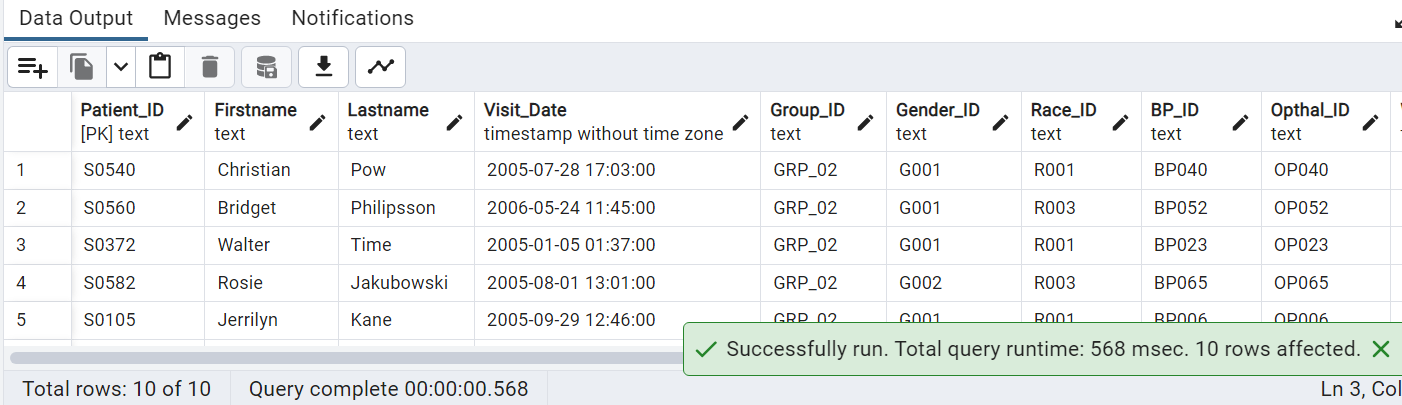
1. Display any 10 random DM patients.

select \* from "Patients"

where "Group\_ID" in(select "Group\_ID" from "Group" where "Group"='DM')

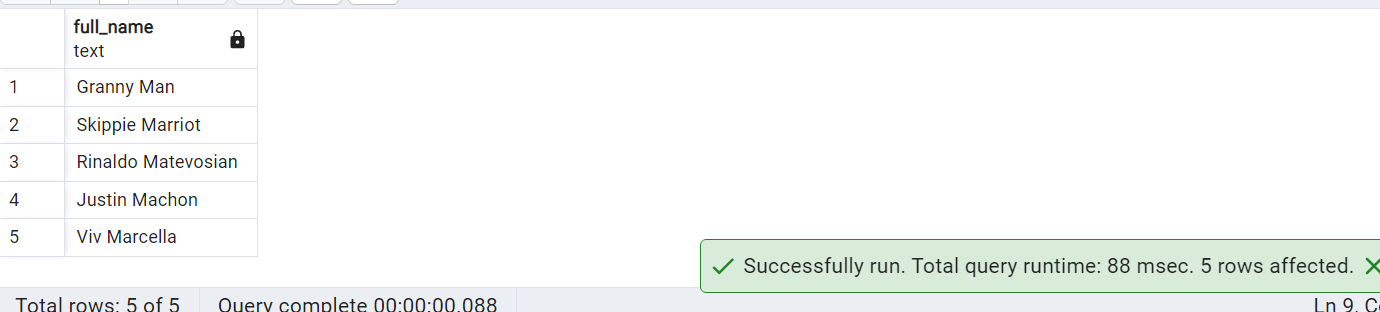
order by random() limit 10;



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

select CONCAT("Firstname", ' ', "Lastname") AS "full\_name" from "Patients"

where "Lastname" like 'Ma%';



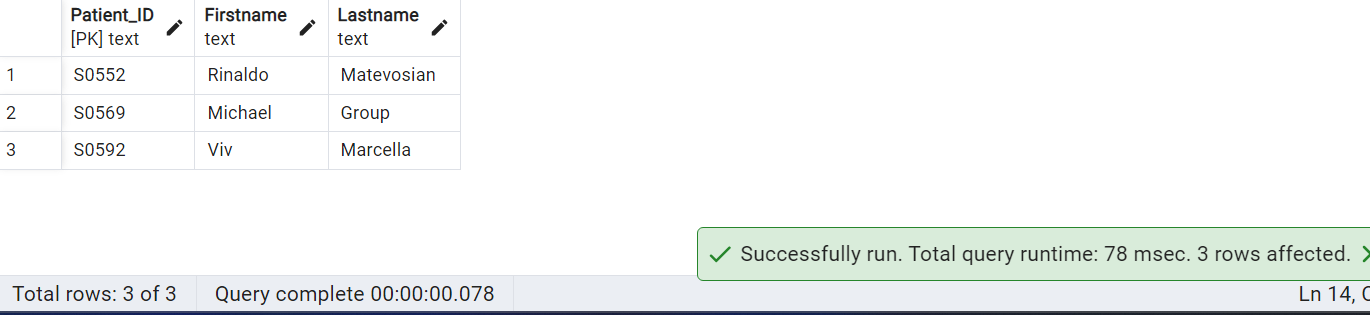
3. Write a query to get a list of patients whose RPE start is at moderate intensity.

Select p."Patient\_ID",p."Firstname",p."Lastname"

from "Patients" p

join "Walking\_Test" wt on p."WalkTest\_ID"=wt."WalkTest\_ID"

where wt."Gait\_RPE\_Start " Between 4 and 6;



4. Write a query by using common table expressions and case statements to display birth year ranges.

with cts as

(

select

"Patient\_ID",

"Firstname",

"Lastname",

"Age",

EXTRACT(YEAR FROM CURRENT\_DATE) - "Age" AS "birth\_year"

from

"Patients"

)

select

"Patient\_ID",

"Firstname",

"Lastname",

CASE

When

birth\_year IS NOT NULL

THEN

CONCAT(FLOOR(birth\_year / 10)\*10, '-', FLOOR(birth\_year / 10)\*10 + 9)

ELSE

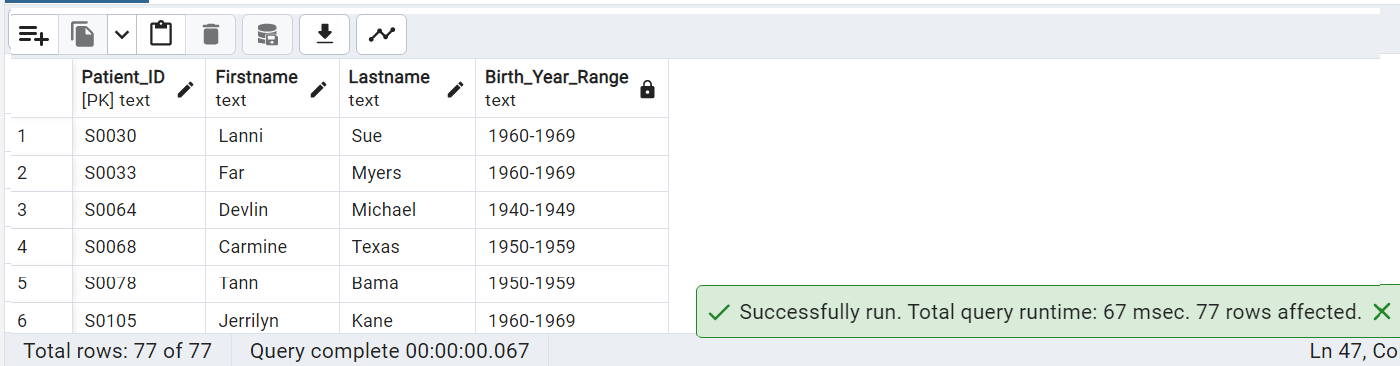
'unknown'

END

AS "Birth\_Year\_Range"

From

cts;



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

CREATE INDEX idx\_patients ON public."Patients" ("Patient\_ID", "Group\_ID");

CREATE INDEX idx\_blood\_pressure ON public."Blood\_Pressure" ("Patient\_ID");

CREATE INDEX idx\_group ON public."Group" ("Group");

WITH dm\_patients AS (

SELECT

P."Firstname",

P."Lastname",

((2 \* BP."24Hr\_Day\_DBP") + BP."24Hr\_Day\_SBP") / 3 AS day\_map,

((2 \* BP."24Hr\_Night\_DBP") + BP."24Hr\_Night\_SBP") / 3 AS night\_map,

ROW\_NUMBER() OVER (ORDER BY ((2 \* BP."24Hr\_Day\_DBP") + BP."24Hr\_Day\_SBP") / 3 DESC) AS rn

FROM public."Patients" P

JOIN public."Blood\_Pressure" BP ON BP."Patient\_ID" = P."Patient\_ID"

JOIN public."Group" G ON G."Group\_ID" = P."Group\_ID"

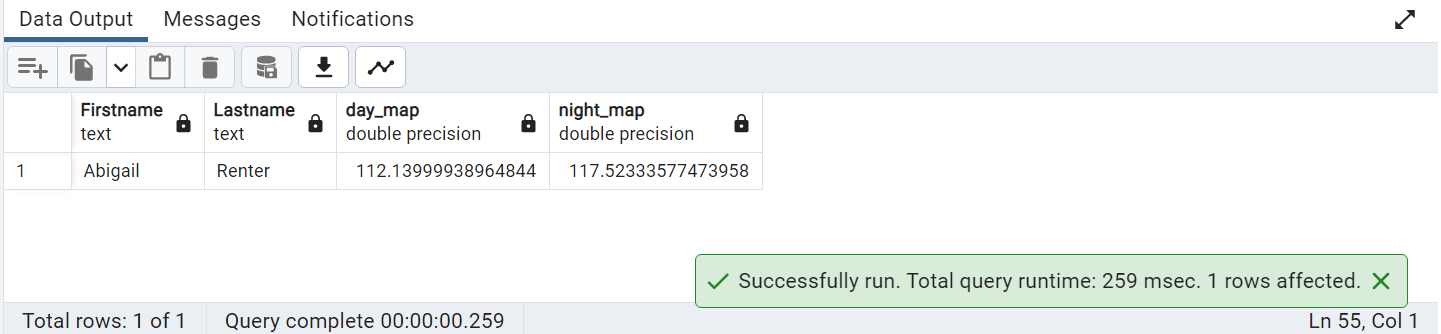
WHERE G."Group" LIKE '%DM%'

)

SELECT "Firstname", "Lastname", day\_map, night\_map

FROM dm\_patients

WHERE rn = 1;



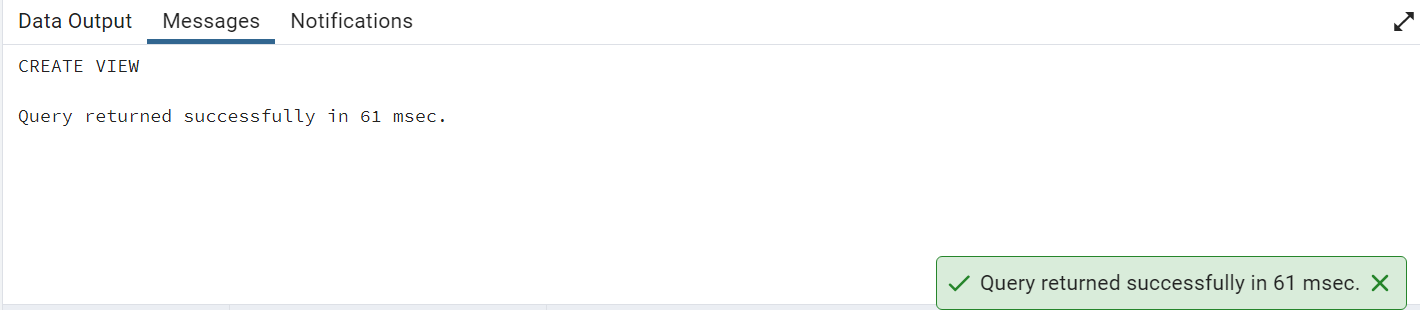
6. Create view on table Lab Test by selecting some columns and filter data using Where condition.

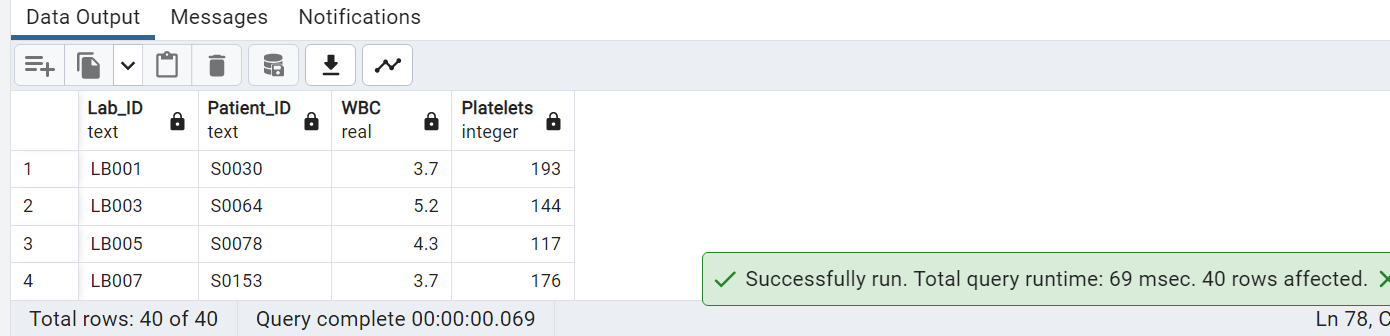
create or replace view bc\_test\_result as

select "Lab\_ID","Patient\_ID","WBC","Platelets" from "Lab\_Test"

where "WBC" between 3 and 6;

select \* from bc\_test\_result;

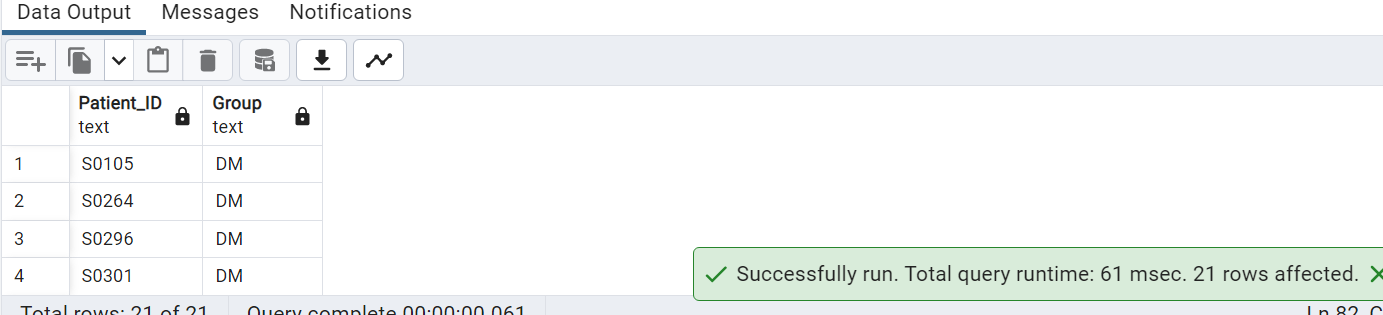




7. Display a list of Patient IDs and their Group whose diabetes duration is greater than 10 years.

select "Patient\_ID","Group" from "Patients" join "Group" on "Patients"."Group\_ID"= "Group"."Group\_ID"

where "Diabetes\_Duration" > 10;



8. 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.

select p."Patient\_ID","Firstname","Lastname","Age",g."Gender",bp."24Hr\_Day\_SBP",bp."24Hr\_Day\_DBP" from "Patients" as p

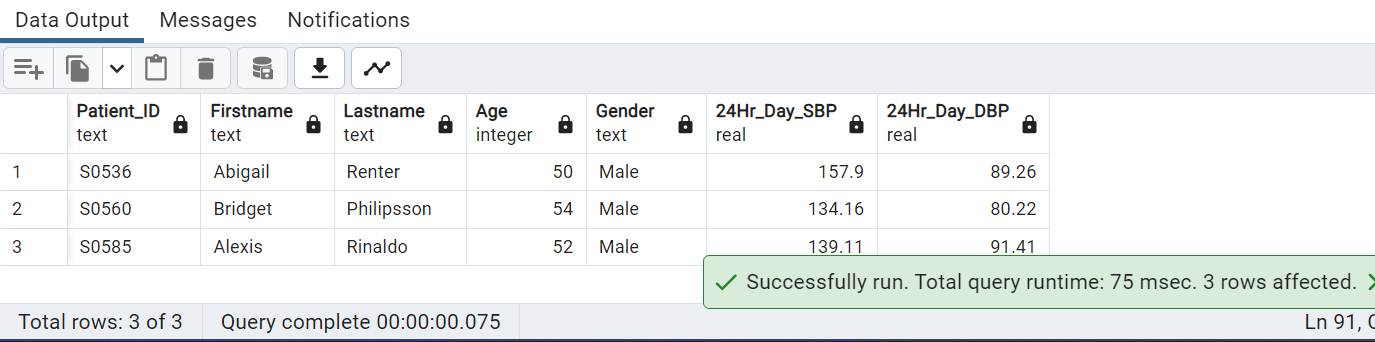
join "Gender" as g on p."Gender\_ID"=g."Gender\_ID"

join "Blood\_Pressure" as bp on bp."BP\_ID"=p."BP\_ID"

where g."Gender"='Male'

and p."Age" between 40 and 60

and bp."24Hr\_Day\_SBP" > 120 and bp."24Hr\_Day\_DBP" >80;



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

CREATE OR REPLACE FUNCTION calculate\_lab\_visit\_percentage()

RETURNS TABLE (month\_name text, year integer, percentage numeric)

AS $$

DECLARE

total\_visits bigint;

BEGIN

SELECT EXTRACT(YEAR FROM current\_date) AS year, COUNT(DISTINCT "Patient\_ID") AS total\_visits

INTO year, total\_visits

FROM public."Patients"

GROUP BY year;

FOR month\_num IN 1..12

LOOP

SELECT TO\_CHAR(DATE\_TRUNC('MONTH', current\_date) + (month\_num - 1) \* INTERVAL '1 MONTH', 'Month') AS month\_name, year,

(COUNT(DISTINCT "Patient\_ID") \* 100) / total\_visits

INTO month\_name, year, percentage

FROM public."Patients"

WHERE EXTRACT(MONTH FROM "Visit\_Date") = month\_num

GROUP BY month\_name, year

ORDER BY EXTRACT(MONTH FROM DATE\_TRUNC('MONTH', current\_date) + (month\_num - 1) \* INTERVAL '1 MONTH');

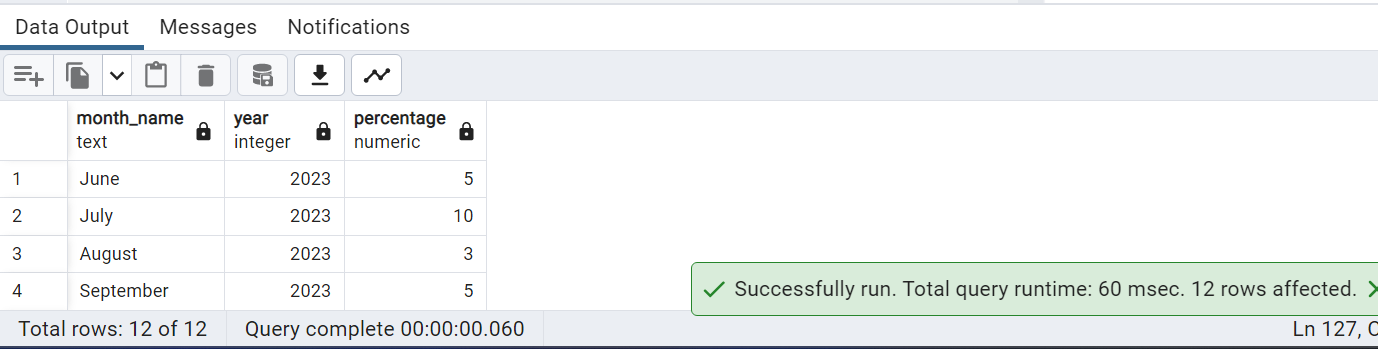
RETURN NEXT;

END LOOP;

END;

$$ LANGUAGE plpgsql;

