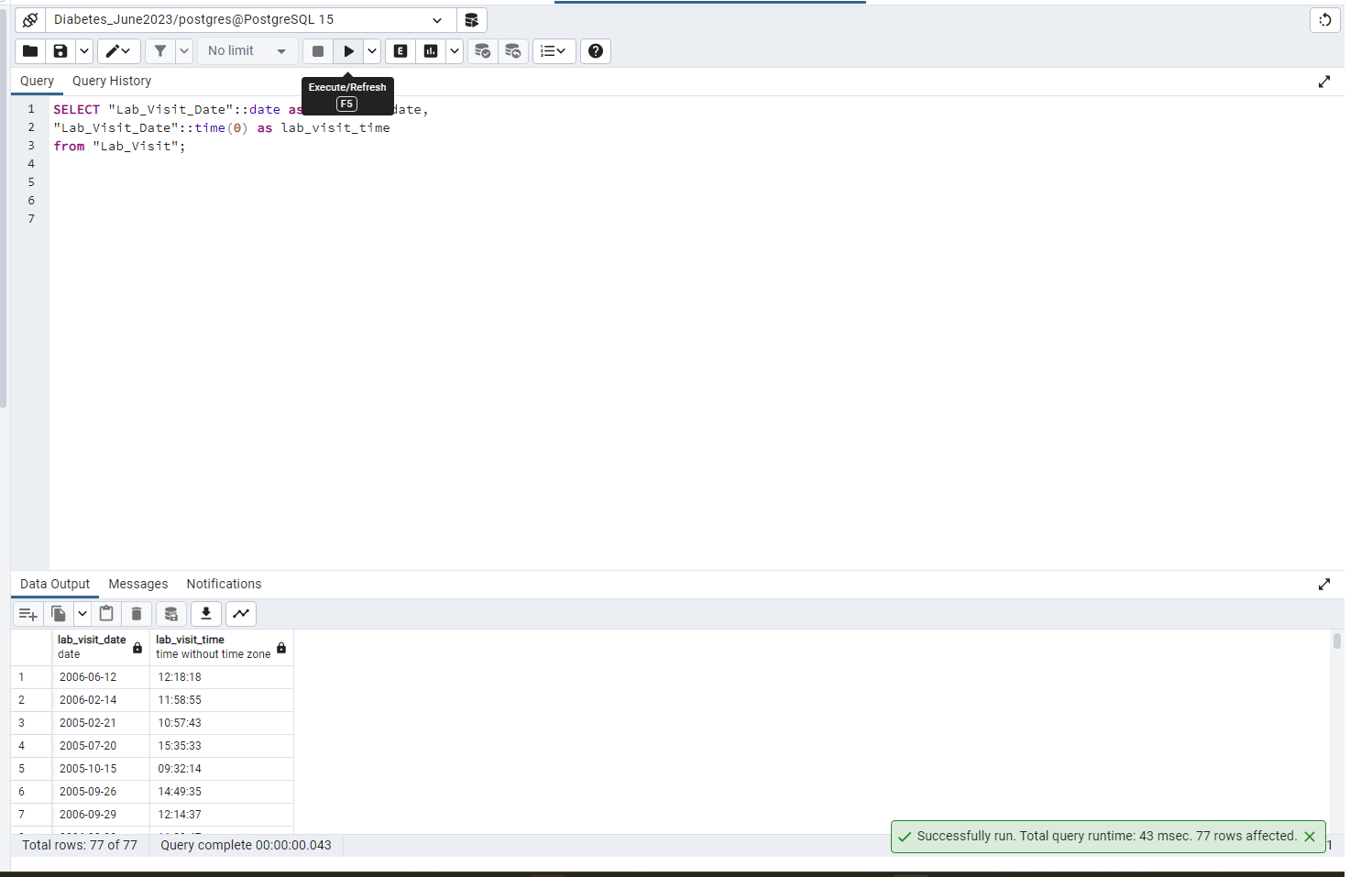
Query:

CREATE INDEX idx\_Race ON "Patients"("Race\_ID");  
EXPLAIN ANALYZE SELECT \* FROM "Patients" where "Race\_ID"='R002'

44. Write a query to split the lab visit date into two different columns lab\_visit\_date  and lab\_visit\_time.

Query:

SELECT "Lab\_Visit\_Date"::date as lab\_visit\_date,  
"Lab\_Visit\_Date"::time(0) as lab\_visit\_time  
from "Lab\_Visit";



45. Please go through the below screenshot and create the exact output.

A screenshot of a computer

Description automatically generated with medium confidence



Query:

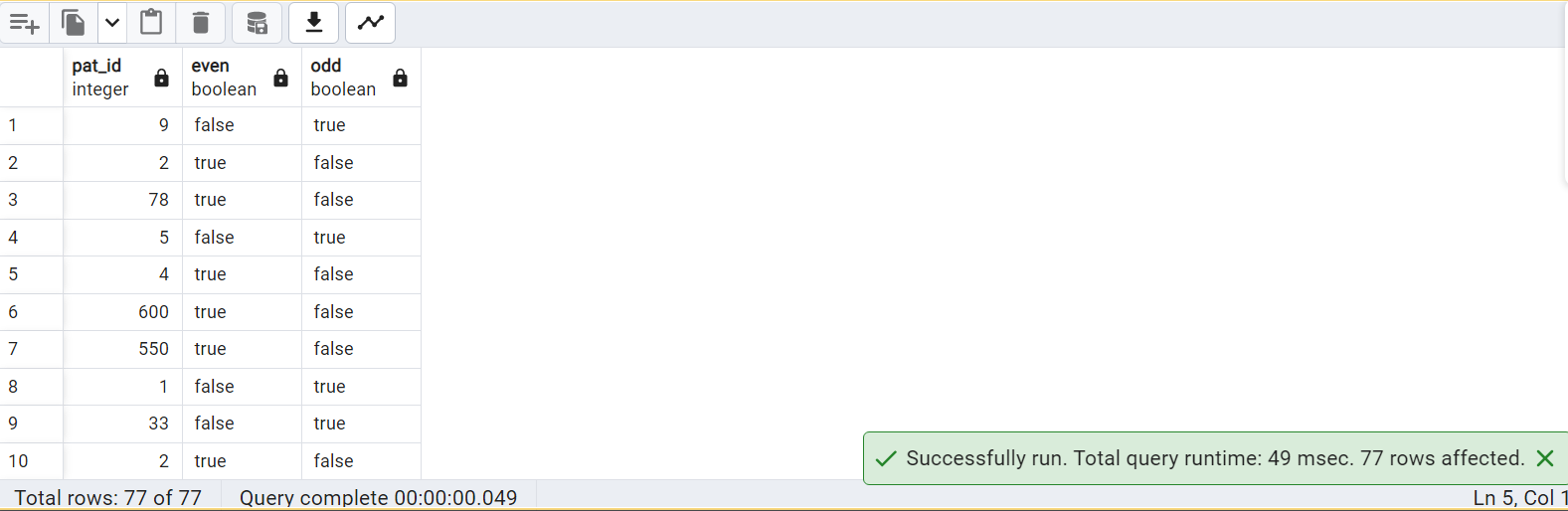
select SUBSTRING("Patient\_ID",POSITION('0' IN REVERSE("Patient\_ID"))+1,LENGTH("Patient\_ID"))::INTEGER pat\_id,

CASE WHEN CAST( SUBSTRING("Patient\_ID", 3, 3) AS int)%2 = 0 THEN 'true'::boolean ELSE 'false'::boolean END as even,

