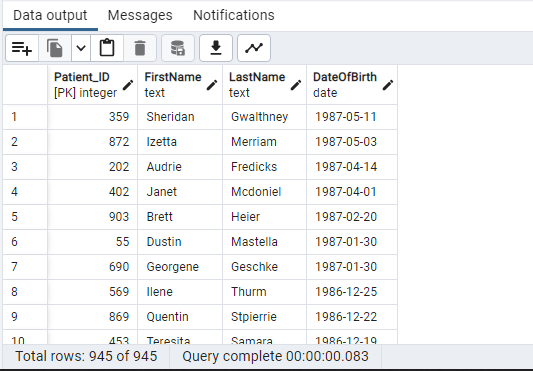
**SQL HACKATHON - 80 QUERIES/OUTPUT**

1. Get list of Patients ORDER BY DateOfBirth descending order.

Query:

SELECT "Patient\_ID", "FirstName","LastName" ,"DateOfBirth" FROM "Patients" ORDER BY "DateOfBirth" DESC



2. Display the firstname and lastname of patients who speaks English language.

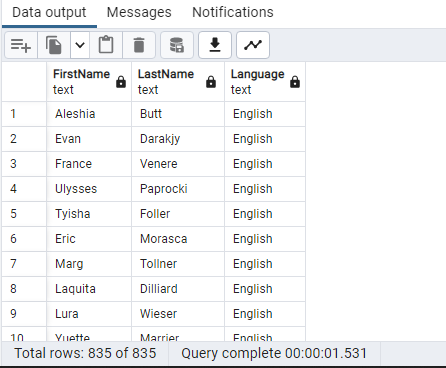
Query:

SELECT "FirstName", "LastName","Language"

FROM "Patients" P JOIN "Language" L

ON P."Language\_ID" = L."Language\_ID"

WHERE "Language" = 'English';



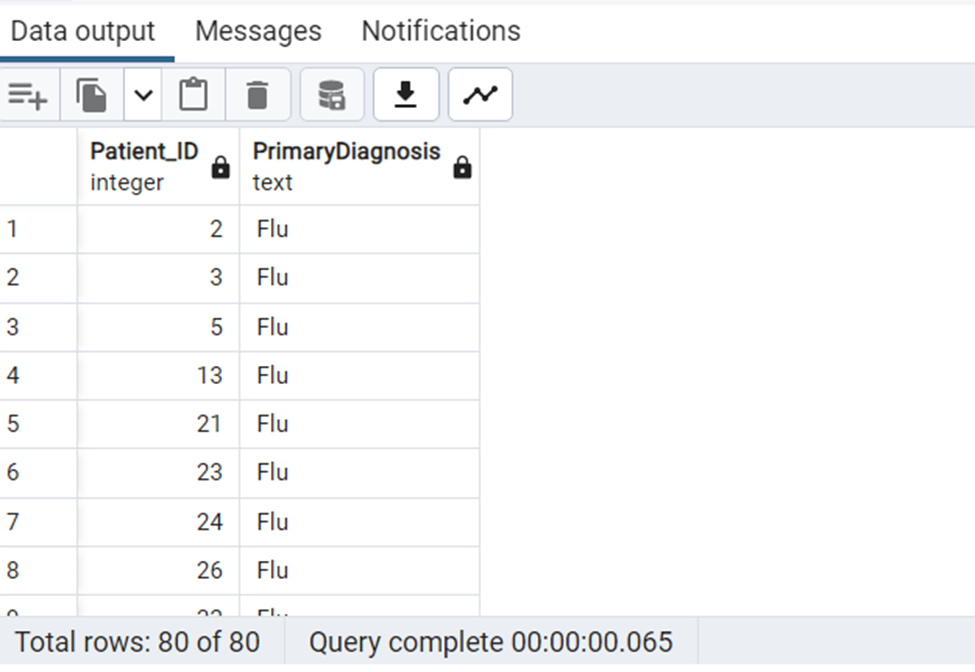
3.Write a query to get list of patient ID's whose PrimaryDiagnosis is 'Flu' ORDER BY patient\_ID

Query:

SELECT d."Patient\_ID" ,pd."PrimaryDiagnosis" FROM public."Discharges" as d

JOIN public."PrimaryDiagnosis" as pd

ON d."Diagnosis\_ID"=pd."Diagnosis\_ID" WHERE "PrimaryDiagnosis"='Flu' ORDER BY "Patient\_ID";