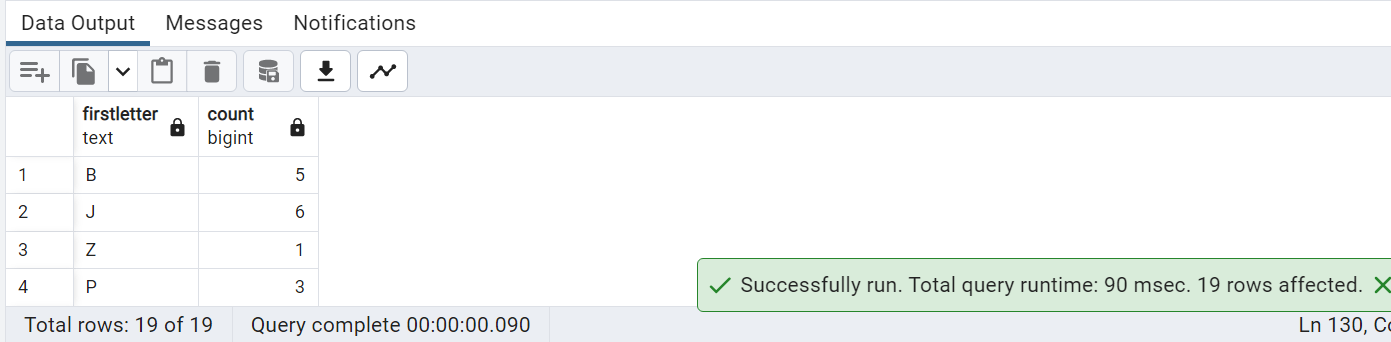
select \* from calculate\_lab\_visit\_percentage();



10. Count of patients by first letter of firstname.

select left("Firstname",1) as firstletter, count(1) from "Patients"

group by firstletter;



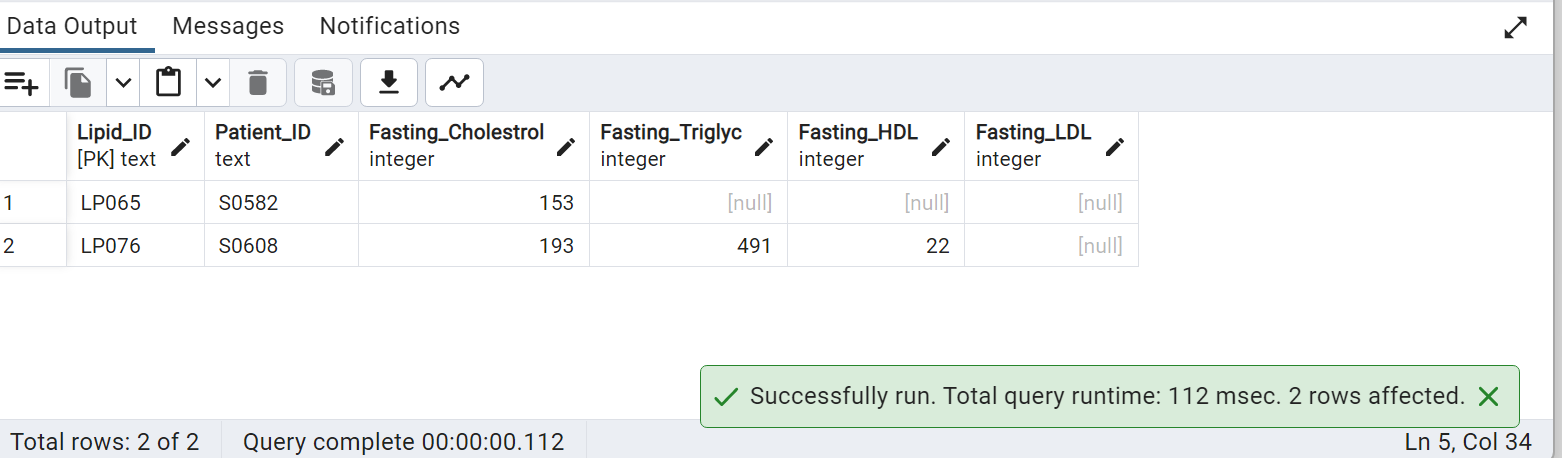
11) write a query to get the list of patients whose lipid test value is null

SELECT \* FROM "Lipid\_Lab\_Test" where "Fasting\_Cholestrol" IS NULL

OR "Fasting\_Triglyc" IS NULL

OR "Fasting\_HDL" IS NULL

OR "Fasting\_LDL" IS NULL;



12)Create a stored procedure to make user ids for the given patient id

create or replace procedure GenerateUserID(patient\_id int)

language plpgsql

as $$

Declare

userID varchar(20);

usernumber Integer;

Begin

--usernumber is calculated by multiplying by 1000

usernumber:=patient\_id\*1000;

--concatenate 'UID' with the converted userNumber to VARCHAR and assigns it to userId.

userId:='UID' || usernumber::VARCHAR;

--raise a notice message with generated userID

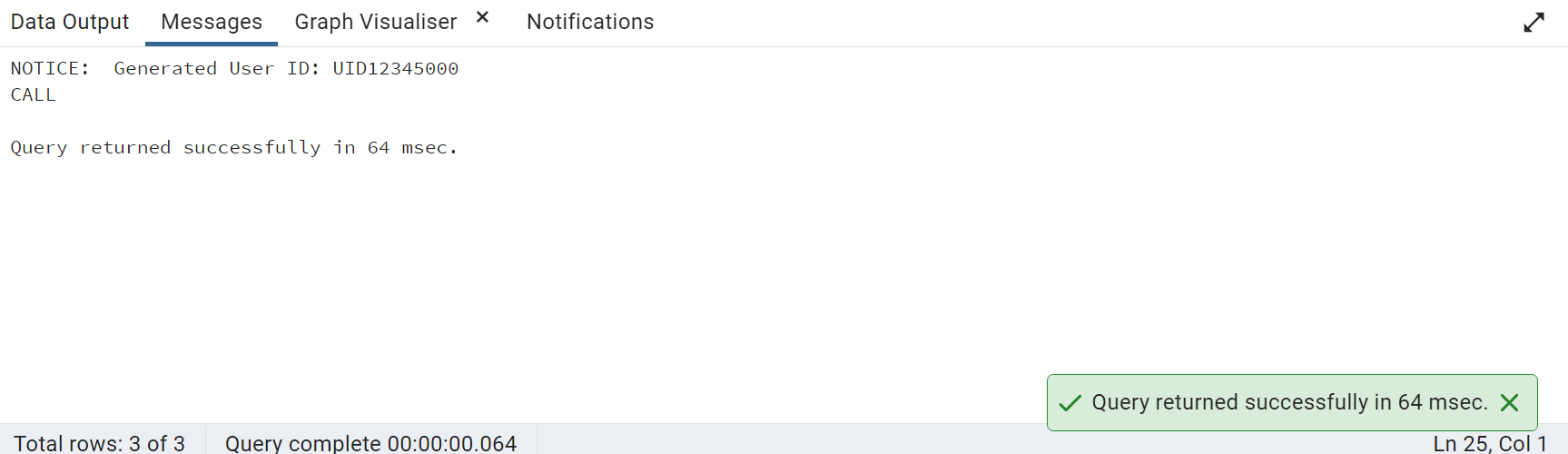
RAISE NOTICE 'Generated User ID: %' , userID;

End;

$$;

--Call procedure by passing PatientID as argument

call GenerateUserID(12345);

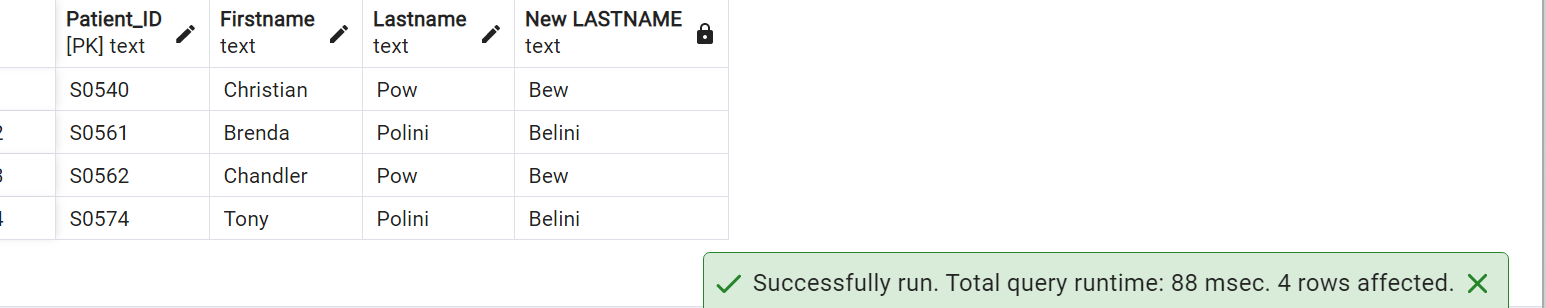


13)Display Patients With LastName contains 'Po' and replace it with 'Be'.

SELECT "Patient\_ID","Firstname","Lastname",REPLACE("Lastname",'Po','Be') As "New LASTNAME"

FROM public."Patients"

WHERE "Lastname" LIKE 'Po%';



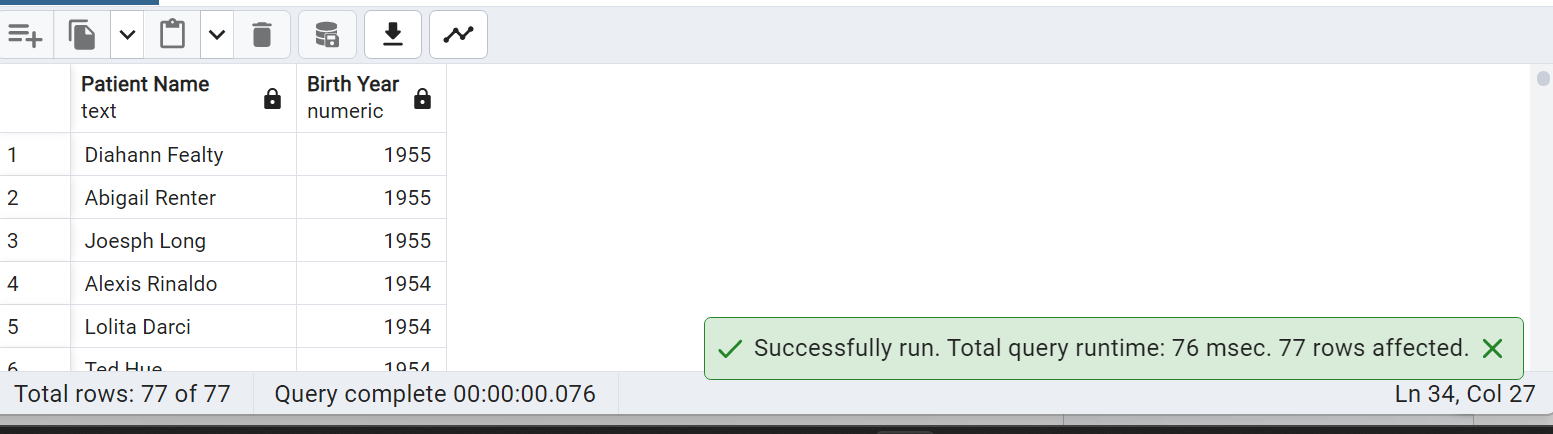
14)Calculate the patient's birth year in descending order

SELECT CONCAT("Firstname",' ',"Lastname") As "Patient Name",

(Extract(year from "Visit\_Date"))-"Age" as "Birth Year"

FROM public."Patients"

ORDER BY "Birth Year" DESC;



15) Find the patients that have eye damage due to diabetes.

SELECT CONCAT(p."Firstname", ' ', p."Lastname") AS "Patients With Eye Damage"

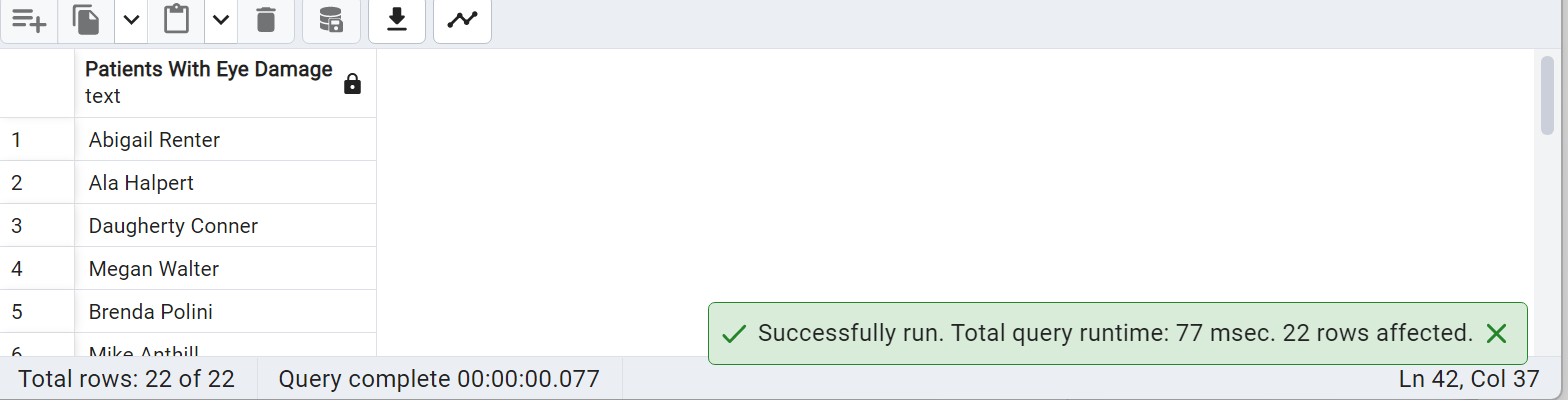
FROM public."Patients" p

JOIN public."Opthalmology" o ON p."Opthal\_ID" = o."Opthal\_ID"

-- Filter the rows where either "Diabetic\_Retinopathy" or "Macular\_Edema" > 0

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

GROUP BY "Patients With Eye Damage";



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

SELECT "Patient\_ID",

CASE WHEN "Gait\_RPE\_End "= 0 THEN 'Rest'

WHEN "Gait\_RPE\_End " BETWEEN 1 AND 3 THEN 'EASY INTENSITY'

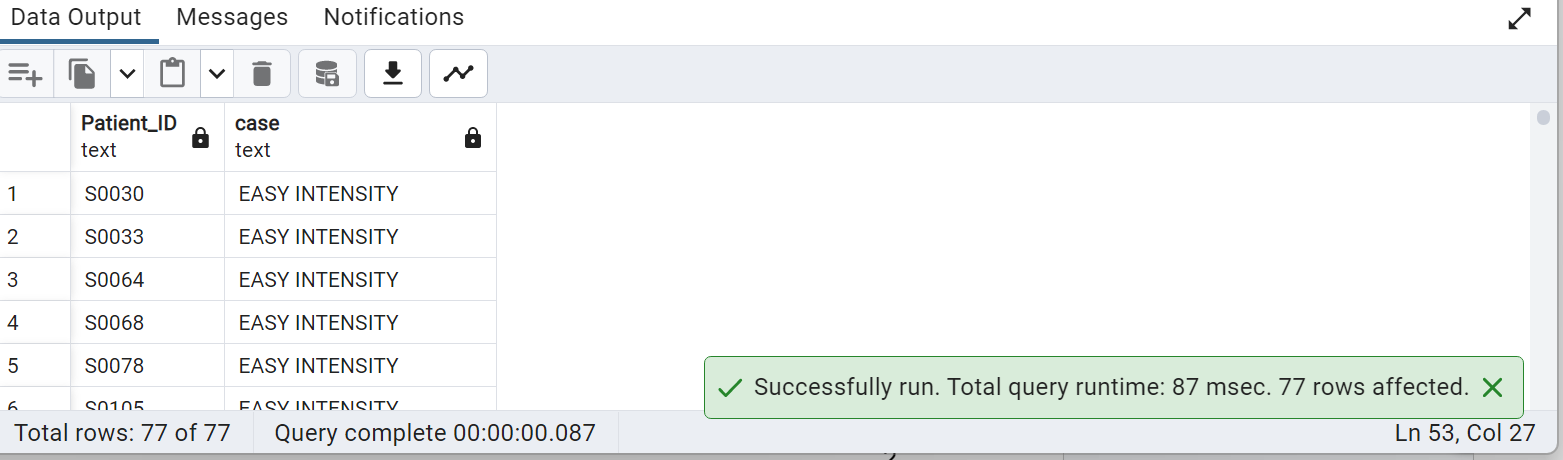
WHEN "Gait\_RPE\_End " BETWEEN 4 AND 6 THEN 'MODERATE INTENSITY'

WHEN "Gait\_RPE\_End " BETWEEN 7 AND 9 THEN 'HARD INTENSITY'

WHEN "Gait\_RPE\_End " = 10 THEN 'MAX EFFORT INTENSITY'

END

FROM public."Walking\_Test";

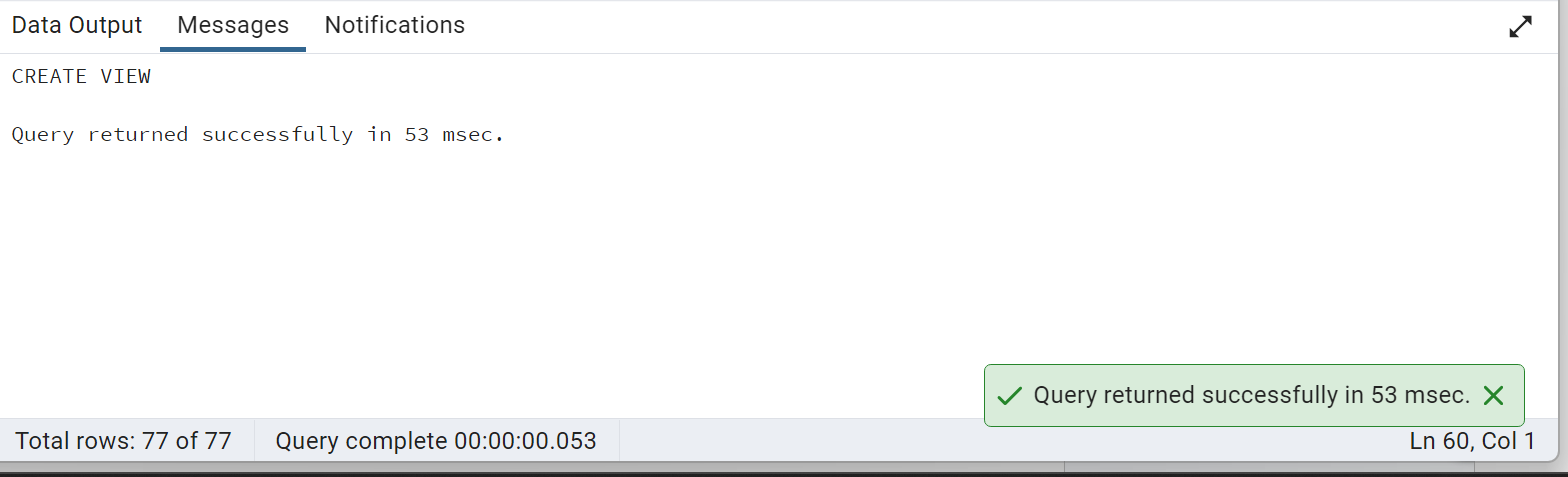


17) Create view on patient table with check constraint condition

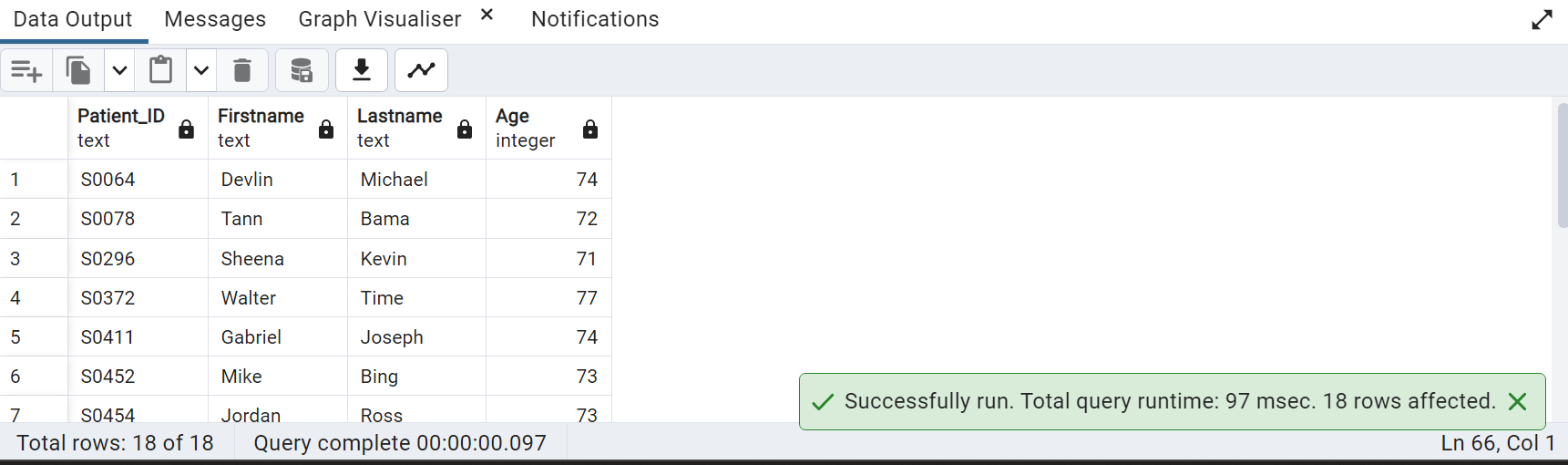
CREATE VIEW Vw\_Patient\_Senior as

SELECT "Patient\_ID","Firstname", "Lastname","Age" FROM public."Patients"