CASE WHEN CAST( SUBSTRING("Patient\_ID", 3, 3) AS int)%2 != 0 THEN 'true'::boolean ELSE 'false'::boolean END as odd

FROM "Patients"

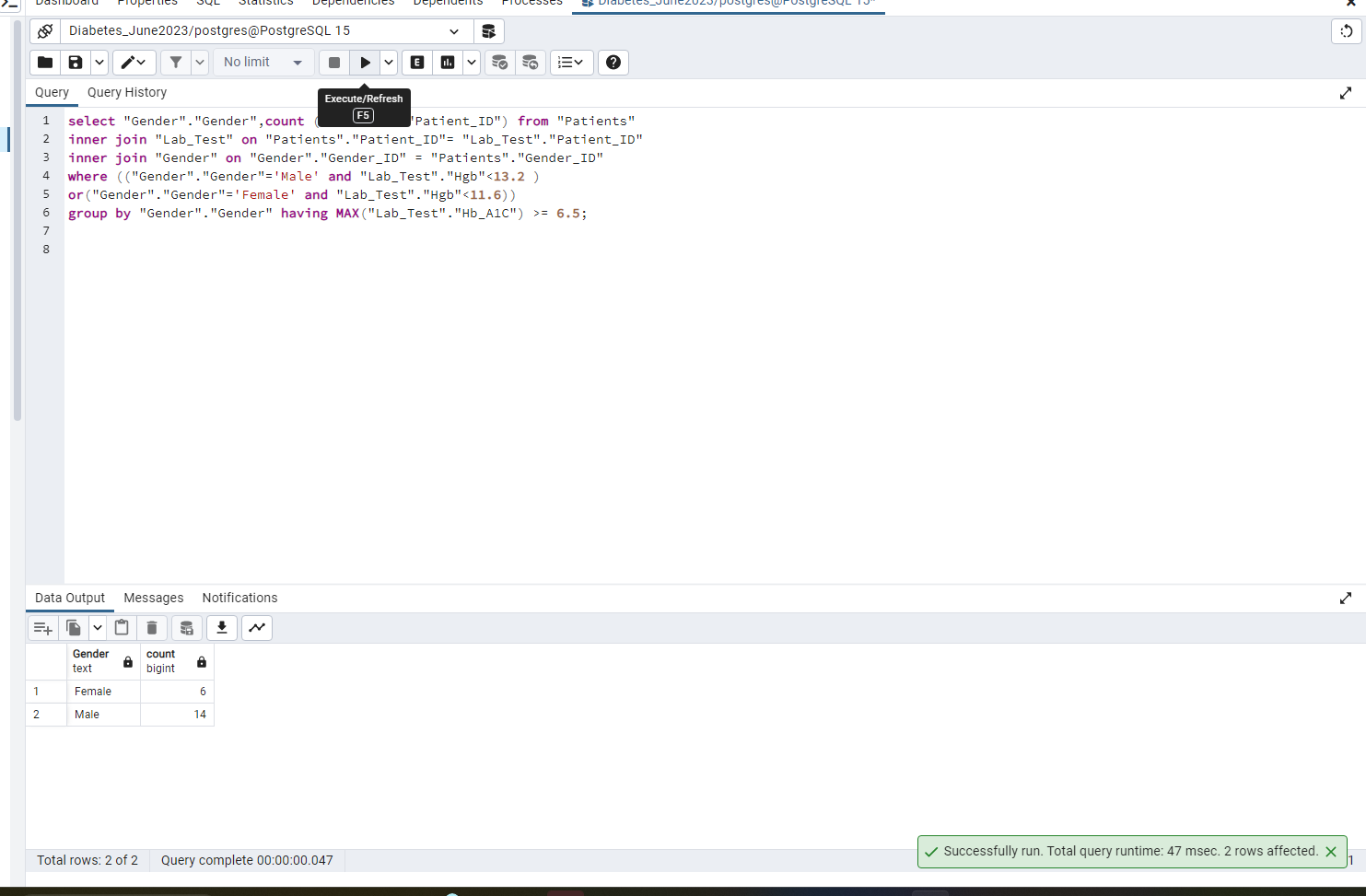
ORDER BY RANDOM();



46. Calculate the Number of Diabetic Male and Female patients who are Anemic.

Query:

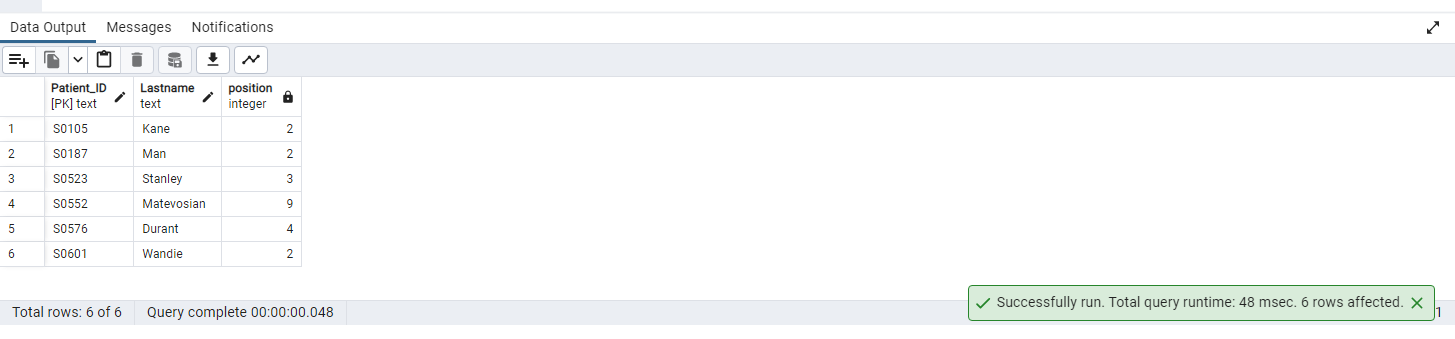
select "Gender"."Gender",count ("Patients"."Patient\_ID") from "Patients"  
inner join "Lab\_Test" on "Patients"."Patient\_ID"= "Lab\_Test"."Patient\_ID"  
inner join "Gender" on "Gender"."Gender\_ID" = "Patients"."Gender\_ID"  
where (("Gender"."Gender"='Male' and "Lab\_Test"."Hgb"<13.2 )  
or("Gender"."Gender"='Female' and "Lab\_Test"."Hgb"<11.6))  
group by "Gender"."Gender" having MAX("Lab\_Test"."Hb\_A1C") >= 6.5;



47. Write a query to display the Patient\_ID, last name, and  
the position of the substring 'an' in the last name column for those patients who have a substring 'an'.