WHERE "Age">70;



SELECT \* FROM Vw\_Patient\_Senior;

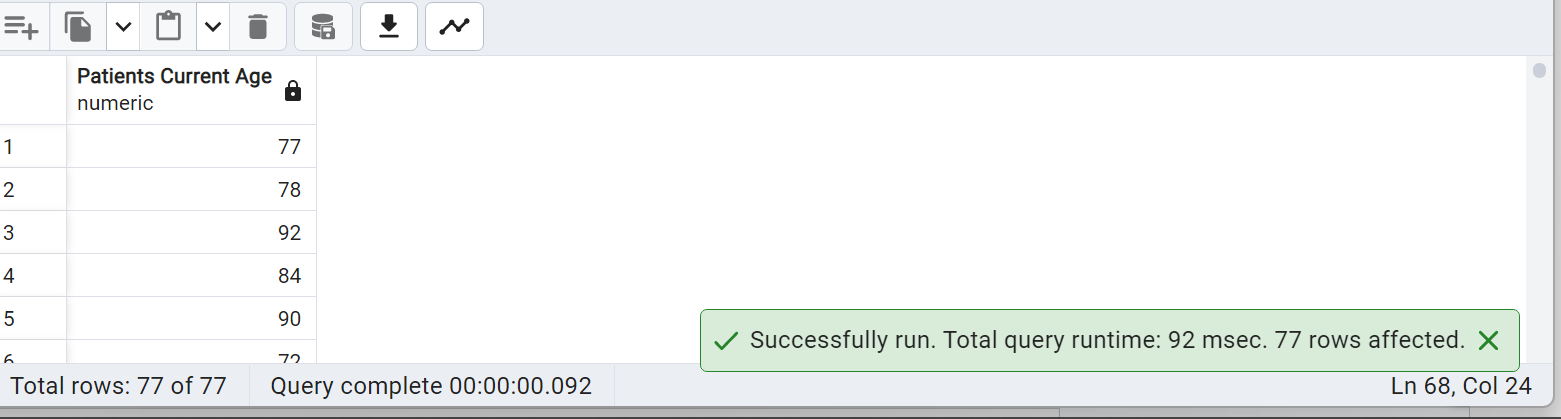


18) Calculate the patient's current age

-- Subtracting Visit\_Date from current year and adding Age at Visit time gives patients' Current Age

SELECT (Extract(year from now()))-(Extract(year from "Visit\_Date"))+"Age" as "Patients Current Age"

FROM public."Patients";



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

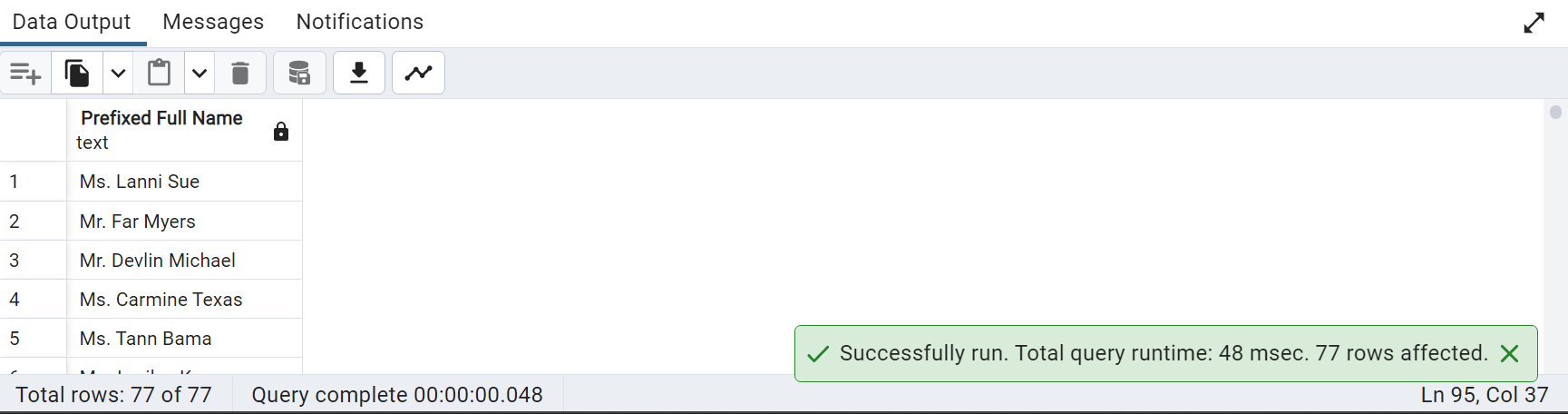
SELECT CASE WHEN G."Gender"='Male' THEN CONCAT('Mr. ', P."Firstname",' ',P."Lastname")

WHEN G."Gender"='Female' THEN CONCAT('Ms. ', P."Firstname",' ', P."Lastname") END

As " Prefixed Full Name"

FROM public."Patients" P, public."Gender" G

WHERE G."Gender\_ID" = P."Gender\_ID";



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

SELECT CONCAT(p."Firstname", ' ', p."Lastname") AS "Patient", wt."Gait\_DT\_Distance", wt."Gait\_DT\_Speed"

FROM public."Patients" p

JOIN public."Walking\_Test" wt ON p."Patient\_ID"=wt."Patient\_ID"

WHERE "Gait\_DT\_Distance">400 AND "Gait\_DT\_Speed" > 1;

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

CREATE VIEW PatientsView

AS

SELECT pnt."Patient\_ID",pnt."Firstname", pnt."Lastname", pnt."Visit\_Date",pnt."Age", pnt."Height", pnt."BMI",

grp."Group",gnr."Gender",race."Race",bp."24Hr\_Day\_SBP",bp."24Hr\_Day\_DBP",bp."24Hr\_Day\_HR",

opt."Diabetic\_Retinopathy", opt."Macular\_Edema", lr."Lipid\_ID", lr."Lab\_ID"

FROM public."Patients" AS pnt

INNER JOIN public."Group" AS grp ON pnt."Group\_ID" = grp."Group\_ID"

INNER JOIN public."Gender" AS gnr ON pnt."Gender\_ID" = gnr."Gender\_ID"

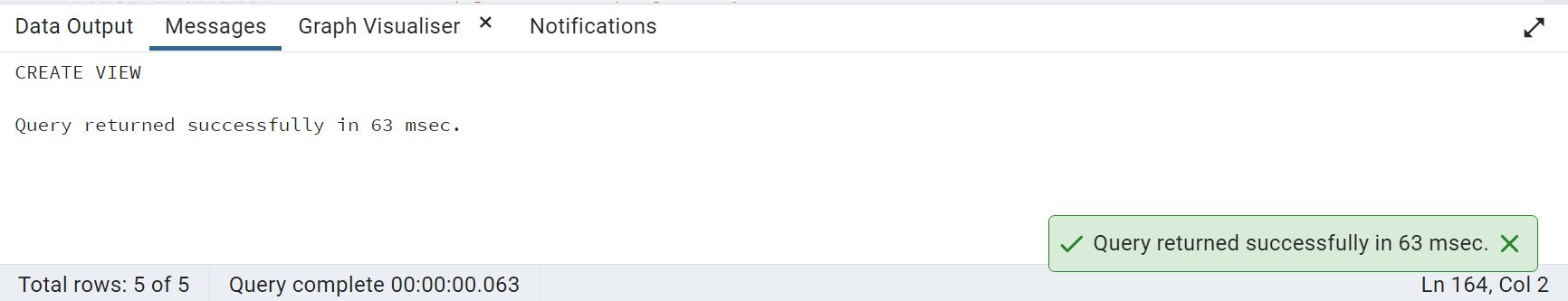
INNER JOIN public."Race" AS race ON pnt."Race\_ID" = race."Race\_ID"

INNER JOIN public."Blood\_Pressure" AS bp ON pnt."BP\_ID" = bp."BP\_ID"

INNER JOIN public."Opthalmology" AS opt ON pnt."Opthal\_ID" = opt."Opthal\_ID"

INNER JOIN public."Link\_Reference" AS lr ON pnt."Link\_Reference\_ID" = lr."Link\_Reference\_ID"

;



CREATE FUNCTION patient\_prevent\_delete()

RETURNS TRIGGER

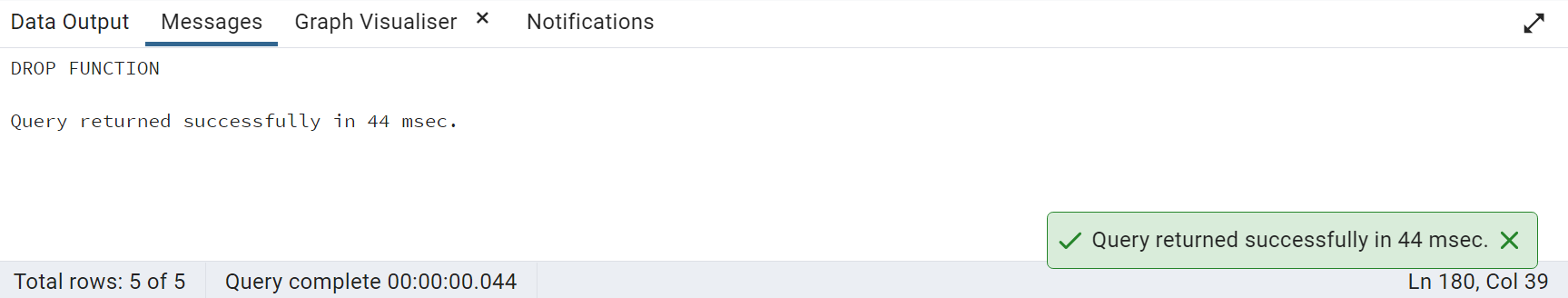
AS $$

BEGIN

RAISE EXCEPTION 'You cannot delete records from View';

END;

$$



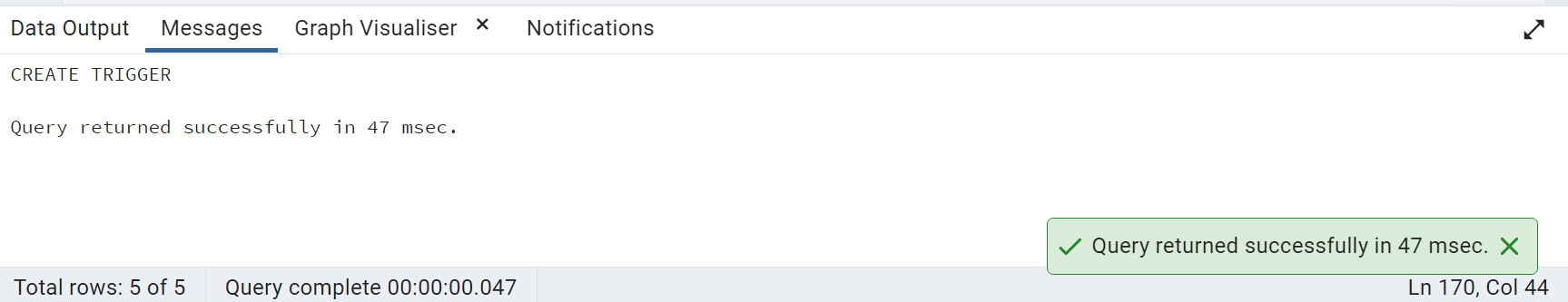
CREATE TRIGGER TG\_PATIENT\_PREVENT\_DELETE

INSTEAD OF DELETE

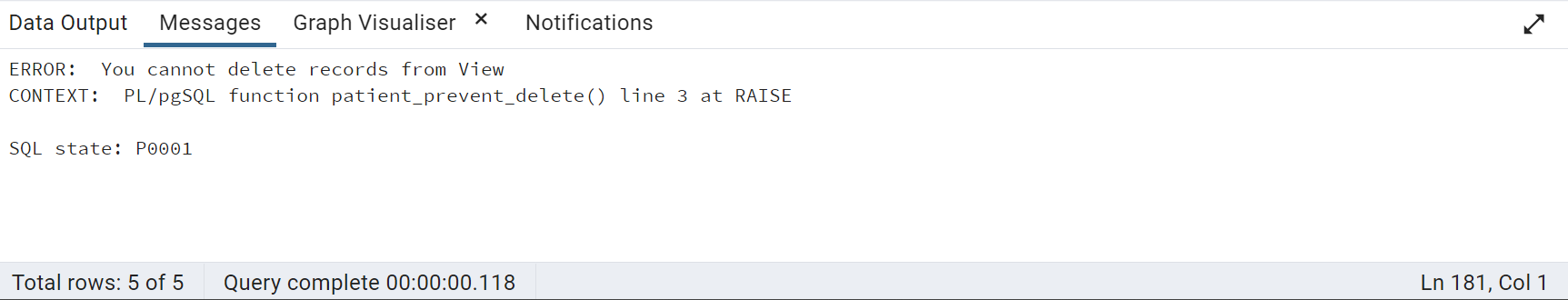
ON PatientsView

FOR EACH ROW

EXECUTE PROCEDURE patient\_prevent\_delete();



DELETE FROM patientsview WHERE "Patient\_ID"='S0030';



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.

SELECT CONCAT("Firstname",

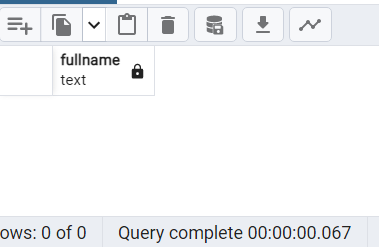
"Lastname") as "fullname"

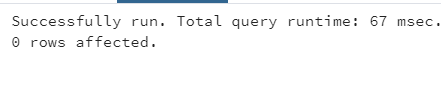
FROM "Patients"

WHERE CONCAT("Firstname",

"Lastname") iLIKE 's%r%b%';

--TO MATCH CASE INSENSITIVE CASES WE USE "ILIKE" IN POSTGRES NOT STANDARD TO SQL BUT AS POSTGRES EXTENSION"





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

SELECT "Race"

FROM "Race"

WHERE "Race\_ID" =

(SELECT "Race\_ID"

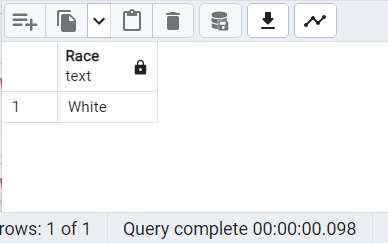
FROM "Patients"

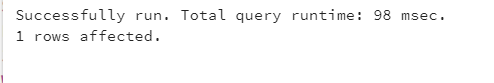
WHERE "Diabetes\_Duration" != 0

GROUP BY "Race\_ID"

ORDER BY COUNT("Race\_ID") DESC

LIMIT 1);





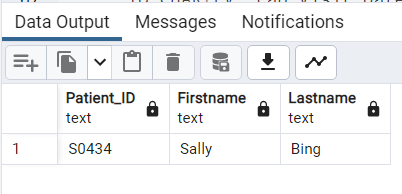
--24.Find the list of Patients who has leukopenia.

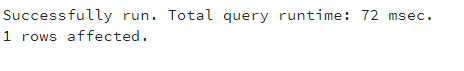
SELECT LT."Patient\_ID",P."Firstname",P."Lastname"

FROM public."Lab\_Test" LT

JOIN public."Patients" P ON P."Patient\_ID" = LT."Patient\_ID"

WHERE "WBC"<3;





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

SELECT "Lab\_names",

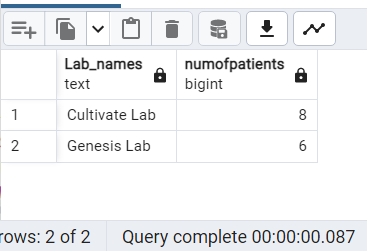
COUNT ("Lab\_visit\_ID")NUMOFPATIENTS

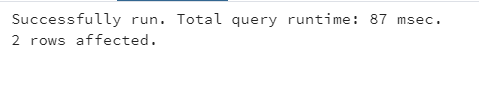
FROM PUBLIC."Lab\_Visit"

WHERE "Lab\_names" in ('Cultivate Lab','Genesis Lab')

AND (EXTRACT(YEARFROM "Lab\_Visit\_Date")) = 2005

GROUP BY "Lab\_names";





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

SELECT LLT."Patient\_ID",

LLT."Fasting\_Cholestrol",

TO\_CHAR(LV."Lab\_Visit\_Date",'MONTH')AS MONTH,

TO\_CHAR(LV."Lab\_Visit\_Date",'YYYY')AS YEAR

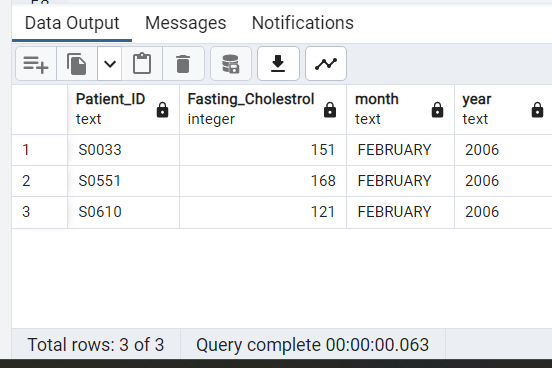
FROM PUBLIC."Lipid\_Lab\_Test" LLT

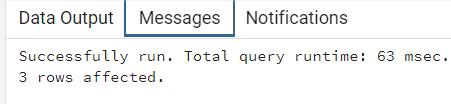
LEFT JOIN PUBLIC."Link\_Reference" LR ON LR."Lipid\_ID" = LLT."Lipid\_ID"

JOIN PUBLIC."Lab\_Visit" LV ON LV."Lab\_visit\_ID" = LR."Lab\_visit\_ID"

WHERE EXTRACT (YEAR FROM LV."Lab\_Visit\_Date") = 2006

AND EXTRACT(MONTH FROM LV."Lab\_Visit\_Date") = 2;



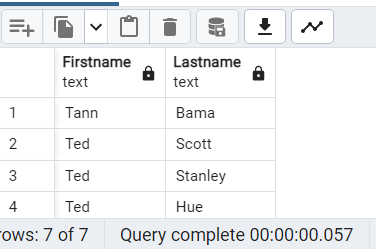


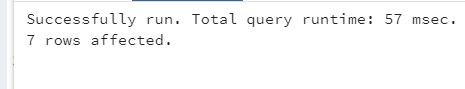
27.Write a query to get a list of patients whose first names is starting with the letter T

SELECT "Firstname"

FROM PUBLIC."Patients"

WHERE "Firstname" ILIKE '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 1.5 M.

SELECT P."Firstname",

P."Lastname"

FROM PUBLIC."Patients" P

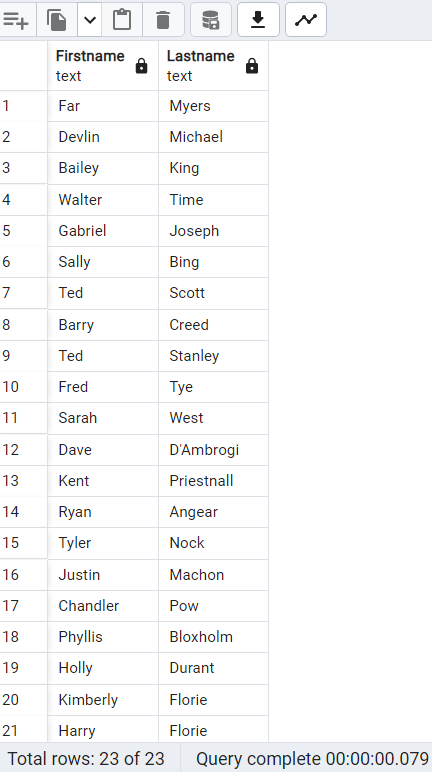
LEFT JOIN PUBLIC."Gender" G ON G."Gender\_ID" = P."Gender\_ID"

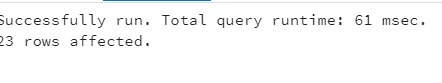
WHERE G."Gender" = 'Male'

AND P."Age" > 60

AND P."BMI" > 18.5

AND P."Height" > 1.5;





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

SELECT CEILING(UT."Creatinine") creatininelevel

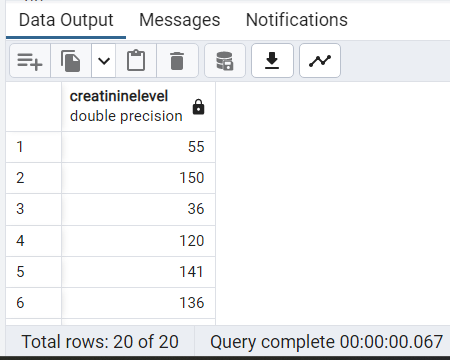
FROM PUBLIC."Urine\_Test" UT

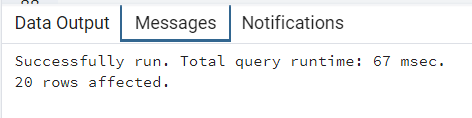
JOIN PUBLIC."Link\_Reference" LR ON LR."Urine\_ID" = UT."Urine\_ID"

JOIN PUBLIC."Patients" PT ON PT."Link\_Reference\_ID" = LR."Link\_Reference\_ID"

JOIN public."Gender" G ON PT."Gender\_ID" = G."Gender\_ID"

WHERE "Age" > 35 AND "Creatinine" NOT BETWEEN 65.4 AND 119.3 AND "Gender" = 'Male'



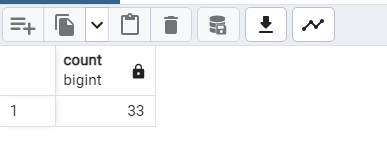


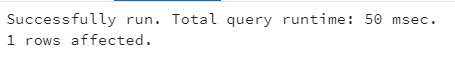
30.Write a query to get the number of patients who visited the Lab between 9 am to 12 am.

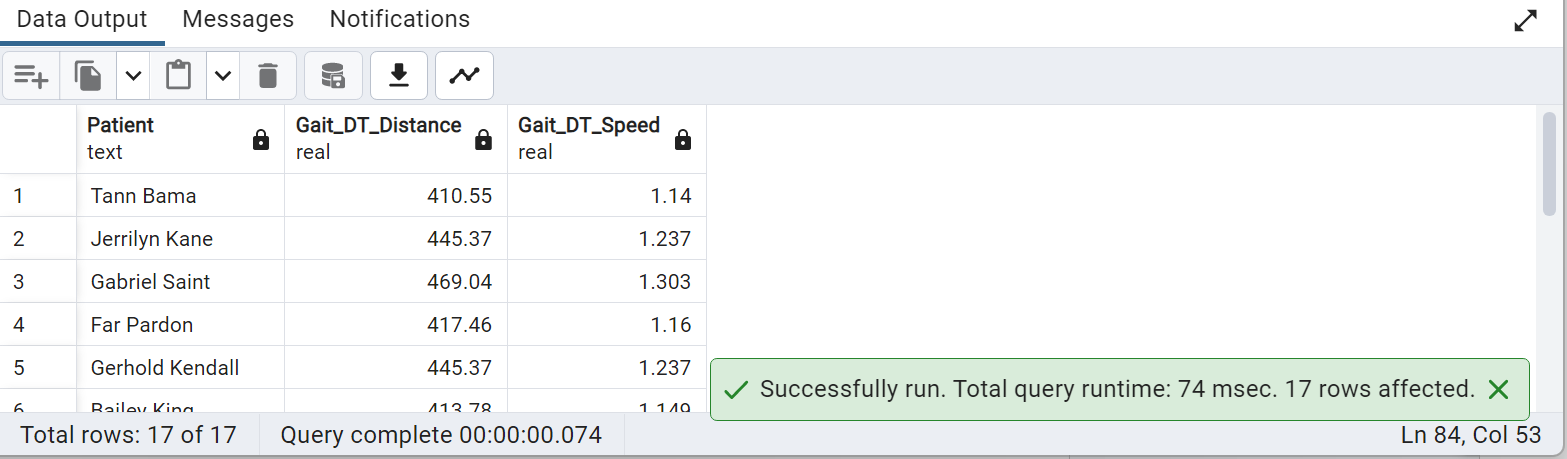
SELECT COUNT("Lab\_visit\_ID")

FROM PUBLIC."Lab\_Visit"

WHERE EXTRACT (HOUR FROM "Lab\_Visit\_Date") BETWEEN 9 AND 12;







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.

*-- Function to check for space and case in column1 and column2*

CREATE FUNCTION check\_space\_and\_case() RETURNS TRIGGER AS $$

DECLARE

column1\_value text;

column2\_value text;

BEGIN

*-- Assign the values of column1 and column2 from the new row being inserted or updated*

column1\_value := NEW.column1;

column2\_value := NEW.column2;

*-- Check if column1 or column2 contains a space*

IF column1\_value ~ '\s' OR column2\_value ~ '\s' THEN

*-- Raise an exception if a space is found in column1 or column2*

RAISE EXCEPTION 'Space is not allowed in column1 or column2';

END IF;

*-- Check if column1 or column2 is not in lowercase*

IF column1\_value <> lower(column1\_value) OR column2\_value <> lower(column2\_value) THEN

*-- Raise an exception if column1 or column2 is not in lowercase*

RAISE EXCEPTION 'Column1 or column2 must be in lowercase';

END IF;

*-- If the checks pass, return the new row*

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

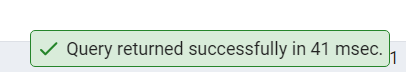
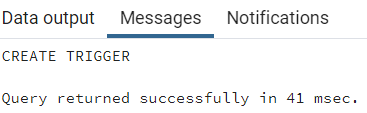
*-- Trigger to enforce the space and case check before inserting or updating rows in the* "Patients" table

CREATE TRIGGER space\_case\_check\_trigger

BEFORE INSERT OR UPDATE ON public."Patients"

FOR EACH ROW

EXECUTE FUNCTION check\_space\_and\_case();



32) Write a query to calculate the running moving averages of diabetes\_duration for Group 2 using the moving windows/sliding dynamic average windows.

SELECT "Patient\_ID", "Firstname", "Lastname","Diabetes\_Duration",

*-- Calculate the moving average of diabetes duration within each group*

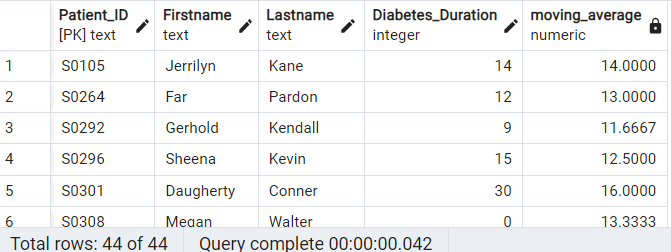
AVG("Diabetes\_Duration") OVER (

PARTITION BY "Group\_ID" ORDER BY "Patient\_ID" ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW

) AS moving\_average

FROM "Patients"

WHERE "Group\_ID" = 'GRP\_02';





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

CREATE TABLE public."Patient\_Detail" AS

SELECT a."Patient\_ID", a."Firstname", a."Lastname", g."Gender", c."Race", a."Age", a."Height", a."BMI",

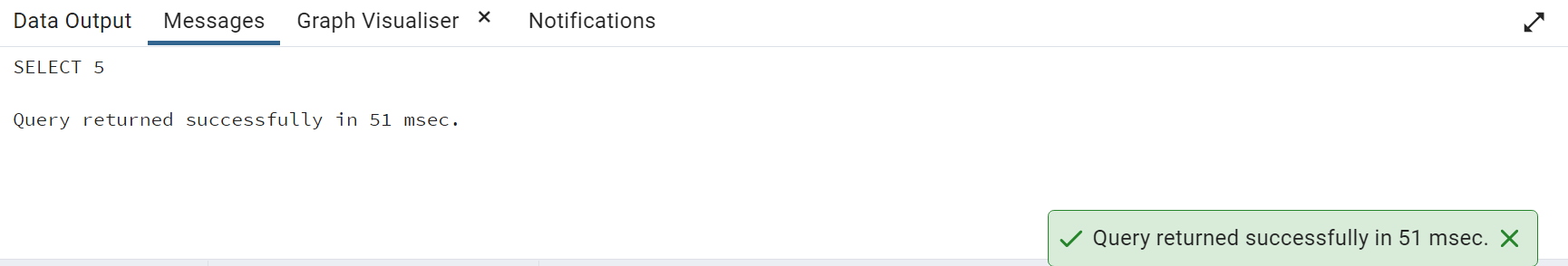
(extract(year from a."Visit\_Date")) - a."Age" as Birth\_year

FROM public."Patients" a

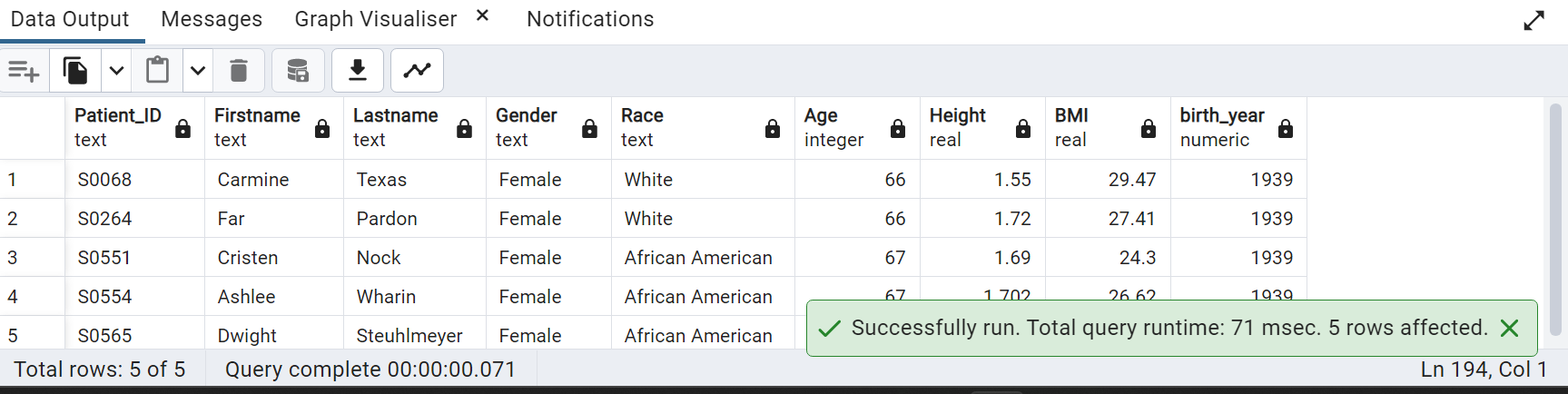
JOIN public."Gender" g ON a."Gender\_ID"=g."Gender\_ID"

JOIN public."Race" c ON a."Race\_ID"=c."Race\_ID"

WHERE ((extract(year from a."Visit\_Date")) - a."Age")='1939';



SELECT \* FROM public."Patient\_Detail"



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

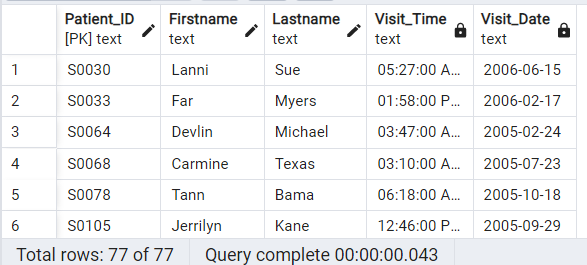
SELECT "Patient\_ID","Firstname","Lastname",

*-- Format the visit time using the TO\_CHAR() function*

TO\_CHAR("Visit\_Date", 'HH:MI:SS AM') AS "Visit\_Time",

TO\_CHAR("Visit\_Date", 'YYYY-MM-DD') AS "Visit\_Date"

FROM public."Patients";





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

SELECT TO\_CHAR("Visit\_Date", 'Month') AS "Month\_Name",COUNT(\*) AS "Number\_of\_Patients"

FROM "Patients"

*-- Group the data by the month extracted from the "Visit\_Date" column*

GROUP BY TO\_CHAR("Visit\_Date", 'Month')

ORDER BY MIN("Visit\_Date");





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).

SELECT COUNT(\*) AS "Patient\_Count",

*-- Concatenate the abnormal conditions using CONCAT\_WS() function and conditional expressions*

CONCAT\_WS(', ',

CASE WHEN VC."RCFT\_IR" > 71 THEN 'RCFT' END,

CASE WHEN VC."TM" > 42 THEN 'TM' END,

CASE WHEN VC."Clock" <= 2 THEN 'Clock' END,

CASE WHEN MC."GDS" >= 15 THEN 'GDS' END

) AS "Abnormal\_Conditions"

FROM public."Visual/Motor\_Cog" VC

JOIN public."Link\_Reference" LR ON LR."VM\_ID" = VC."VM\_ID"

JOIN public."Memory\_Cognitive" MC ON LR."MC\_ID" = MC."MC\_ID"

WHERE (

*-- The CASE statements are used to conditionally include abnormal conditions based on specific criteria.*

CASE WHEN VC."RCFT\_IR" > 71 THEN 1 ELSE 0 END +

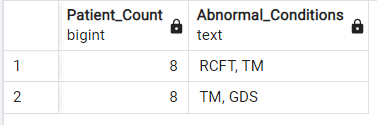
CASE WHEN VC."TM" > 42 THEN 1 ELSE 0 END +

CASE WHEN VC."Clock" <= 2 THEN 1 ELSE 0 END +

CASE WHEN MC."GDS" >= 15 THEN 1 ELSE 0 END

) >= 2

GROUP BY "Abnormal\_Conditions";





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

SELECT "Patient\_ID"

FROM public."Lab\_Test"

WHERE "Fasting\_Glucose" IN (80, 85, 89);





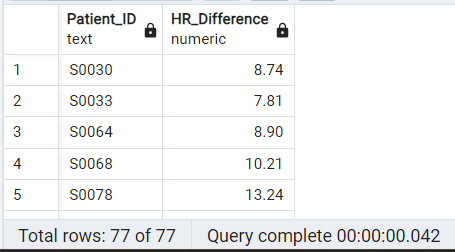
38) calculate the difference between Day and night HR. (Display 2 decimal only)

SELECT "Patient\_ID",

*-- Calculate the difference between "24Hr\_Day\_HR" and "24Hr\_Night\_HR" and round it to two decimal places, ensuring a numeric data type*

ROUND(("24Hr\_Day\_HR" - "24Hr\_Night\_HR")::numeric, 2) AS "HR\_Difference"

FROM "Blood\_Pressure";



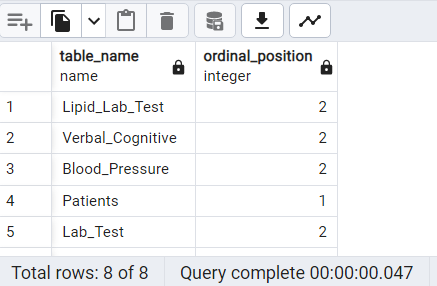


39) Find out the tables where column Patient\_ID is present.(Display column position number with respective table also)

SELECT table\_name,ordinal\_position

FROM information\_schema.columns

WHERE column\_name = 'Patient\_ID';





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

SELECT P."Patient\_ID",P."Firstname",P."Lastname",G."Group",(UT."Creatinine" / UT."Albumin") AS "Creat:Alb"

FROM public."Patients" P

JOIN "Link\_Reference" LR ON P."Link\_Reference\_ID" = LR."Link\_Reference\_ID"

JOIN "Urine\_Test" UT ON LR."Urine\_ID" = UT."Urine\_ID"

JOIN "Group" G ON P."Group\_ID" = G."Group\_ID"

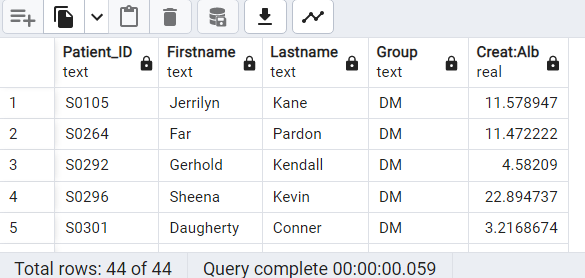
JOIN public."Patients" DM ON P."Patient\_ID" = DM."Patient\_ID"

WHERE DM."Group\_ID" IN (

SELECT "Group\_ID"

FROM "Group"

WHERE "Group" = 'DM');





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

SELECT G."Gender", COUNT(P."Patient\_ID") AS "Number of Patients"

FROM public."Patients" P

JOIN public."Gender" G ON G."Gender\_ID" = P."Gender\_ID"

JOIN public."Link\_Reference" LR ON LR."Link\_Reference\_ID" = P."Link\_Reference\_ID"

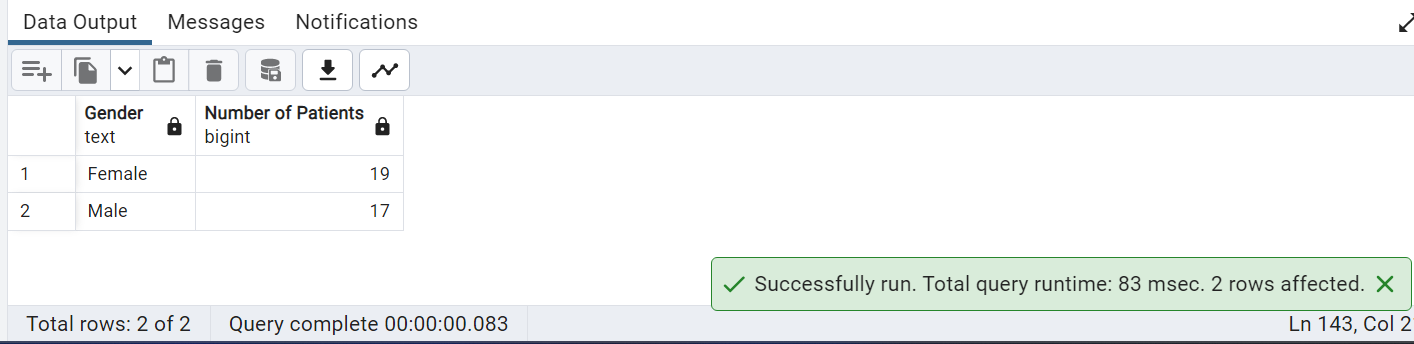
JOIN public."Urine\_Test" U ON U."Urine\_ID" = LR."Urine\_ID"

WHERE

(G."Gender" = 'Male' AND U."Creatinine" BETWEEN 65.4 AND 119.3) OR

(G."Gender" = 'Female' AND U."Creatinine" BETWEEN 52.2 AND 91.9)

GROUP BY G."Gender";



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

select \* from "Lab\_Visit"

where "Lab\_visit\_ID"='LB002'