Query:

select "Patient\_ID", "Lastname", position('an' in "Lastname") from "Patients"  
where "Lastname" like '%an%';



# 48. List of patients from rows 30-40 without using the where condition

Query:

# SELECT \* FROM "Patients" LIMIT 11 OFFSET 29;

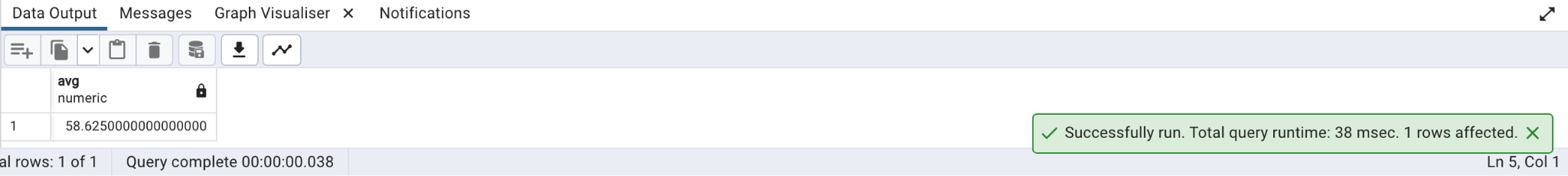
# 

# 49.Write a query to find Average age for patients with high blood pressure.

Query:

select avg(p."Age") from "Patients" p , "Blood\_Pressure" bp

where bp."Patient\_ID" = p."Patient\_ID" and bp."24Hr\_Day\_SBP" > 130 and bp."24Hr\_Day\_DBP" > 80;



50. Create materialized view with no data, to display no of male and female patients.

Query:

CREATE MATERIALIZED VIEW patient\_geneder\_count

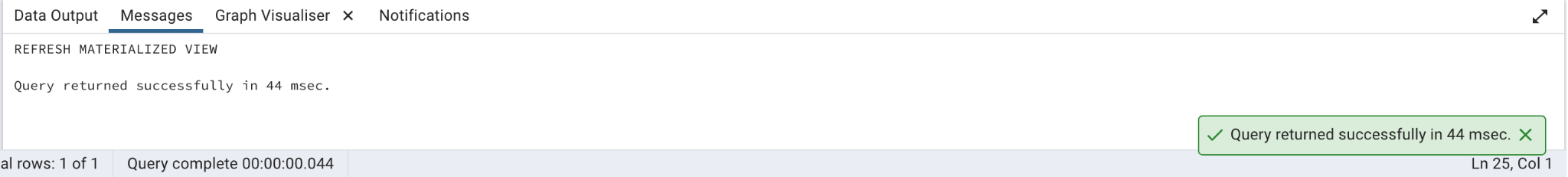
AS

select g."Gender" , count(\*) from "Gender" g , "Patients" p where g."Gender\_ID" = p."Gender\_ID" group by g."Gender"

WITH NO DATA;

REFRESH MATERIALIZED VIEW patient\_geneder\_count;

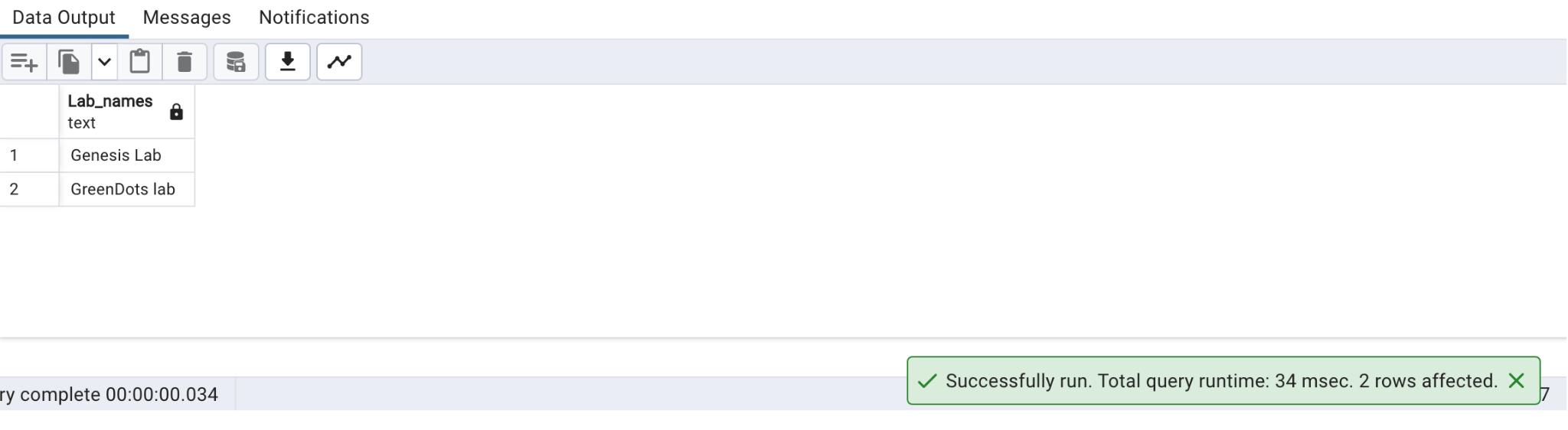
Drop MATERIALIZED VIEW patient\_geneder\_count;



51.Get a list of unique lab names whose names is starting with G and end with b.

Query:

select "Lab\_names" from "Lab\_Visit" where "Lab\_names" like 'G%b' Group By 1;



52. Write the query to create an Index on table Verbal\_Cognitive by selecting a column and also write the query drop the same index.

Query:

Create Index Verbal\_Cognitive\_patient On "Verbal\_Cognitive" ("Patient\_ID");

SELECT

tablename,

indexname,

indexdef

FROM

pg\_indexes

WHERE

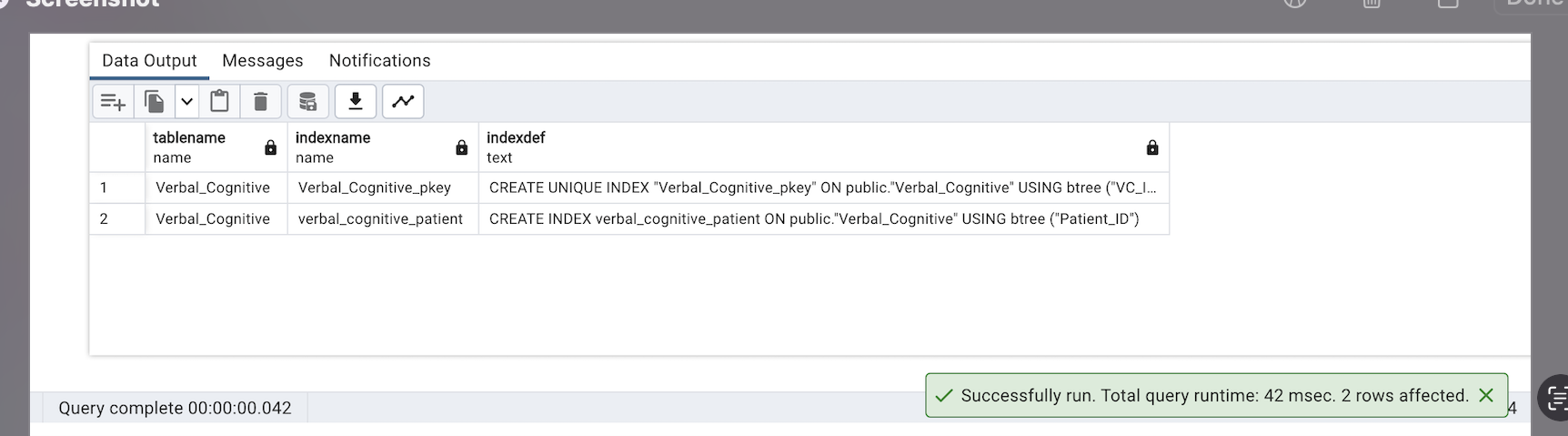
schemaname = 'public'

AND tablename = 'Verbal\_Cognitive'