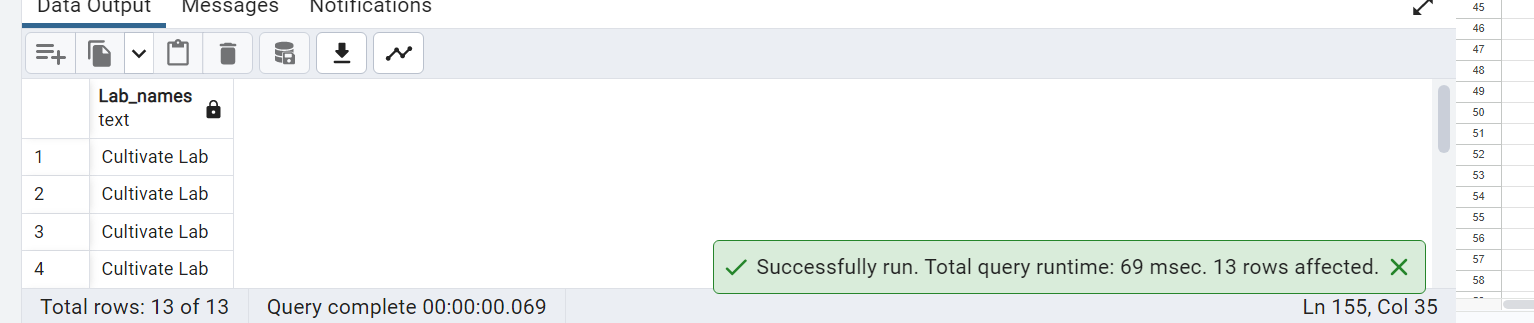
update "lab\_visit"

set "Lab\_visit\_ID"='LB002'

where "Lab\_names"='Cultivate Lab'

select "Lab\_names" from "Lab\_Visit"

where "Lab\_names"='Cultivate Lab'



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

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

create index "index\_firstname" on "Patients"

(

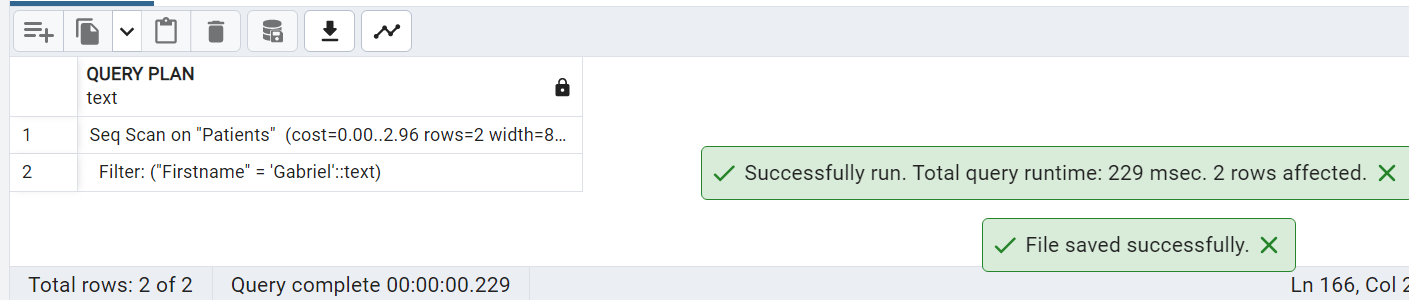
"Firstname"

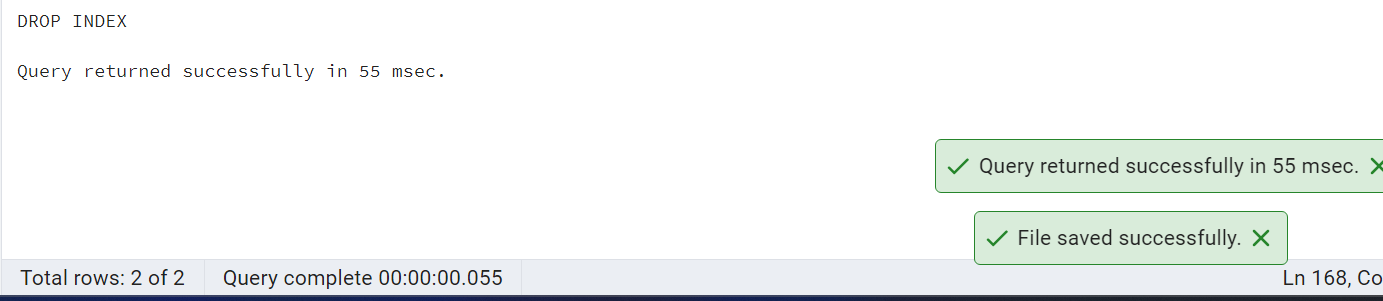
)

Explain select \* from public."Patients"

where "Firstname"= 'Gabriel'

drop index "index\_firstname"





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

select CAST ("Lab\_Visit\_Date" AS DATE) as lab\_visit\_date, CAST ("Lab\_Visit\_Date" AS TIME) as lab\_visit\_time from "Lab\_Visit"



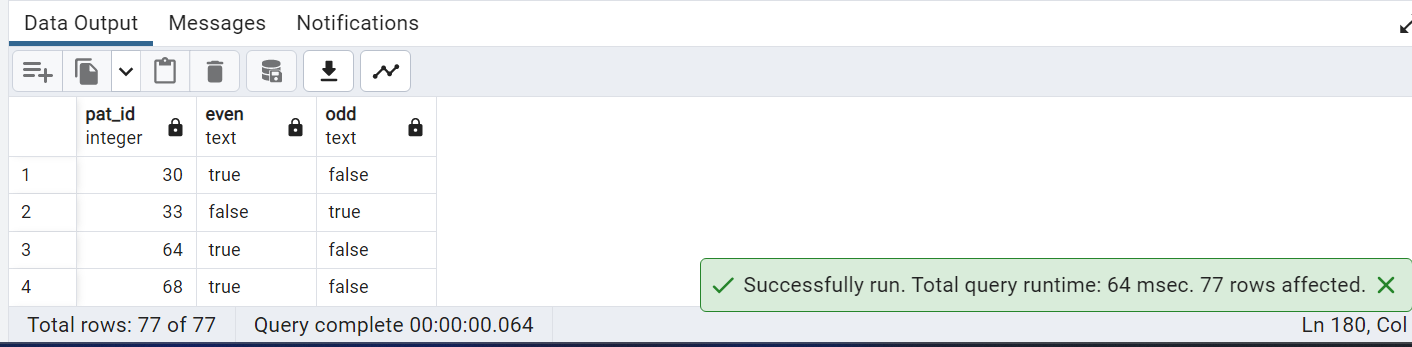
45 Please go through the below screenshot and create the exact output.

SELECT SUBSTRING("Patient\_ID" FROM 2)::INTEGER AS "pat\_id",

CASE WHEN SUBSTRING("Patient\_ID" FROM 2)::INTEGER % 2 = 0 THEN 'true' ELSE 'false' END AS "even",

CASE WHEN SUBSTRING("Patient\_ID" FROM 2)::INTEGER % 2 = 0 THEN 'false' ELSE 'true' END AS "odd"

FROM "Patients";



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

SELECT G."Gender", COUNT(DISTINCT CASE

WHEN (G."Gender" = 'Male' AND LT."Hgb" < 13.2)

OR (G."Gender" = 'Female' AND LT."Hgb" < 11.6)

THEN P."Patient\_ID"

END) AS anemic\_count

FROM public."Patients" P

JOIN public."Gender" G ON G."Gender\_ID" = P."Gender\_ID"

JOIN public."Lab\_Test" LT ON LT."Patient\_ID" = P."Patient\_ID"

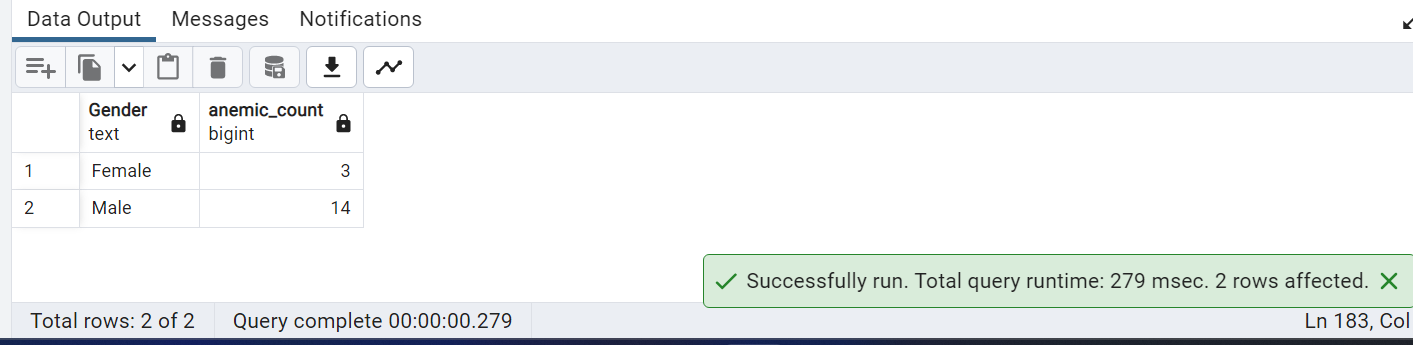
WHERE P."Patient\_ID" IN (

SELECT "Patient\_ID"

FROM public."Patients"

WHERE "Diabetes\_Duration" > 0)

GROUP BY G."Gender";

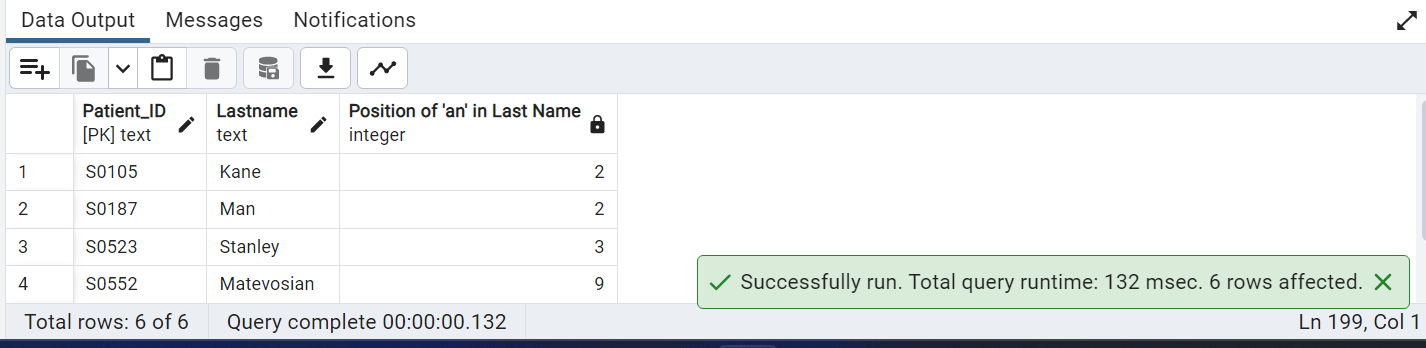


47. Write a query to display the Patient\_ID, last name, and the position of the substring 'an' in the last name column for those patients who have a substring 'an'.

SELECT "Patient\_ID", "Lastname",

POSITION ('an' IN "Lastname") AS "Position of 'an' in Last Name" FROM "Patients"

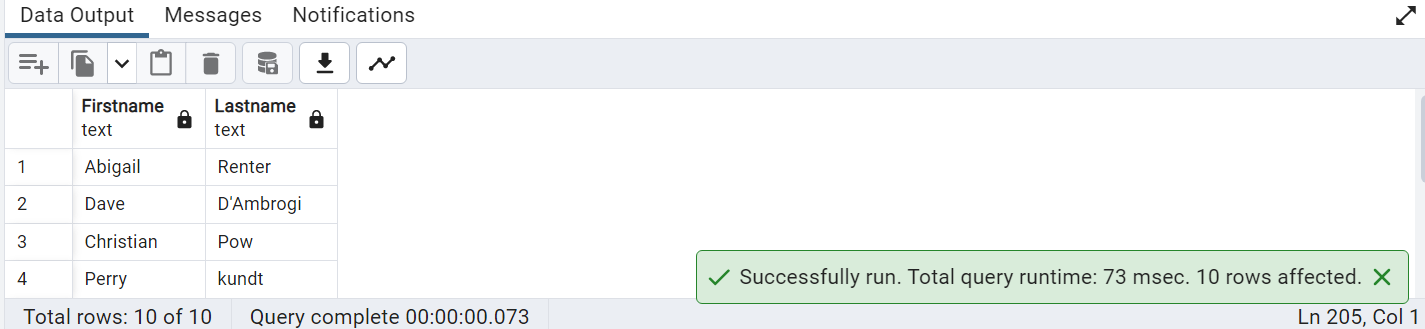
WHERE "Lastname" LIKE '%an%'



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

select "Firstname","Lastname" from "Patients"

limit 10 offset 37



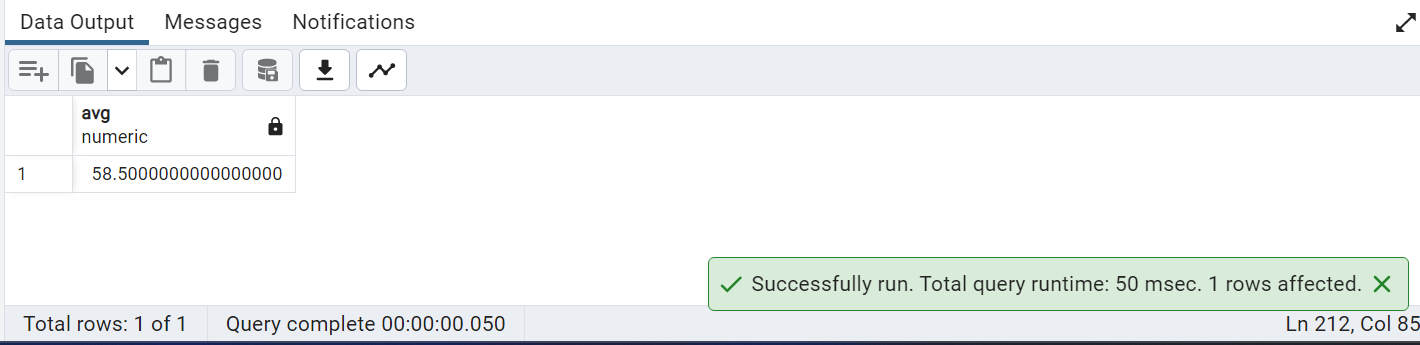
49. Write a query to find Average age for patients with high blood pressure

select avg("Age") from "Blood\_Pressure"

join "Patients" as p on p."Patient\_ID"="Blood\_Pressure"."Patient\_ID"

where "Blood\_Pressure"."24Hr\_Day\_SBP" > 129 and "Blood\_Pressure"."24Hr\_Night\_SBP" >129

and "Blood\_Pressure"."24Hr\_Day\_DBP" >79 and "Blood\_Pressure"."24Hr\_Night\_DBP" >79



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

CREATE MATERIALIZED VIEW Number\_Of\_Gender

AS

select G."Gender",count(G."Gender") "No\_Of\_Gender" from

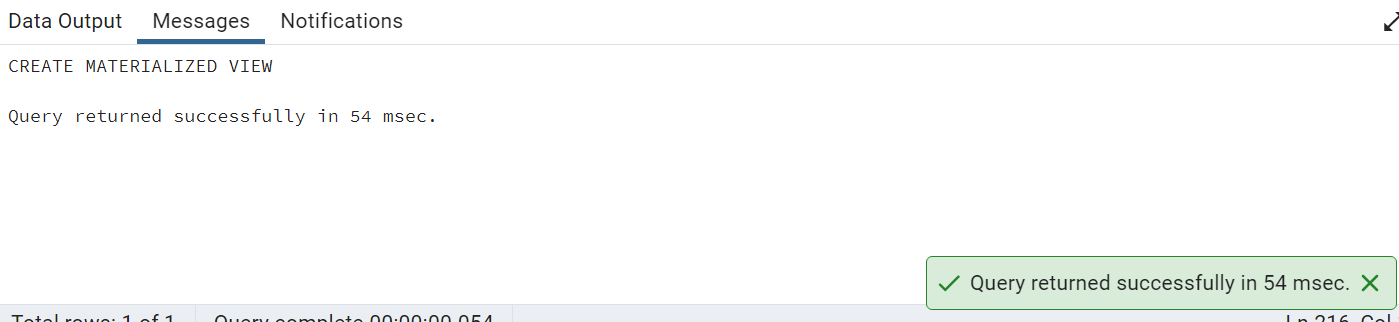
"Patients" P

inner join "Gender" G on G."Gender\_ID"=P."Gender\_ID"

group by G."Gender"

WITH NO DATA;



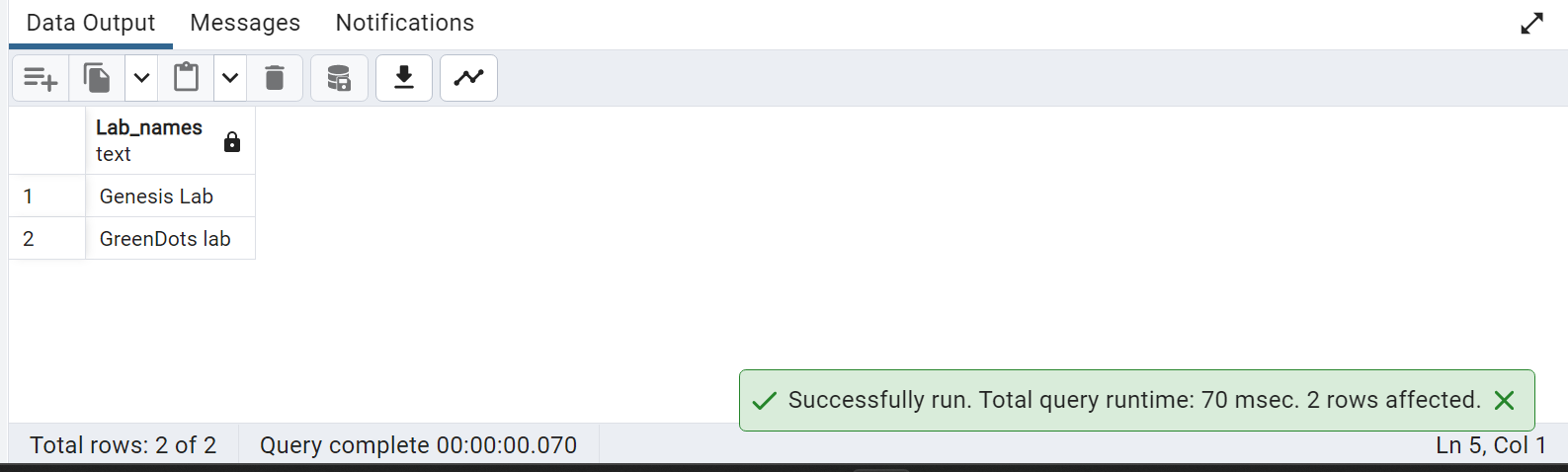


—

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

SELECT DISTINCT("Lab\_names") FROM public."Lab\_Visit"

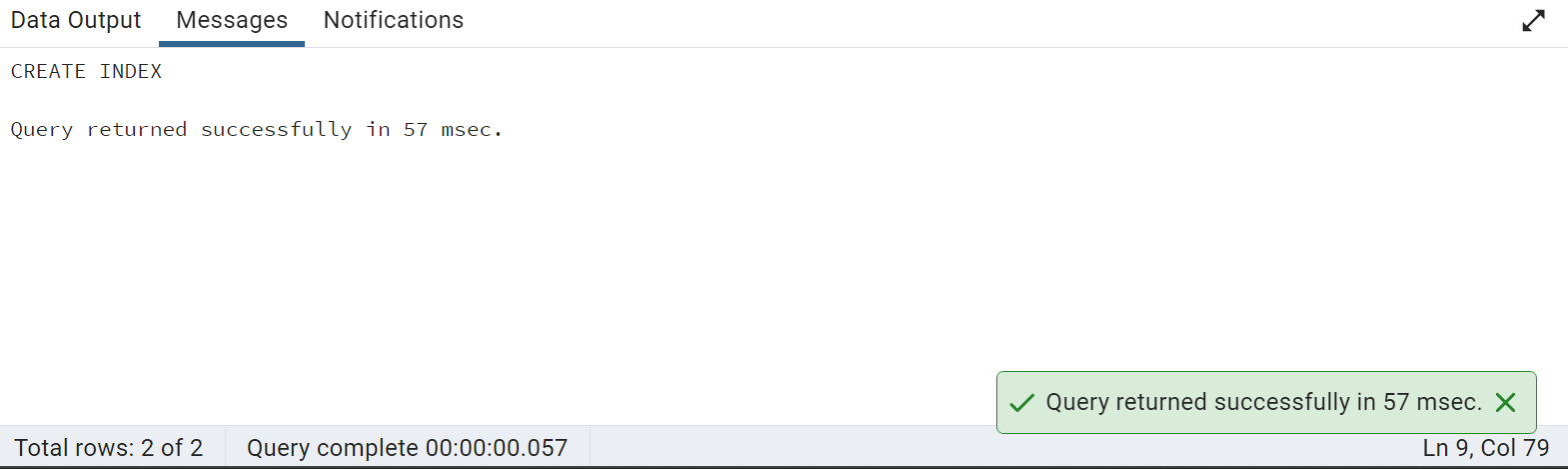
WHERE "Lab\_names" LIKE 'G%b';



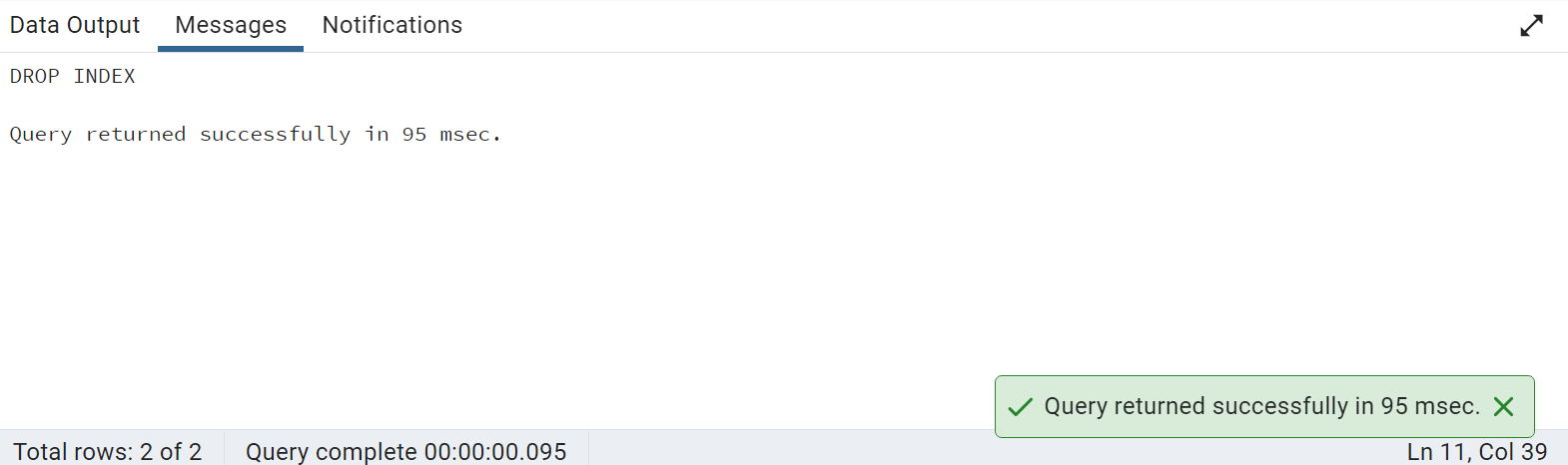
52)Write the query to create an Index on table Verbal\_Cognitive by selecting a column and also write the query drop the same index.

– index created on Verbal Cognitive table based on VC\_ID column

CREATE INDEX idx\_verbal\_cognitive\_vc\_id ON public."Verbal\_Cognitive"("VC\_ID");



DROP INDEX idx\_verbal\_cognitive\_vc\_id;



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

–Birth Year is calculated and MOD function is used to calculate remainder to check if birth year is a leap year

SELECT COUNT(\*)As "Patients born in Leap Year" FROM public."Patients"

WHERE MOD(((Extract(year from "Visit\_Date"))-"Age"),4)=0;



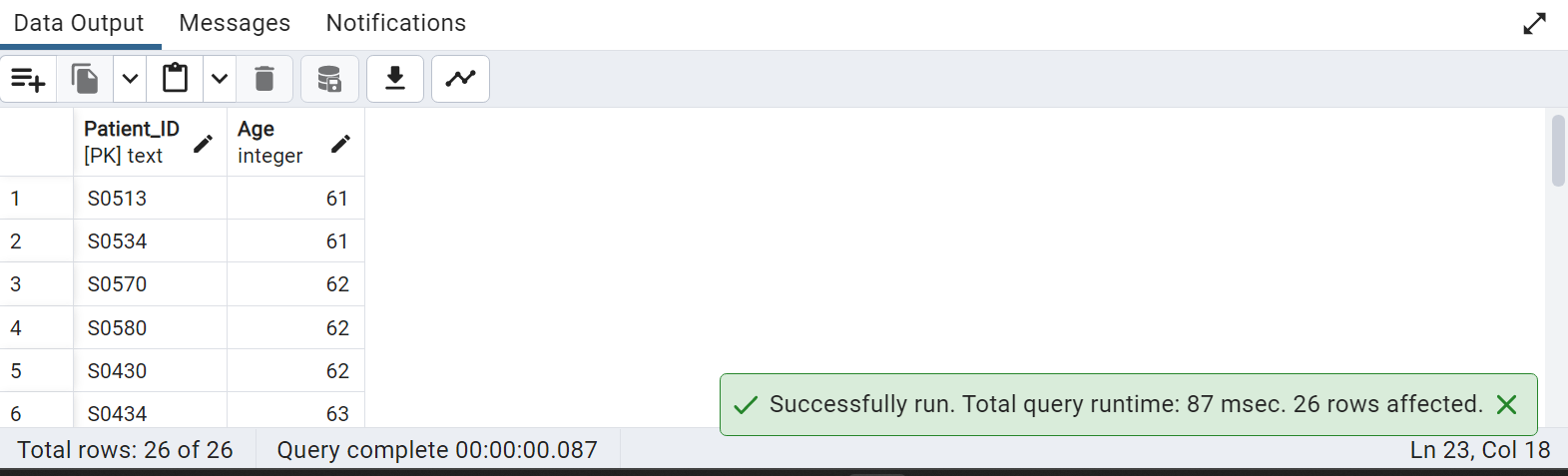
54) Write a query to get a list of patient IDs from the DM group and above age 60 in sequence.

SELECT p."Patient\_ID",p."Age" FROM public."Patients" p

JOIN public."Group" g ON p."Group\_ID"=g."Group\_ID"

WHERE g."Group"='DM' AND p."Age">60

ORDER BY p."Age";



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

WITH max\_damage AS

(

– Subquery to calculate maximum diabetic retinopathy

SELECT "Opthal\_ID", MAX("Diabetic\_Retinopathy") AS most\_damage FROM public."Opthalmology"

GROUP BY "Opthal\_ID"

)

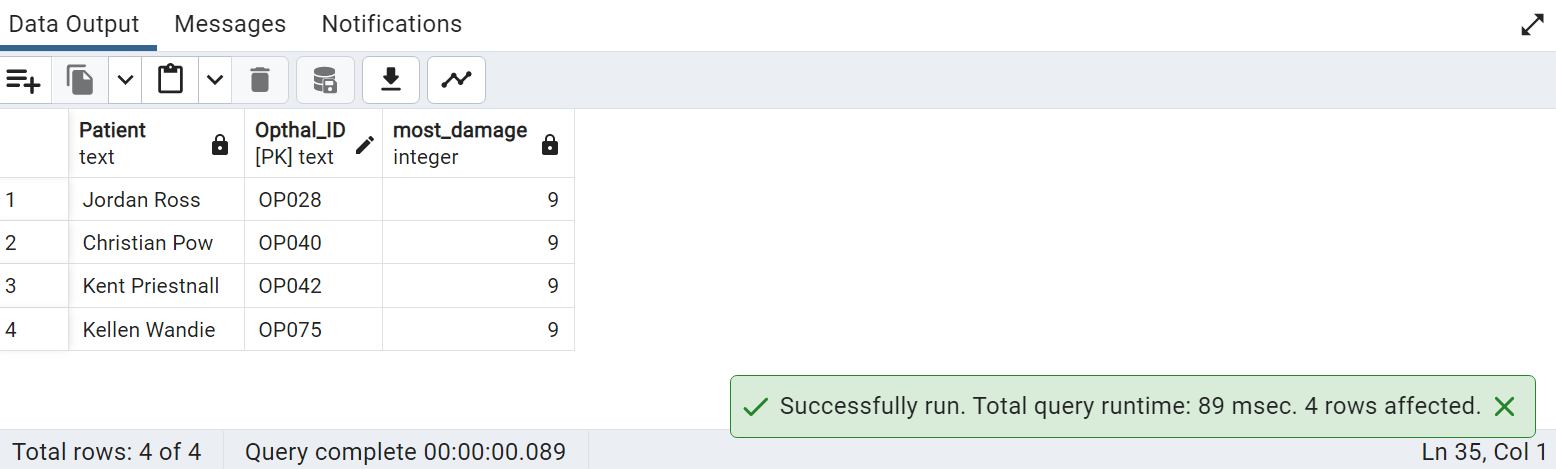
– concatenated names and highest retinopathy damage is selected

SELECT CONCAT(p."Firstname",' ', p."Lastname") As "Patient", o."Opthal\_ID", o.most\_damage FROM max\_damage o

JOIN "Patients" p ON p."Opthal\_ID" = o."Opthal\_ID"

WHERE o.most\_damage = (SELECT MAX(most\_damage) FROM max\_damage)

ORDER BY o.most\_damage DESC;



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

CREATE OR REPLACE PROCEDURE chk\_if\_race\_exists(IN race\_name VARCHAR)

LANGUAGE plpgsql

AS $$

BEGIN

– Check if the race exists in ‘Race’ Table

IF EXISTS(SELECT 1 FROM public."Race" WHERE "Race"=race\_name)

– Raise a notice message indicating race exists

THEN RAISE NOTICE 'Race % Exists',race\_name;

ELSE

– Raise a notice message indicating race does not exist

RAISE NOTICE 'Race % Does Not Exist',race\_name;

END IF;

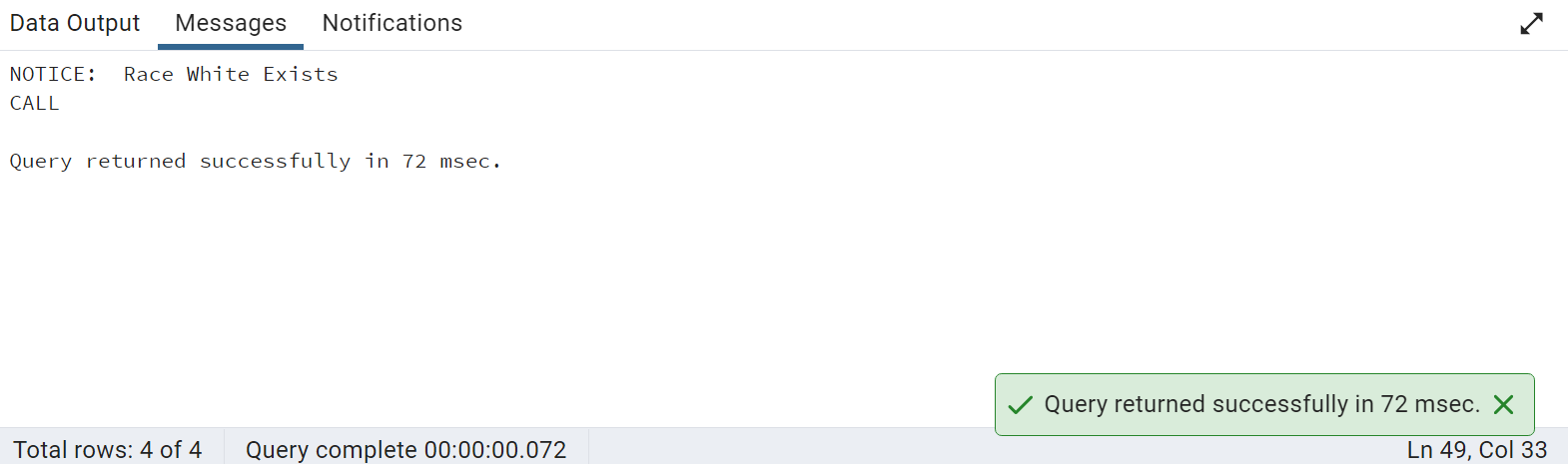
END;

$$;



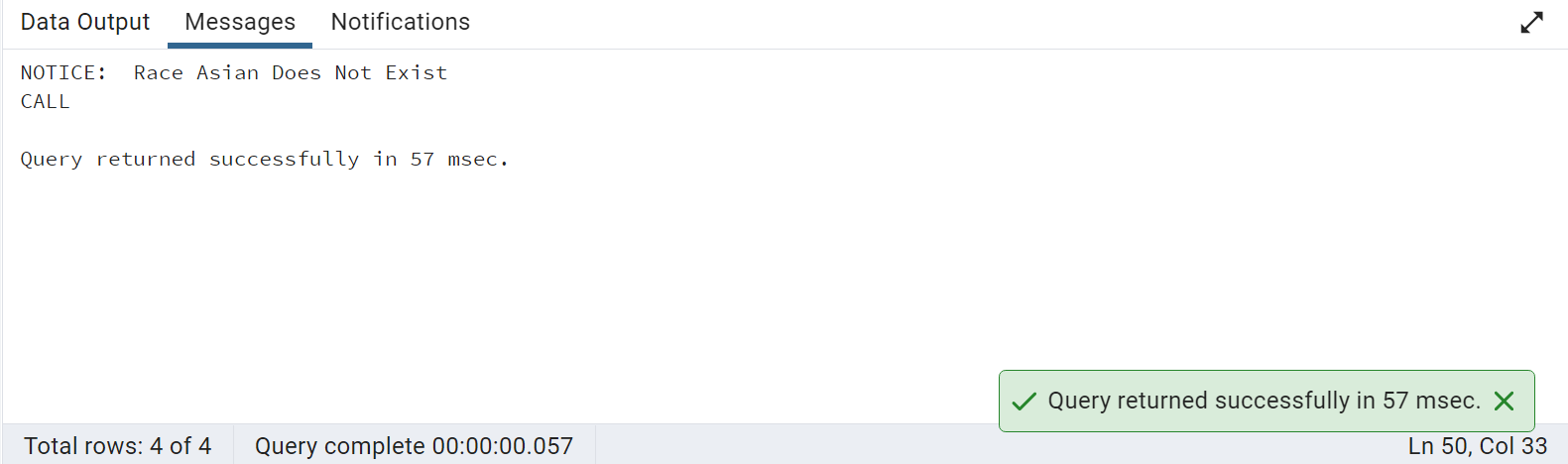
-- Check procedure with existing race

CALL chk\_if\_race\_exists('White')



-- Check procedure with non existing race

CALL chk\_if\_race\_exists('Asian')



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

SELECT p."Patient\_ID",p."Firstname",p."Lastname",p."Age",gn."Gender",r."Race",gr."Group",llt."Fasting\_Triglyc"

FROM public."Patients" p

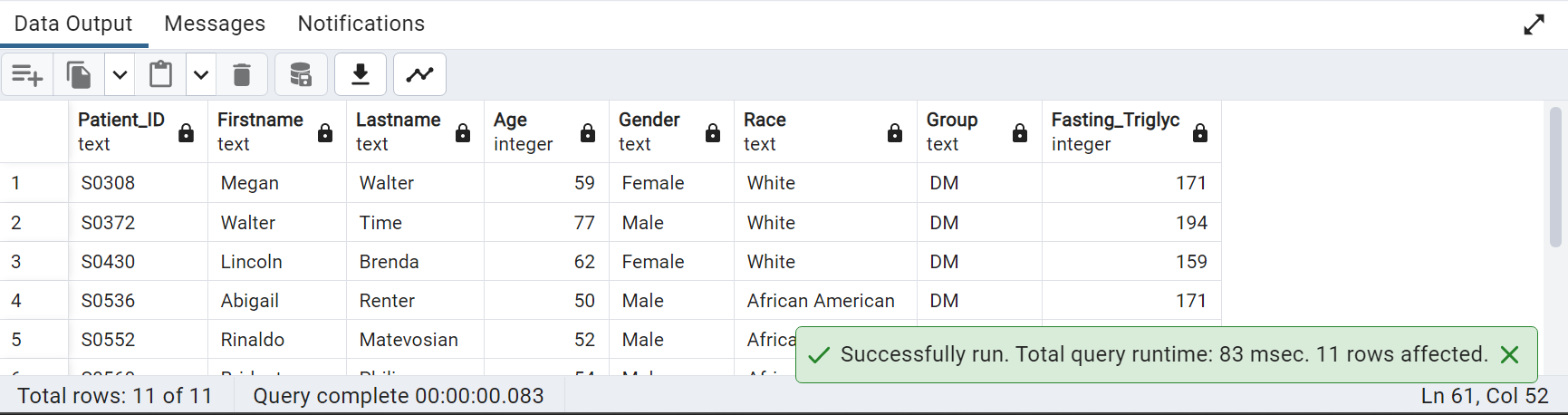
JOIN public."Gender" gn ON p."Gender\_ID"=gn."Gender\_ID"

JOIN public."Race" r ON p."Race\_ID"=r."Race\_ID"

JOIN public."Lipid\_Lab\_Test"llt ON p."Patient\_ID"=llt."Patient\_ID"

JOIN public."Group" gr ON gr."Group\_ID"=p."Group\_ID"

WHERE gr."Group"='DM' AND llt."Fasting\_Triglyc">150



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

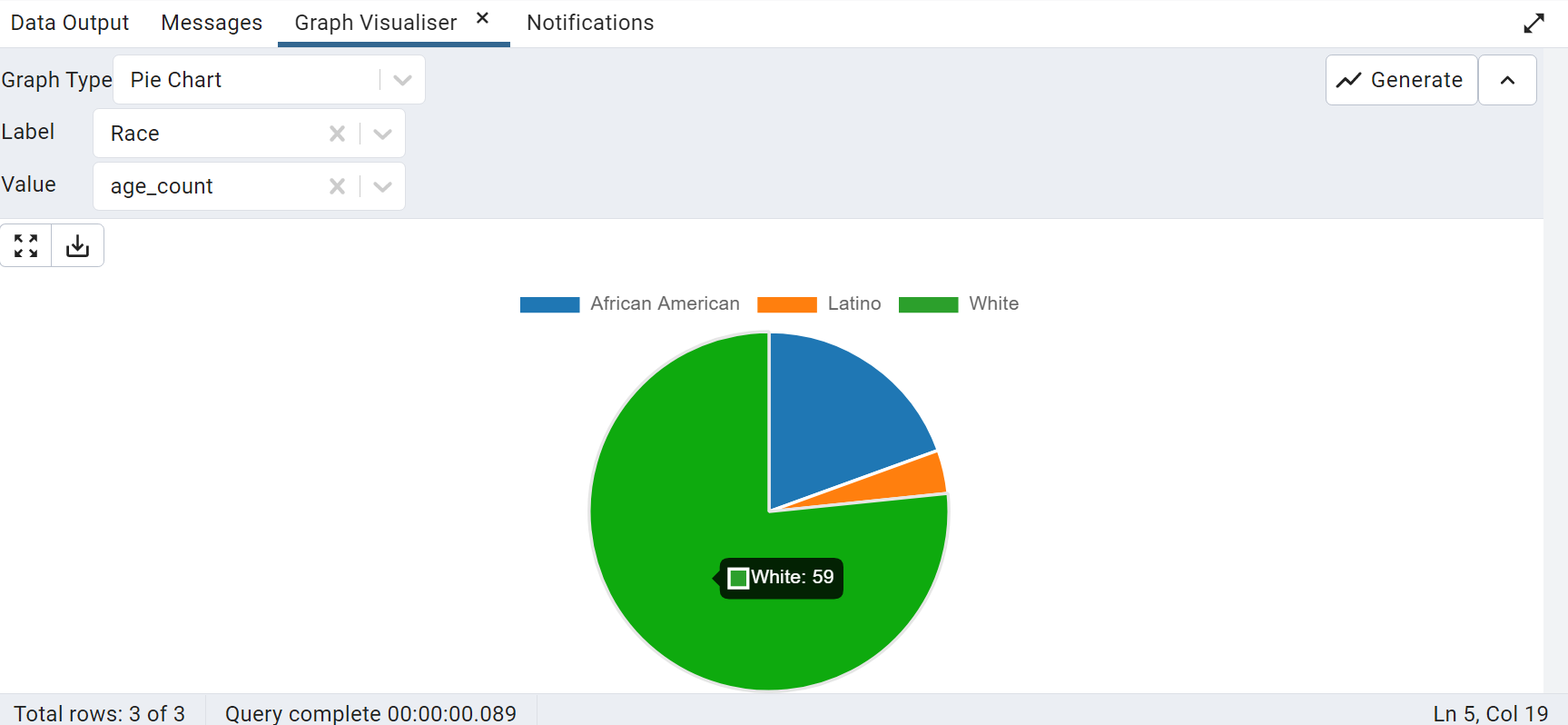
SELECT R."Race", COUNT(\*) AS Age\_Count

FROM public."Patients" P

JOIN public."Race" R ON R."Race\_ID" = P."Race\_ID"

GROUP BY R."Race"

ORDER BY R."Race";



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.