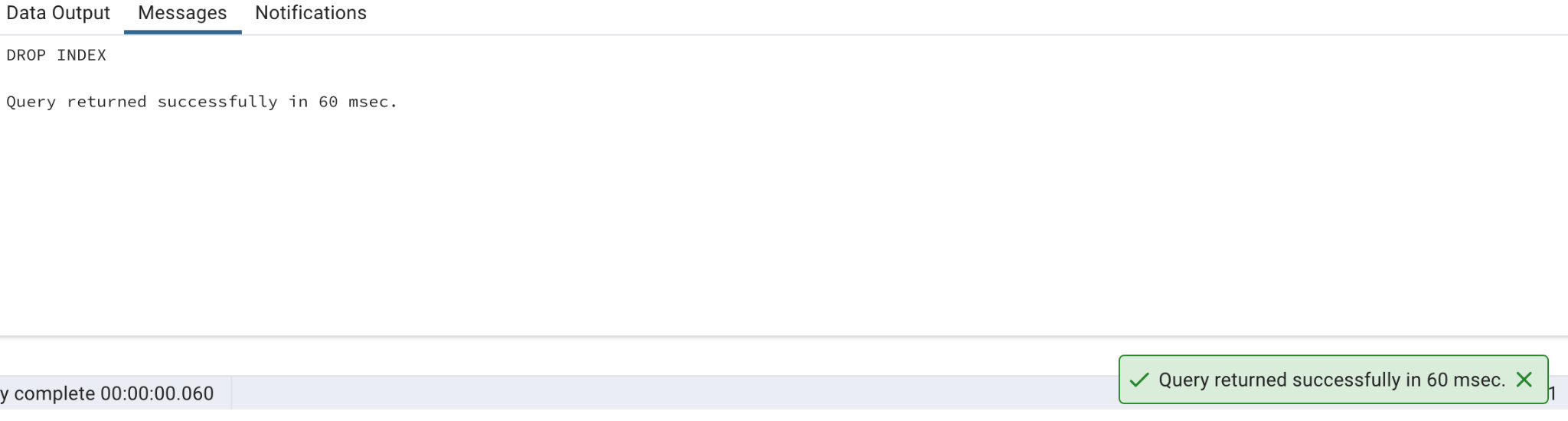
ORDER BY

tablename,

Indexname



Drop Index Verbal\_Cognitive\_patient;



53. Get the number of patients born in a leap year.

Query:

create function is\_leap\_year (double precision)

returns boolean as $$

begin

return date\_part(

'day',

make\_date($1::int, 3, 1) - '1 day'::interval

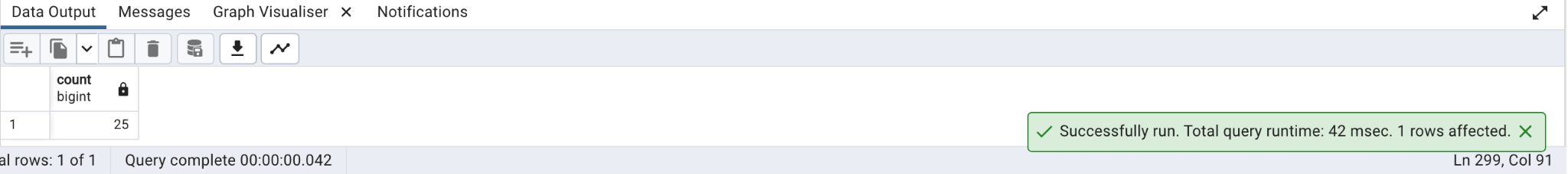
) = 29;

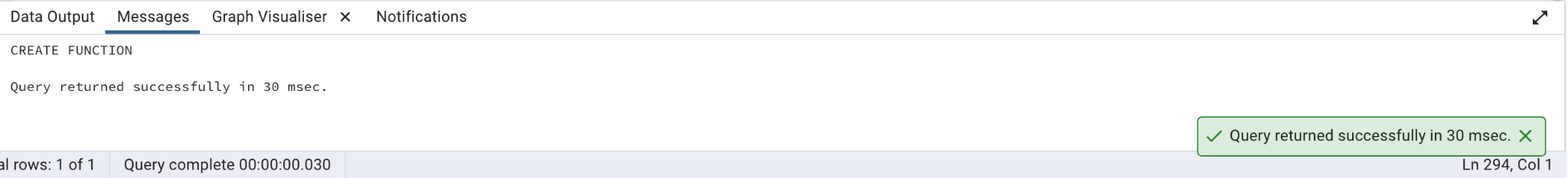
end

$$ language plpgsql;

drop function is\_leap\_year;

select count(\*) from "Patients" where is\_leap\_year(DATE\_PART('year',"Visit\_Date") - "Age")



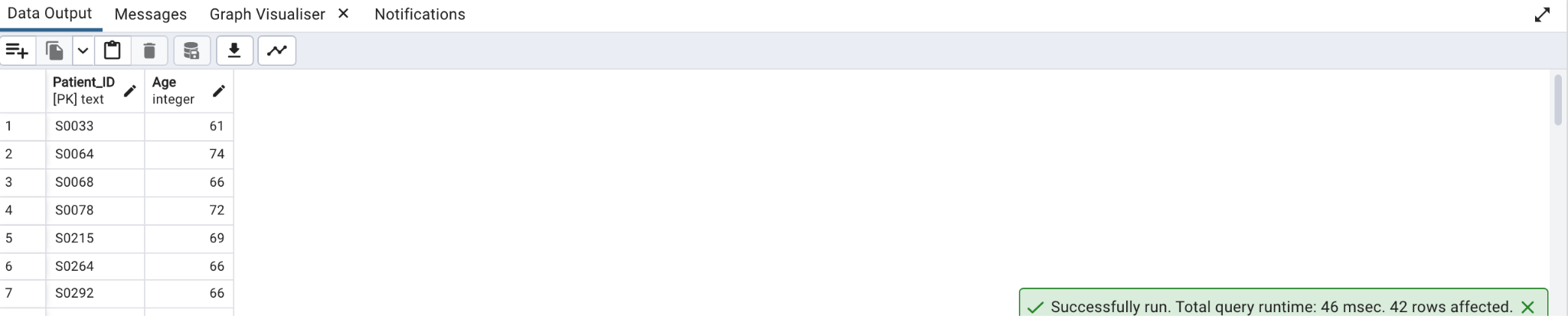


# 54. Write a query to get a list of patient IDs from the DM group and above age 60 in sequence.

select "Patient\_ID","Age" from "Patients"

where "Age">60;

Query:



55.Find the patient who has the most damage in the eyes with the use of a max function.