CREATE TABLE "Patient" (

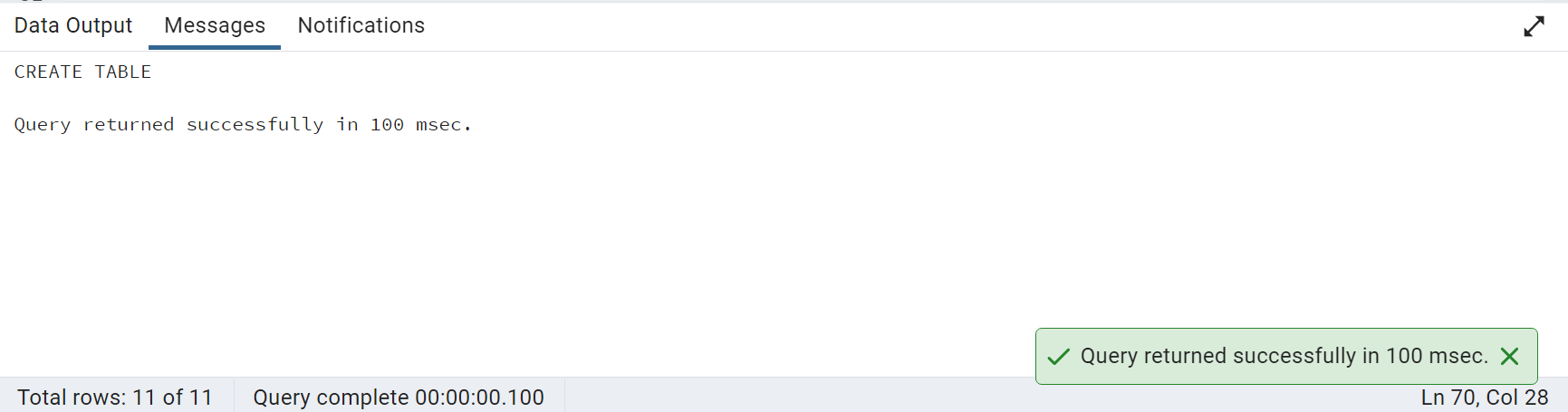
"patient\_id" INT PRIMARY KEY,

"Firstname" VARCHAR(50),

"last\_name" VARCHAR(50),

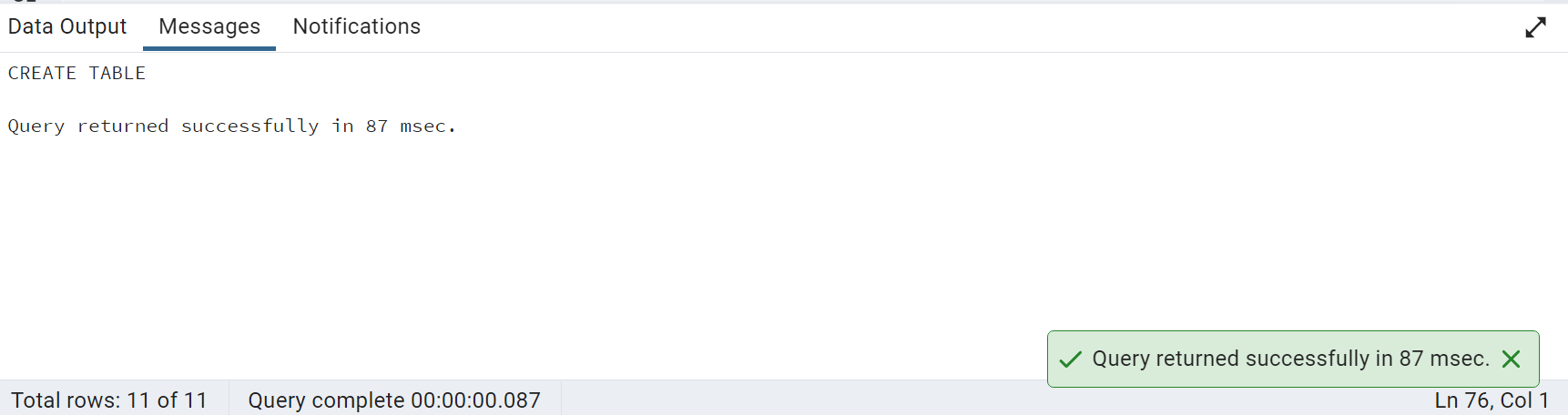
"Visit\_Date" DATE,

"Gender\_ID" VARCHAR(10));



CREATE TABLE "ChildPatient" (

) INHERITS ("Patient");



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

CREATE FUNCTION fn\_chk\_abn\_values() RETURNS TRIGGER AS $$

BEGIN

-- Check if the inserted row has abnormal HbA1C and fasting glucose values

IF NEW."Hb\_A1C">5.7 AND NEW."Fasting\_Glucose">100 THEN

-- Raise a notice message indicating abnormal values

RAISE NOTICE 'Abnormal HbA1C and fasting glucose values detected for patient %', NEW."Patient\_ID";

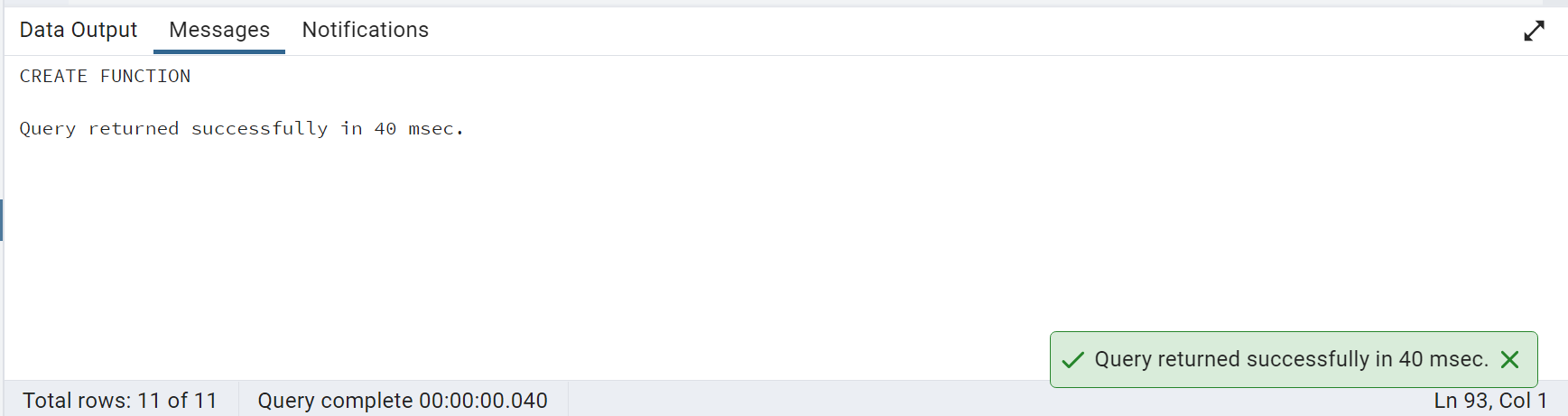
END IF;

-- Return the new row to complete the trigger

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

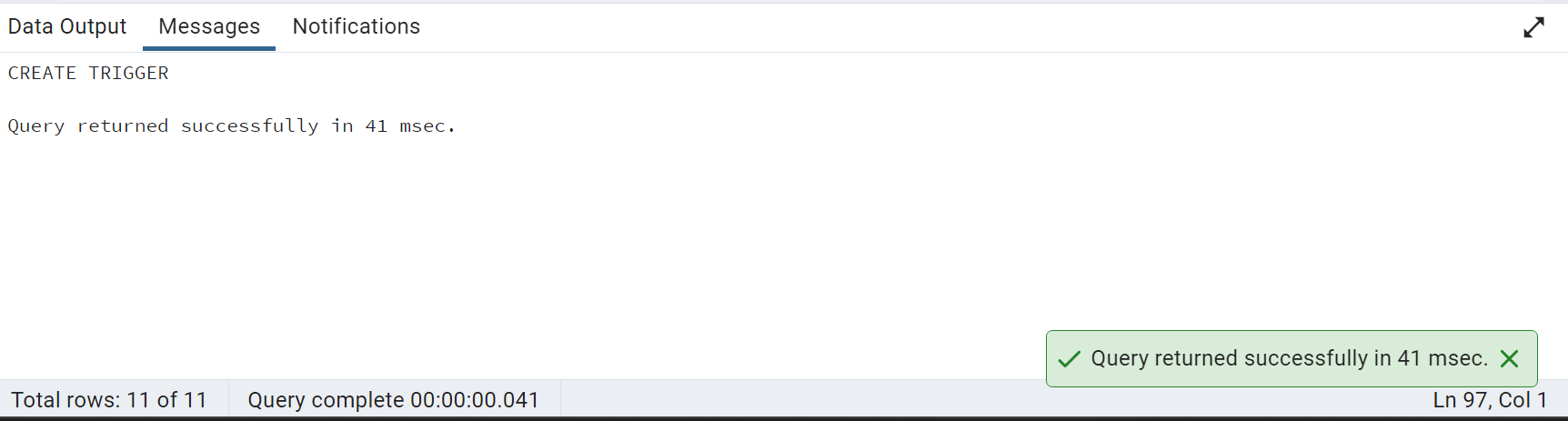


-- For each insert operation, trigger function to check abnormal values

CREATE TRIGGER chk\_abn\_values\_trigger

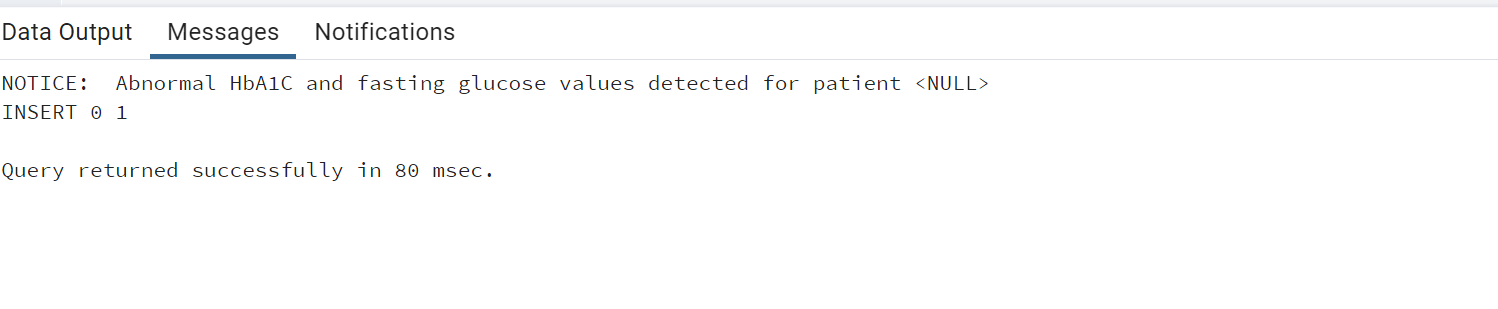
AFTER INSERT ON public."Lab\_Test"

FOR EACH ROW EXECUTE FUNCTION fn\_chk\_abn\_values();



– Try to insert abnormal values in Lab test table to check trigger

INSERT INTO public."Lab\_Test"("Lab\_ID","Hb\_A1C","Fasting\_Glucose") VALUES ('LB079',6.0,120)



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

SELECT

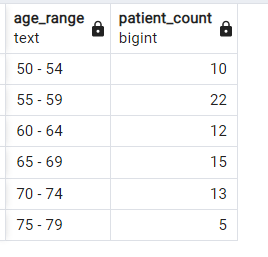
CONCAT((WIDTH\_BUCKET(p."Age", 0, 100, 20) - 1) \* 5, ' - ', WIDTH\_BUCKET(p."Age", 0, 100, 20) \* 5 - 1) AS age\_range,

COUNT (\*) AS PATIENT\_COUNT

FROM PUBLIC."Patients" P

GROUP BY WIDTH\_BUCKET(p."Age", 0, 100, 20)

ORDER BY age\_range ASC;





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

-- /\*normal platelet count in adults ranges from 150,000 to 450,000 platelets per microliter of blood.

select GP."Group\_ID", count(P."Patient\_ID") "patients\_Count"

from "Patients" P

inner join "Gender" G on G."Gender\_ID"=P."Gender\_ID"

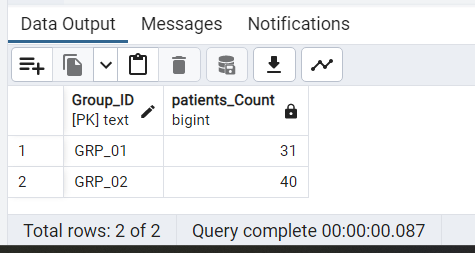
inner join "Lab\_Test" LT on LT."Patient\_ID"=P."Patient\_ID"

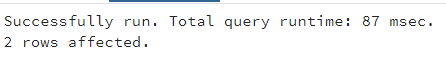
inner join "Link\_Reference" LR on LT."Lab\_ID"=LR."Lab\_ID"

inner join "Group" GP on GP."Group\_ID"=P."Group\_ID"

where LT."Platelets" between 150 and 450

group by GP."Group\_ID"





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

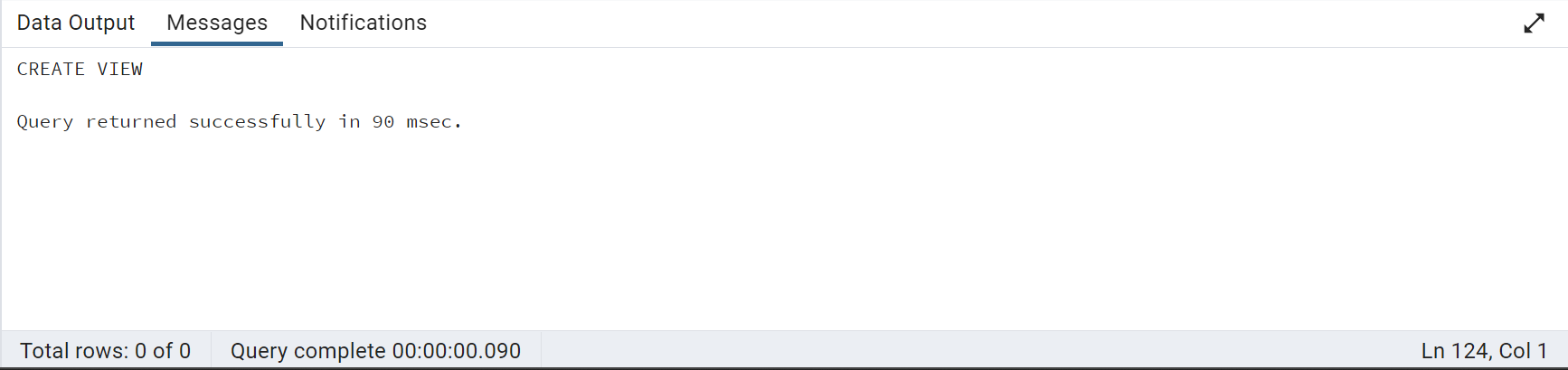
– Create a view on Blood\_Pressure table

CREATE OR REPLACE VIEW vw\_blood\_pressure

AS

(SELECT "BP\_ID","Patient\_ID","24Hr\_Day\_HR","24Hr\_Night\_HR"

FROM public."Blood\_Pressure");



–create a function to check abnormal heart rate

CREATE OR REPLACE FUNCTION check\_hr\_input() RETURNS TRIGGER AS $$

BEGIN

-- Check if HR columns are inserted abnormal values

IF NEW."24Hr\_Day\_HR">120 OR NEW."24Hr\_Night\_HR" >100 THEN

RAISE NOTICE 'Abnormal Heart Rate';

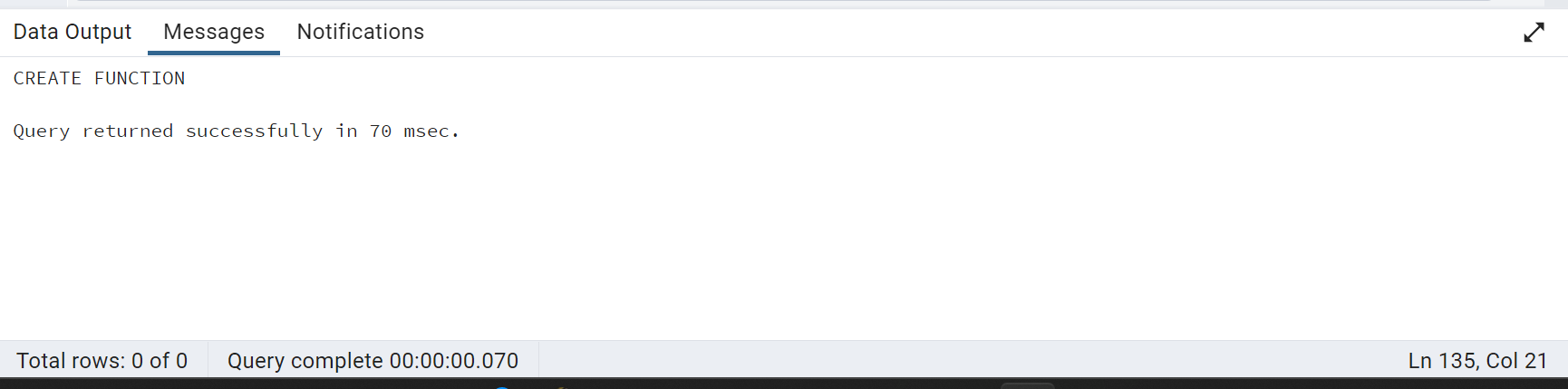
END IF;

-- Return the result of the trigger function

RETURN NEW;

END;

$$ LANGUAGE plpgsql;



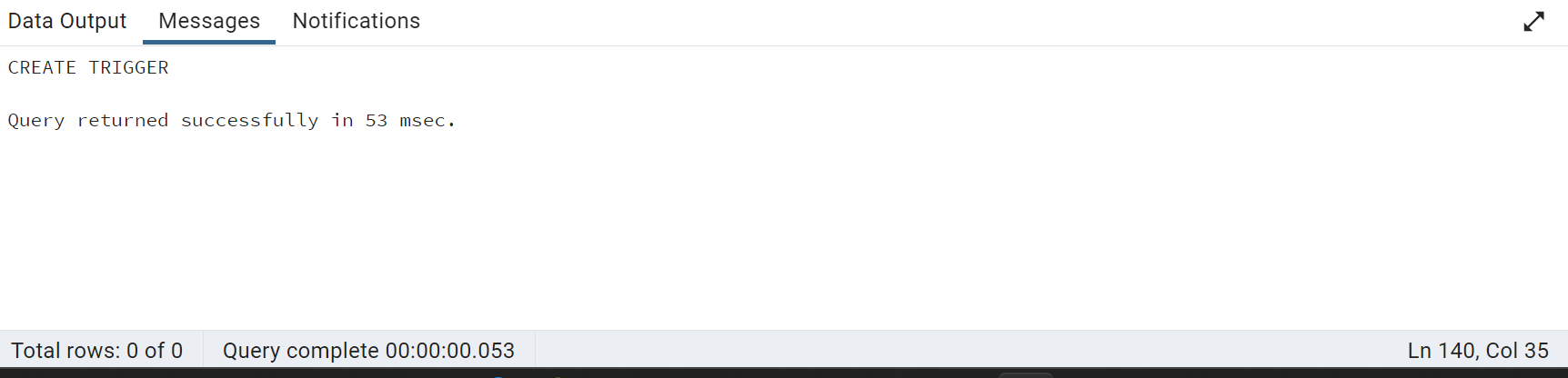
– Create a trigger on view

CREATE TRIGGER check\_hr\_input\_trigger

INSTEAD OF INSERT ON "vw\_blood\_pressure"