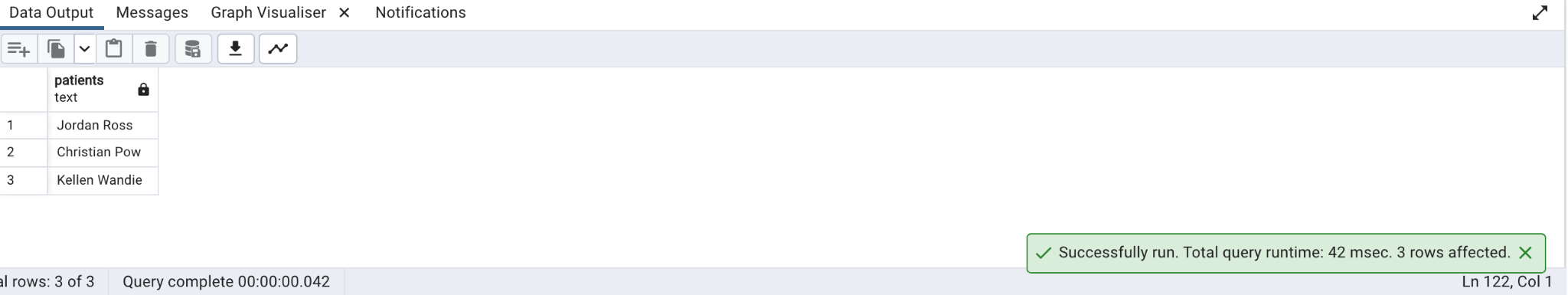
Query:

select "Firstname"||' '||"Lastname" Patients

from "Patients" as pat , "Opthalmology" AS opt where pat."Opthal\_ID" = opt."Opthal\_ID" and

opt. "Diabetic\_Retinopathy" = ( select max( "Diabetic\_Retinopathy") from "Opthalmology" )

and opt."Macular\_Edema" = (select max("Macular\_Edema" ) from "Opthalmology" );



# 

# 56. Create a procedure for checking if Race exists using an if else statement.

Query:

CREATE OR REPLACE PROCEDURE public.chkrace(

IN \_inrace text)

LANGUAGE 'plpgsql'

AS $BODY$

declare raceout bigint;

begin

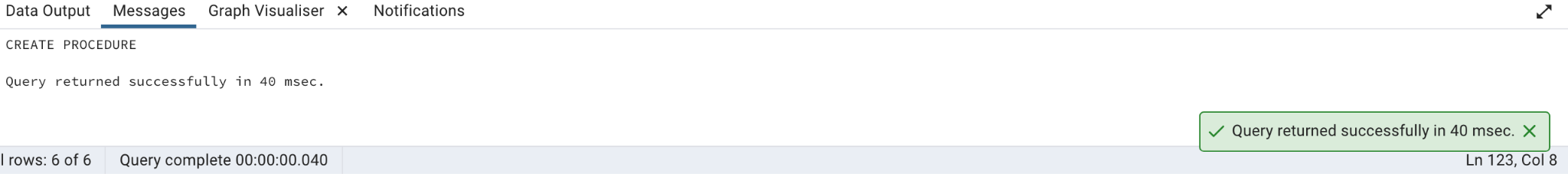
raceout= (select count(\*) from public."Race"

where "Race"=\_inrace);

RAISE notice 'No. Of Races Found: %', raceout;

end;

$BODY$;



57.Write a query to display the DM patients and their high fasting triglycerides based upon their age ,gender and race.

Query:

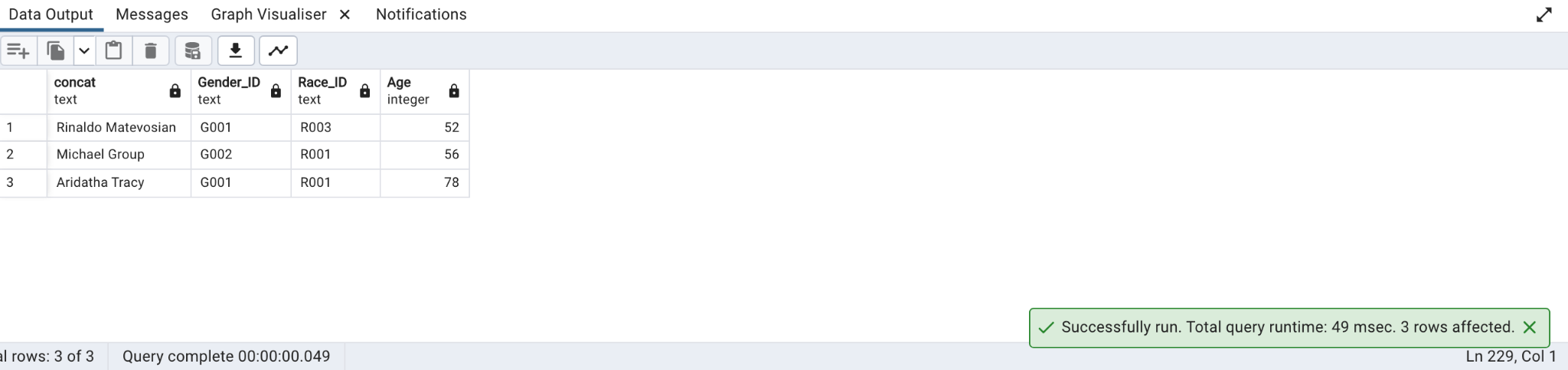
Select concat(pt."Firstname" ,' ',pt."Lastname"),pt."Gender\_ID",pt."Race\_ID",pt."Age" from

"Patients" pt, "Lipid\_Lab\_Test" l,"Link\_Reference" lr

Where l."Lipid\_ID"=lr."Lipid\_ID" and lr."Link\_Reference\_ID"=pt."Link\_Reference\_ID"

And l. "Fasting\_Triglyc" > 200

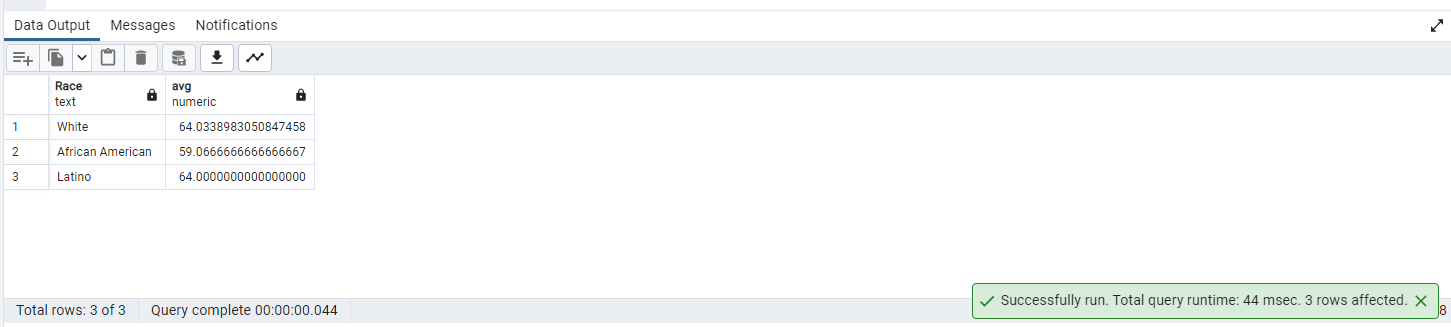
And pt."Group\_ID" ='GRP\_02';



58. Create a pie chart based on race vs age.

Query:

SELECT A."Race",avg(pat."Age") from public."Race" A, public."Patients" pat where pat."Race\_ID"=A."Race\_ID" group by 1



59. Write a query to create a master Patient table and its child table. Make sure that the child table inherits all the fields from the parent Patient table.

Query:

CREATE TABLE PatientsMaster (

Patient\_ID text,

Age Int

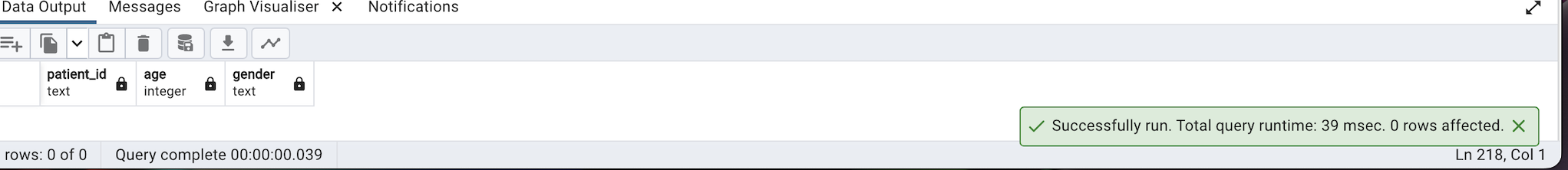
);

CREATE TABLE PatientsGender (

Gender text

) INHERITS (PatientsMaster);

select \* from PatientsGender



60. Write a query using the trigger after insert on the lab test table. If the patient has abnormal HbA1C and fasting glucose values.

Query:

CREATE OR REPLACE FUNCTION Lab\_Test\_view() RETURNS TRIGGER AS $$

BEGIN

--write function

if (New."Hb\_A1C">6.4)then

raise notice 'HBAc high';

end if;

if (New."Fasting\_Glucose">120)then

raise notice 'Fasting Glucose high';

end if;

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

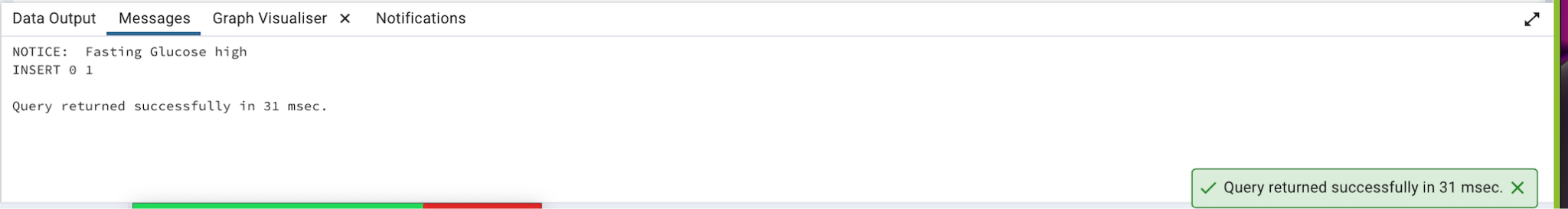
Create trigger Lab\_tgr

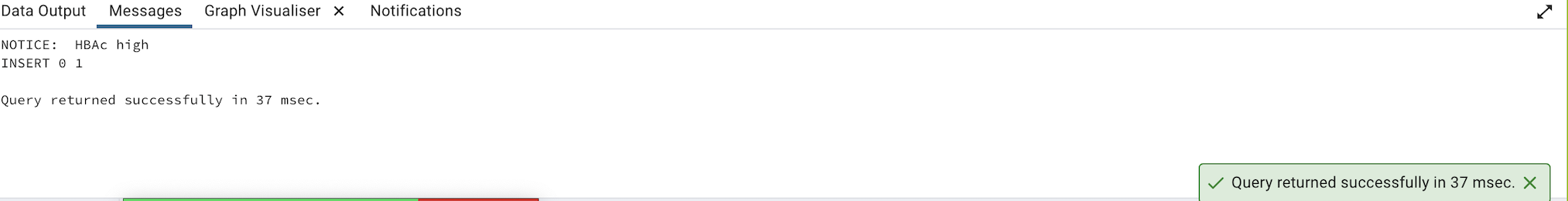
After insert ON public."Lab\_Test"

for each row Execute function Lab\_Test\_view();

insert into public."Lab\_Test" ("Lab\_ID", "Patient\_ID" ,"Hb\_A1C","Fasting\_Glucose") Values('3','5','6.6','71');

insert into public."Lab\_Test" ("Lab\_ID", "Patient\_ID" ,"Hb\_A1C","Fasting\_Glucose") Values('5','7','6.2','131');





61. write a query to get the number of patients for each age bin without using the CASE statement.(Bin size - 5)

Query:

with AgeInt as (

select "Age"::int Age, "Patient\_ID" PID from public."Patients"

)

select width\_bucket(Age,0,100,20) as Agebucket,

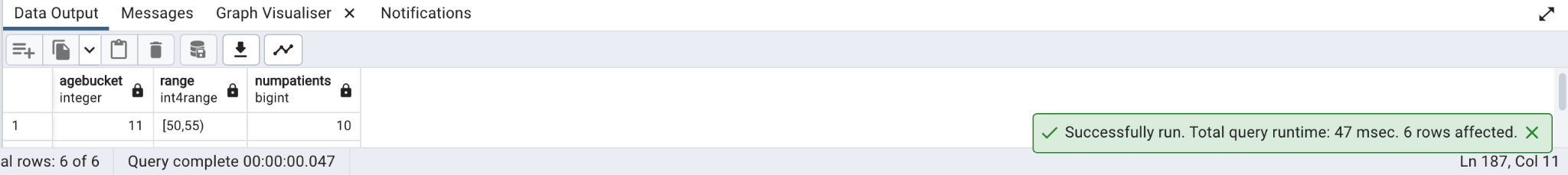
int4range(min(Age),max(Age),'[]') as range,

count(PID) as NumPatients

from AgeInt

group by 1

order by 1



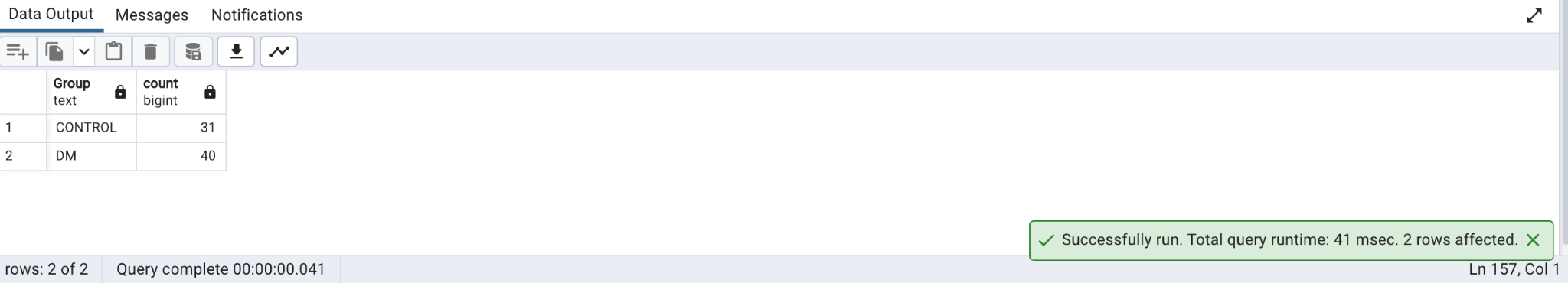
62.Write a query to get the number of patients who have normal platelets for each group.

Query:

select g."Group", count(p."Patient\_ID") from "Patients" p, "Group" g where p."Group\_ID" = g."Group\_ID" and "Patient\_ID" in (

select "Patient\_ID" from "Lab\_Test" where "Platelets" between 150 and 450)

group by g."Group"



63.Create a trigger on a view of the Blood Pressure table.