FOR EACH ROW

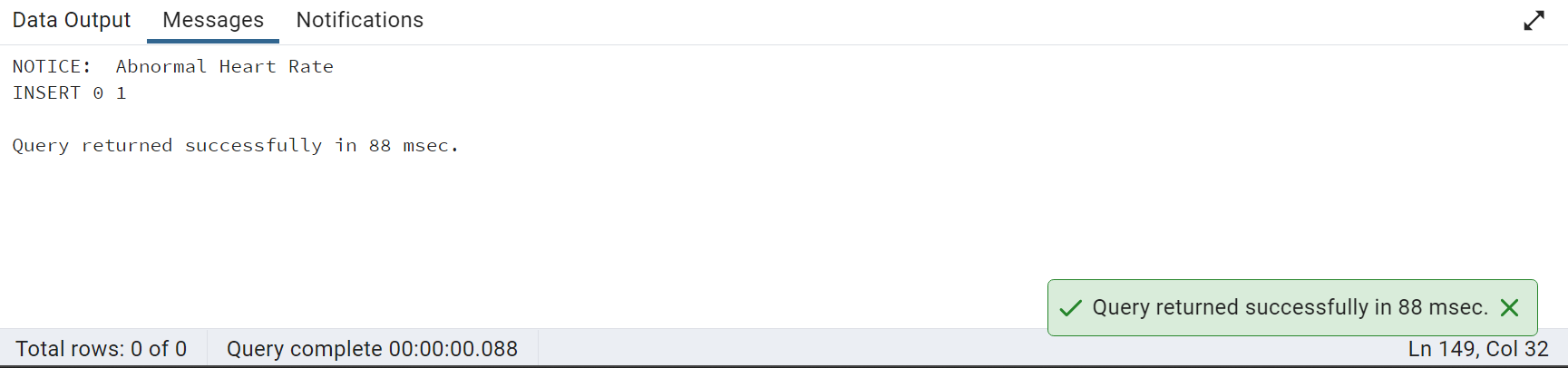
EXECUTE FUNCTION check\_hr\_input();



– Try inserting a record in View with abnormal HR value

INSERT INTO vw\_blood\_pressure("BP\_ID","Patient\_ID","24Hr\_Day\_HR","24Hr\_Night\_HR")

VALUES('BP079','S0612',125,105)



64 Write a query to find the number of Patients whose Gait RPE start is greater than the end and vice versa

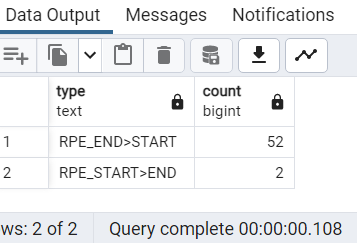
SELECT 'RPE\_END>START' AS type,COUNT ("Patient\_ID") FROM public."Walking\_Test"

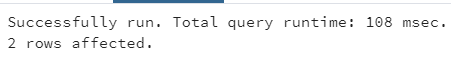
WHERE ("Gait\_RPE\_Start " < "Gait\_RPE\_End " )

UNION

SELECT 'RPE\_START>END'AS type,COUNT ("Patient\_ID") FROM public."Walking\_Test"

WHERE ("Gait\_RPE\_Start " > "Gait\_RPE\_End " )





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

CREATE TABLE Customers

(customer\_id int NOT NULL,

customer\_name char(50) NOT NULL,

address char(80),

place char(25));

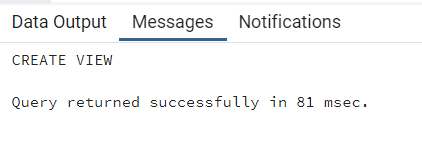
INSERT INTO Customers(customer\_id,customer\_name,address,place )

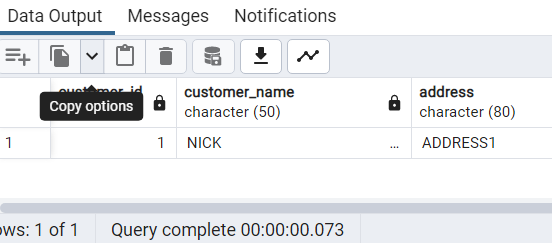
VALUES(1,'NICK','ADDRESS2','USA');

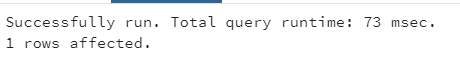
CREATE VIEW CustomerViews AS

SELECT customer\_id, customer\_name, address, place

FROM Customers;







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

select "Firstname","Lastname"

from "Patients" P

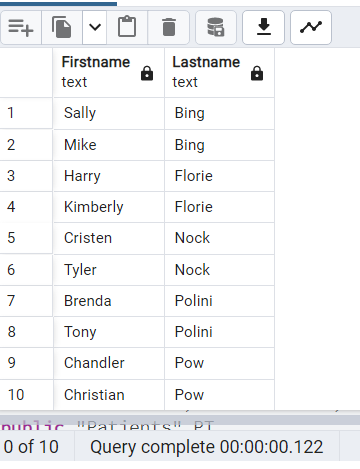
where exists (select "Firstname","Lastname"

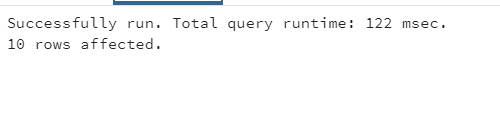
from "Patients" P2

where P2."Lastname" = P."Lastname"

and P2."Patient\_ID" <> P."Patient\_ID")

order by "Lastname"

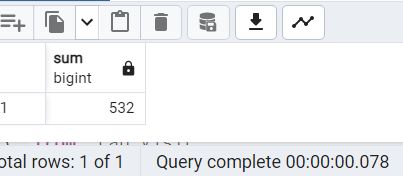


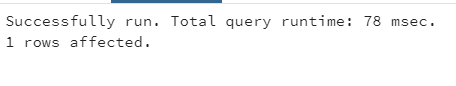


67)Write a query to get the Sum of Diabetes Duration for Group id 'GRP\_02'.

SELECT SUM("Diabetes\_Duration") FROM public."Patients"

WHERE "Group\_ID"= 'GRP\_02';





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

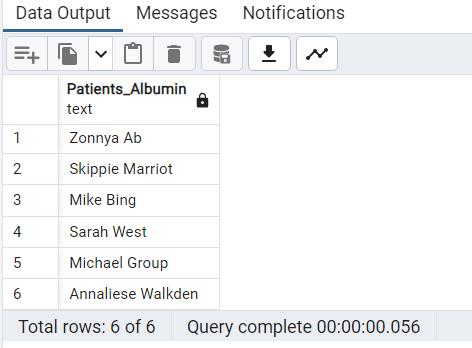
SELECT CONCAT (A."Firstname" || ' ' || A."Lastname") AS "Patients\_Albumin"

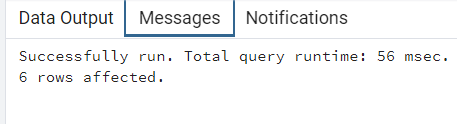
FROM "Patients" A

INNER JOIN "Link\_Reference" B ON A."Link\_Reference\_ID" = B."Link\_Reference\_ID"

INNER JOIN "Urine\_Test"C ON B."Urine\_ID" = C."Urine\_ID"

WHERE "Albumin" >= '30';





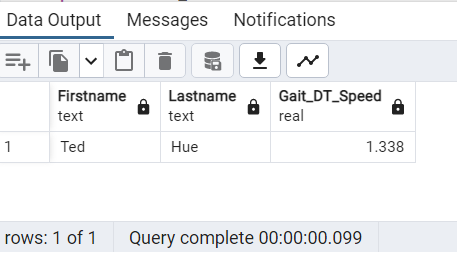
--69) Get the patient's name who has a max speed.

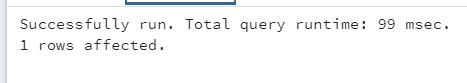
SELECT PT."Firstname",PT."Lastname","Gait\_DT\_Speed" FROM public."Walking\_Test" W

JOIN public."Patients" PT

ON PT."WalkTest\_ID" = W."WalkTest\_ID"

ORDER BY "Gait\_DT\_Speed" DESC LIMIT 1;





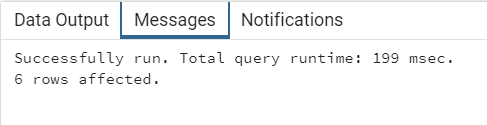
70.Write a query to find out the percentage of Lab visits by Lab names.

SELECT "Lab\_names", (count(\*)/ SUM(count(\*)) OVER ()) \* 100 AS percentage

FROM public."Lab\_Visit"

GROUP by "Lab\_names";





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)

SELECT VC."Patient\_ID"

FROM public."Verbal\_Cognitive" VC

JOIN public."Link\_Reference" LR ON VC."VC\_ID" = LR."VC\_ID"

JOIN public."Memory\_Cognitive" MC ON LR."MC\_ID" = MC."MC\_ID"

WHERE (

(CASE WHEN VC."DS" <= 13 THEN 1 ELSE 0 END) +

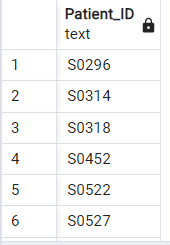
(CASE WHEN VC."HVLT" <= 14 THEN 1 ELSE 0 END) +

(CASE WHEN VC."VF" <= 42 THEN 1 ELSE 0 END) +

(CASE WHEN VC."WTAR" <= 20 THEN 1 ELSE 0 END) +

(CASE WHEN MC."GDS" >= 15 THEN 1 ELSE 0 END)

) >= 2;





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

SELECT P."Patient\_ID",P."Firstname",P."Lastname"

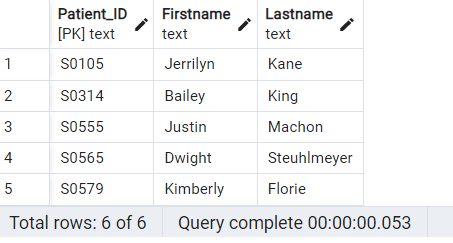
FROM public."Patients" P

JOIN public."Link\_Reference" LR ON LR."Link\_Reference\_ID" = P."Link\_Reference\_ID"

JOIN public."Memory\_Cognitive" MC ON MC."MC\_ID" = LR."MC\_ID"

WHERE P."Diabetes\_Duration" BETWEEN 5 AND 30

AND MC."GDS" >= 15;





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

SELECT COUNT(DISTINCT "Patient\_ID") AS num\_visits

FROM public."Patients"

WHERE "Visit\_Date" >= '2005-03-01' AND "Visit\_Date" < '2006-04-01';





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

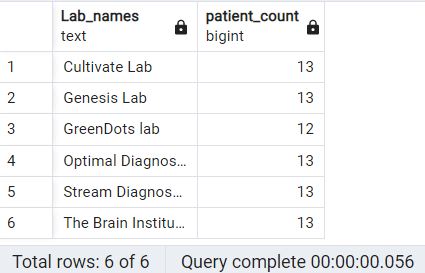
SELECT LV."Lab\_names",COUNT(DISTINCT P."Patient\_ID") AS Patient\_Count

FROM public."Lab\_Visit" LV

JOIN publ*ic."Link\_Reference" LR ON LV."Lab\_visit\_ID" = LR."Lab\_visit\_ID"*

*JOIN public."Patien*ts" P ON LR."Link\_Reference\_ID" = P."Link\_Reference\_ID"

GROUP BY LV."Lab\_names";





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

SELECT G."Group",LV."Lab\_names",COUNT(DISTINCT P."Patient\_ID") AS Patient\_Count

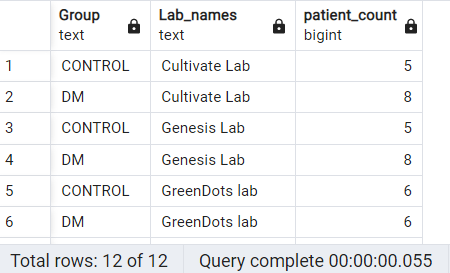
FROM public."Group" G

JOIN public."Patients" P ON G."Group\_ID" = P."Group\_ID"

JOIN public."Link\_Reference" LR ON P."Link\_Reference\_ID" = LR."Link\_Reference\_ID"

JOIN public."Lab\_Visit" LV ON LR."Lab\_visit\_ID" = LV."Lab\_visit\_ID"

GROUP BY LV."Lab\_names",G."Group\_ID";



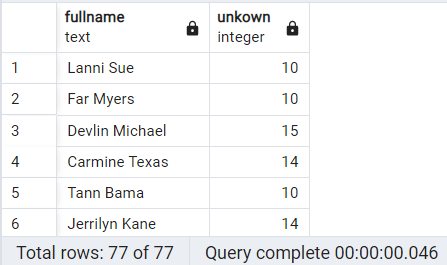


76) Please go through the below screenshot and create the exact output.

SELECT CONCAT("Firstname", ' ',"Lastname") AS Fullname,

LENGTH(CONCAT("Firstname", ' ',"Lastname"))+1 as unkown

FROM public."Patients";





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

SELECT

CONCAT(

P."Patient\_ID", ',',

G."Group", ',',

R."Race", ',',

GD."Gender", ',',

O."Diabetic\_Retinopathy", ',',

LT."Fasting\_Glucose"

) AS PatientID\_Group\_Race\_Gender\_Diabetic\_Retinopathy\_Fasting\_Glucose

FROM public."Patients" P

JOIN public."Group" G ON P."Group\_ID" = G."Group\_ID"

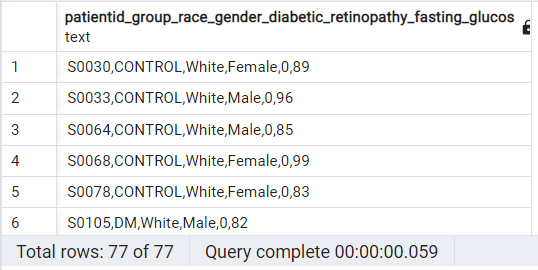
JOIN public."Race" R ON P."Race\_ID" = R."Race\_ID"

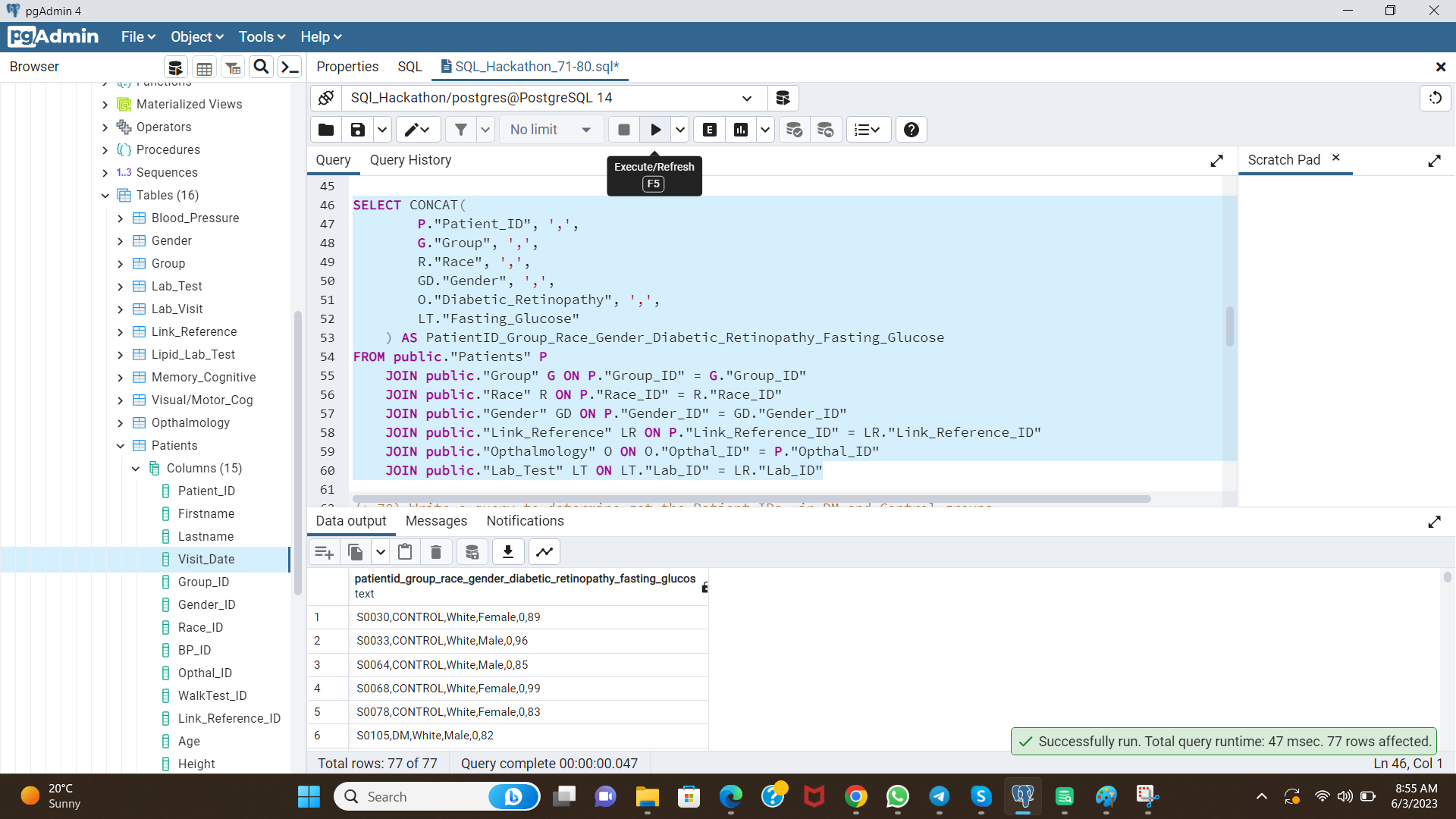
JOIN public."Gender" GD ON P."Gender\_ID" = GD."Gender\_ID"

JOIN public."Link\_Reference" LR ON P."Link\_Reference\_ID" = LR."Link\_Reference\_ID"

JOIN public."Opthalmology" O ON O."Opthal\_ID" = P."Opthal\_ID"

JOIN public."Lab\_Test" LT ON LT."Lab\_ID" = LR."Lab\_ID"





78) Write a query to determine get the Patient IDs ,in DM and Control groups ,

that are in prediabetic stage and label them accordingly.

SELECT P."Patient\_ID",G."Group",

CASE

WHEN LT."Fasting\_Glucose" BETWEEN 100 AND 125 THEN 'Prediabetic'

WHEN LT."Fasting\_Glucose" > 125 THEN 'Diabetic'

ELSE 'Normal'

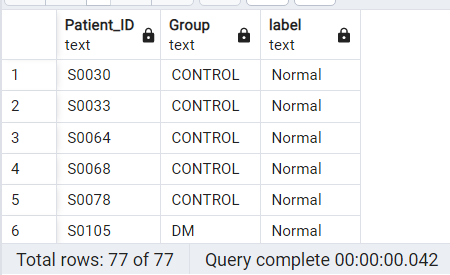
END AS Label

FROM public."Patients" P

JOIN public."Group" G ON G."Group\_ID" = P."Group\_ID"

JOIN public."Link\_Reference" LR ON LR."Link\_Reference\_ID" = P."Link\_Reference\_ID"

JOIN public."Lab\_Test" LT ON LT."Lab\_ID" = LR."Lab\_ID";





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

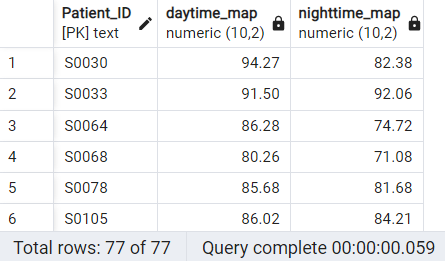
SELECT P."Patient\_ID",

CAST((BP."24Hr\_Day\_DBP" + ((BP."24Hr\_Day\_SBP" - BP."24Hr\_Day\_DBP") / 3.0)) AS numeric(10, 2)) AS daytime\_map,

CAST((BP."24Hr\_Night\_DBP" + ((BP."24Hr\_Night\_SBP" - BP."24Hr\_Night\_DBP") / 3.0)) AS numeric(10, 2)) AS nighttime\_map

FROM public."Patients" P

JOIN public."Blood\_Pressure" BP ON BP."BP\_ID" = P."BP\_ID";





80) Write a query using recursive view.

WITH RECURSIVE GP\_01 AS (

SELECT "Patient\_ID", "Group\_ID", "Firstname"

FROM public."Patients"

WHERE "Group\_ID" = 'GRP\_01'

UNION

SELECT p."Patient\_ID", p."Gender\_ID", p."Firstname"

FROM public."Patients" p

INNER JOIN GP\_01 s ON s."Patient\_ID" = p."Group\_ID")

SELECT "Patient\_ID","Firstname","Group\_ID"

FROM GP\_01;