Query:

Create view blood\_pressure\_view as

(

Select \* from "Blood\_Pressure"

)

Create trigger tgr\_view

instead of delete ON public.Blood\_Pressure\_view

for each row Execute function Blood\_Pressure\_view();

CREATE OR REPLACE FUNCTION Blood\_Pressure\_view() RETURNS TRIGGER AS $$

BEGIN

--write function

if (tg\_op='DELETE') then

raise notice 'for delete';

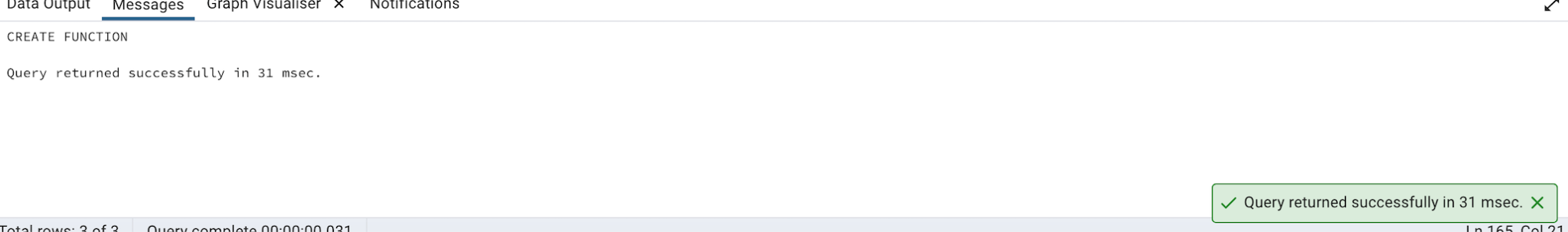
end if;

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

Delete from public.Blood\_Pressure\_view



64.Write a query to find the number of Patients whose Gait RPE start is greater than the end and vice versa. (Display exact output as shown below)

Query:

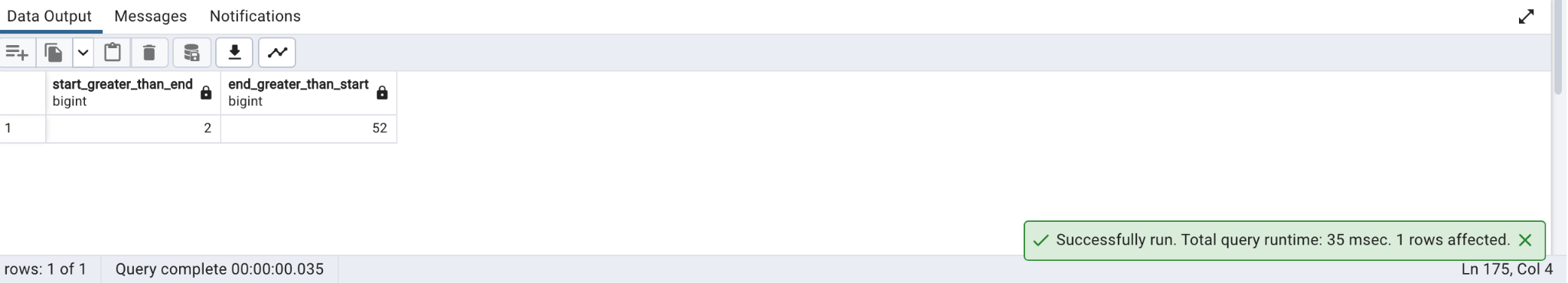
select max(start\_greater\_than\_end) start\_greater\_than\_end, max(end\_greater\_than\_start) end\_greater\_than\_start from (

select count(wt."Patient\_ID") as start\_greater\_than\_end, 0 as end\_greater\_than\_start from "Walking\_Test" wt where (wt."Gait\_RPE\_Start ") > wt."Gait\_RPE\_End "

union all

select 0 as start\_greater\_than\_end,count(wt."Patient\_ID") as end\_greater\_than\_start from "Walking\_Test" wt where wt."Gait\_RPE\_Start " < wt."Gait\_RPE\_End "

) A



65.Create a view without using any schema or table and check the created view using a select statement.

Query:

Create or Replace View NoTabView

as

(

Select 1 Col1, 2 Col2, 3 Col3

);

select \* from NoTabView;

A screenshot of a chat

Description automatically generated with medium confidence

66.Display patients names who have the same last name.

Query:

select pat."Firstname"||' '||pat."Lastname" from public."Patients" Pat

where

upper(pat."Lastname") in

(

select

upper("Lastname") lastname from public."Patients" group by upper("Lastname")

having count(1)>1)

order by pat."Lastname";

A picture containing text, screenshot, font

Description automatically generated

67.Write a query to get the Sum of Diabetes Duration for Group id 'GRP\_02'.

Query:

select sum(pat."Diabetes\_Duration") DiabetesDuration ,dgroup."Group\_ID"

From public."Patients" Pat

left join public."Group" dgroup

on pat."Group\_ID"=dgroup."Group\_ID"

where dgroup."Group\_ID"='GRP\_02'

group by 2;

A screenshot of a computer

Description automatically generated with low confidence

68.Write a query to get a patient name who has a chance to have kidney disease with Albumin

Query:

select pat."Firstname"||' '||pat."Lastname" PatientName, ut."Albumin" AlbuminOver30mg

from public."Patients" pat

join public."Link\_Reference" lref

on pat."Link\_Reference\_ID"=lref."Link\_Reference\_ID"

left join public."Urine\_Test" ut

on lref."Urine\_ID"=ut."Urine\_ID"

where ut."Albumin">30;

A screenshot of a chat

Description automatically generated with low confidence

69.Get the patient's name who has a max speed.

Query:

with TopSpeed as

(

select pat."Firstname"||' '||pat."Lastname" PatientName, rank() over (order by wt."Gait\_DT\_Speed" desc) speedrank from public."Patients" pat

left join public."Walking\_Test" wt

on pat."WalkTest\_ID"=wt."WalkTest\_ID"

)

select \* from TopSpeed where speedrank=1;

A screenshot of a computer

Description automatically generated with medium confidence

70.Write a query to find out the percentage of Lab visits by Lab names.

Query:

SELECT "Lab\_names",

COUNT("Lab\_visit\_ID") "CountLabVisits",

COUNT("Lab\_visit\_ID") / SUM(COUNT("Lab\_visit\_ID")) OVER () \* 100 PercentageLabVisits

FROM public."Lab\_Visit"

GROUP BY 1;

A screenshot of a computer

Description automatically generated with medium confidence

71.Write a query to get Patient IDs for verbally cognitively impaired who satisfy any 2 conditions. (HINT: dementia/cognitive impaired: any patient who has any two abnormal test results).

Query:

With AtLeastTwoConditions as

(

select "Patient\_ID", 'Two Markers Of Dementia' as PatientCondition from

public."Verbal\_Cognitive"

where ("DS"<13 and "HVLT"<14)

group by "Patient\_ID",PatientCondition

Union All

select "Patient\_ID", 'Low Fluency & Adult Reading Scores' as PatientCondition from

public."Verbal\_Cognitive"

where ("VF"<42 and "WTAR"<=20)

group by "Patient\_ID",PatientCondition

Union All

select "Patient\_ID", 'Mixed Markers:verbal-cognitive Imp ' as PatientCondition from

public."Verbal\_Cognitive"

where ("VF"<42 and "HVLT"<14) or

("DS"<13 and "VF"<42) or ("WTAR"<=20 and "HVLT"<14) or

("DS"<13 and "WTAR"<=20)

group by "Patient\_ID",PatientCondition

)

select "Patient\_ID", string\_agg(PatientCondition,' & ') as PatientCondition from AtLeastTwoConditions group by 1;

A screenshot of a computer

Description automatically generated with medium confidence

72.Display a list of patients who are memory cognitively impaired with the GDS test and whose diabetes duration is between 5 to 30.

Query:

select pat."Firstname"||' '||pat."Lastname" PatientName,mcog."GDS", pat."Diabetes\_Duration"

from public."Patients" pat

join public."Link\_Reference" lref

on pat."Link\_Reference\_ID"=lref."Link\_Reference\_ID"

join public."Memory\_Cognitive" mcog

on lref."MC\_ID"=mcog."MC\_ID"

where

mcog."GDS">15 and pat."Diabetes\_Duration" between 5 and 30;

A screenshot of a computer

Description automatically generated with low confidence

73.Write a query to the get number of Patient\_IDs who visited between March 2005 and March 2006

Query:

WITH NameVisit as

(

select Count("Patient\_ID") NumPatients, Cast((concat(Extract(YEAR FROM "Visit\_Date"),to\_char(Extract(MONTH FROM "Visit\_Date"),'FM00'))) as Integer) VisitMonthYear

from public."Patients"

group by 2

)

SELECT NumPatients,

Case

when VisitMonthYear=200503 Then 'Mar-2005'

when VisitMonthYear=200504 Then 'Apr-2005'

when VisitMonthYear=200505 Then 'May-2005'

when VisitMonthYear=200506 Then 'Jun-2005'

when VisitMonthYear=200507 Then 'Jul-2005'

when VisitMonthYear=200508 Then 'Aug-2005'

when VisitMonthYear=200509 Then 'Sep-2005'

when VisitMonthYear=200510 Then 'Oct-2005'

when VisitMonthYear=200511 Then 'Nov-2005'

when VisitMonthYear=200512 Then 'Dec-2005'

when VisitMonthYear=200601 Then 'Jan-2006'

when VisitMonthYear=200602 Then 'Feb-2006'

when VisitMonthYear=200603 Then 'Mar-2006'

end VisitMonYear

FROM NameVisit WHERE VisitMonthYear BETWEEN 200503 AND 200603 ORDER BY VisitMonthYear;

A screenshot of a chat

Description automatically generated with low confidence

74.Get the number of patients who visited each lab using the windows function.

Query:

SELECT labv."Lab\_names",

sum(COUNT(\*)) over (Partition by labv."Lab\_names") NumberVisits

FROM

public."Patients" pat

join public."Link\_Reference" lref

on pat."Link\_Reference\_ID"=lref."Link\_Reference\_ID"

left join public."Lab\_Visit" labv

on labv."Lab\_visit\_ID"=lref."Lab\_visit\_ID"

GROUP BY 1;

A picture containing text, screenshot, font, number

Description automatically generated

75.Find the number of control and DM patients who visited each lab.

Query:

CREATE EXTENSION IF NOT EXISTS tablefunc;

SELECT \* FROM CROSSTAB

(

'select dgroup."Group" CGroup,labv."Lab\_names" LabNames, count(labv."Lab\_names") from

public."Patients" Pat

join public."Link\_Reference" lref

on pat."Link\_Reference\_ID"=lref."Link\_Reference\_ID"

left join public."Lab\_Visit" labv

on labv."Lab\_visit\_ID"=lref."Lab\_visit\_ID"

left join public."Group" dgroup

on pat."Group\_ID"=dgroup."Group\_ID"

group by 1,2

ORDER BY 1,2') AS CT (CGroup text,"Cultivate Lab" bigint,"The Brain Institute" BIGINT,"Optimal Diagnostics" bigint,"Genesis Lab" bigint,"GreenDots lab" bigint,"Stream Diagnostix" bigint);

A screenshot of a computer

Description automatically generated with medium confidence

76.Please go through the below screenshot and create the exact output.

Query:

select pat."Firstname"||' '||pat."Lastname" PatientName, length(pat."Firstname")+length(pat."Lastname")+2 unknown

from public."Patients" pat;

A screenshot of a chat

Description automatically generated with low confidence

77.Write a query to get comma-separated values of patient details .(Use a maximum of 6 columns from different tables)

Query:

--Solution 1: Input Patient Number and get details for that patient. Function output is comma separated by default.

CREATE OR REPLACE FUNCTION getPatientDet(\_patid text)

RETURNS TABLE(FirstName text,LastName text, race text,Age int,BMI real) AS

$$

BEGIN

RETURN QUERY

select pat."Firstname" ,

pat."Lastname" ,

race."Race" ,

pat."Age" Age,

pat."BMI" BMI

from public."Patients" pat

join public."Race" race on

pat."Race\_ID"=race."Race\_ID"

join public."Link\_Reference" lref

on pat."Link\_Reference\_ID"=lref."Link\_Reference\_ID"

join public."Lab\_Test" labt

on labt."Lab\_ID"=lref."Lab\_ID"

and pat."Patient\_ID" = \_patid;

END

$$

LANGUAGE plpgsql;

SELECT getPatientDet('S0030');

A screenshot of a chat

Description automatically generated with medium confidence

--Solution 2: Use a simple concatenate operation

select pat."Firstname"||','|| pat."Lastname"||','||race."Race"||','||pat."Age"||','|| pat."BMI" PatientDetails

from public."Patients" pat

join public."Race" race on

pat."Race\_ID"=race."Race\_ID"

join public."Link\_Reference" lref

on pat."Link\_Reference\_ID"=lref."Link\_Reference\_ID"

join public."Lab\_Test" labt

on labt."Lab\_ID"=lref."Lab\_ID";

A picture containing text, screenshot, font, software

Description automatically generated

--Solution 3: Use string\_agg()

select pat."Firstname"||','|| pat."Lastname" PatientName, string\_agg(race."Race"||' , '||pat."Age"||' , '|| pat."BMI",',') RaceAgeBMI

from public."Patients" pat

join public."Race" race on

pat."Race\_ID"=race."Race\_ID"

join public."Link\_Reference" lref

on pat."Link\_Reference\_ID"=lref."Link\_Reference\_ID"

join public."Lab\_Test" labt

on labt."Lab\_ID"=lref."Lab\_ID"

group by 1;

A picture containing text, screenshot, font

Description automatically generated

78.Write a query to determine get the Patient IDs ,in DM and Control groups ,that are in prediabetic stage and label them accordingly.

Query:

select string\_agg(pat."Patient\_ID",', ') PreDiabeticPatients, count(pat."Patient\_ID"), dgroup."Group"

from

public."Patients" Pat

join public."Lab\_Test" labt

on pat."Patient\_ID"=labt."Patient\_ID"

left join public."Group" dgroup

on pat."Group\_ID"=dgroup."Group\_ID"

where labt."Hb\_A1C" between 5.69 and 6.49

group by 3;

A screenshot of a computer

Description automatically generated with medium confidence

79.Calculate the Patient's Daytime MAP and Nighttime MAP.

Query:

select pat."Firstname"||' '||pat."Lastname" PatientName, ("24Hr\_Day\_DBP")+("24Hr\_Day\_SBP"-"24Hr\_Day\_DBP")/3 DaytimeMAP,

("24Hr\_Night\_DBP")+("24Hr\_Night\_SBP"-"24Hr\_Night\_DBP")/3 NighttimeMAP

from public."Patients" pat

join public."Blood\_Pressure" bp

on bp."Patient\_ID"=pat."Patient\_ID";

A screenshot of a computer

Description automatically generated with low confidence

80.Write a query using recursive view.

Query:

CREATE OR REPLACE RECURSIVE VIEW RunningTotal(n, totals) AS

(

SELECT 1 as n, 9 as totals

union all

SELECT n+1, totals+n from RunningTotal where n<9

);

select \* from RunningTotal;

A picture containing text, screenshot

Description automatically generated