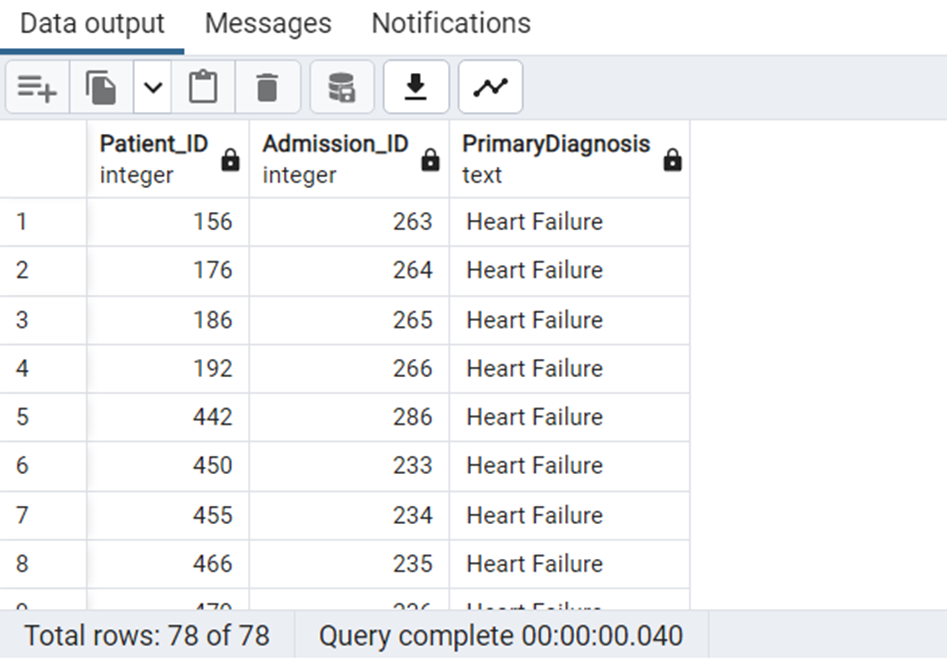
4. Write a query to find the Patient\_ID and Admission\_ID for the patients whose Primary diaganosis is 'Heart Failure'.

Query:

SELECT d."Patient\_ID", d."Admission\_ID" ,pd."PrimaryDiagnosis" FROM public."Discharges" as d

JOIN public."PrimaryDiagnosis" as pd

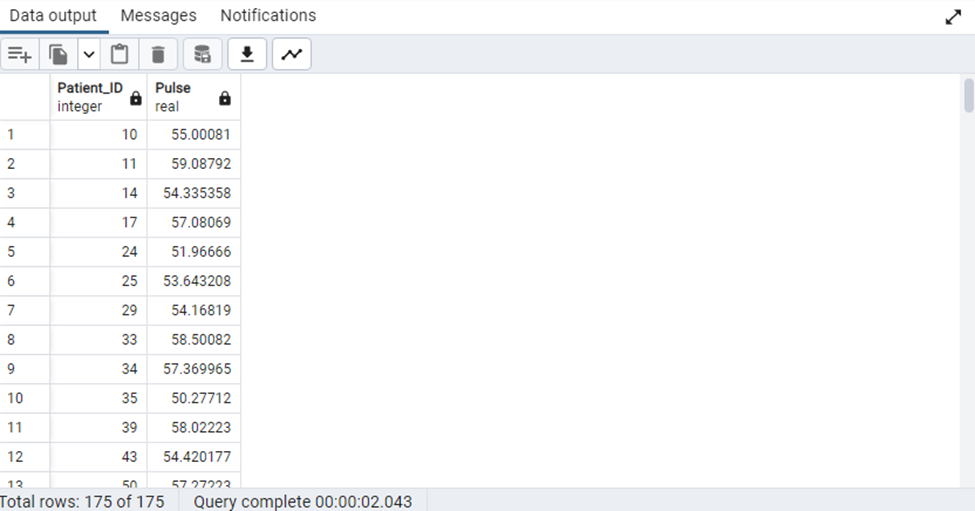
ON d."Diagnosis\_ID"=pd."Diagnosis\_ID" WHERE "PrimaryDiagnosis"='Heart Failure';



5. Write a query to get list of patient ID's whose pulse is below the normal range.

Query:

SELECT "Patient\_ID","Pulse" FROM public."AmbulatoryVisits" WHERE "Pulse" < 60;



--6. Write a query to find the list of patient\_ID's discharged with Service in SID01, SID02, SID03—

Query:

SELECT D."Patient\_ID",D."Service\_ID"

FROM "Discharges" D

WHERE D."Service\_ID" IN ('SID01','SID02','SID03')

Graphical user interface, application, Word

Description automatically generated

--7. Write a query to get list of patients who were admitted because of Stomachache.--

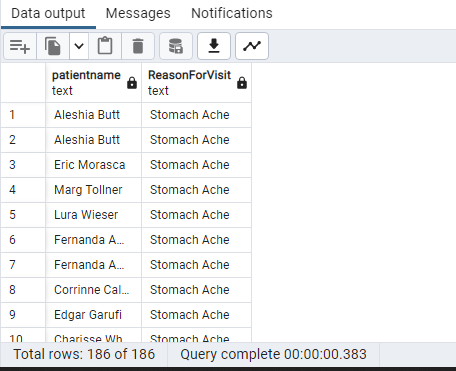
Query:

SELECT P."FirstName" || ' ' || P."LastName" AS PatientName ,"ReasonForVisit"

FROM "Patients" P JOIN "EDVisits" ED ON P."Patient\_ID" = ED."Patient\_ID"

JOIN "ReasonForVisit" RV ON ED."Rsv\_ID" = RV."Rsv\_ID"

WHERE RV."ReasonForVisit" = 'Stomach Ache'



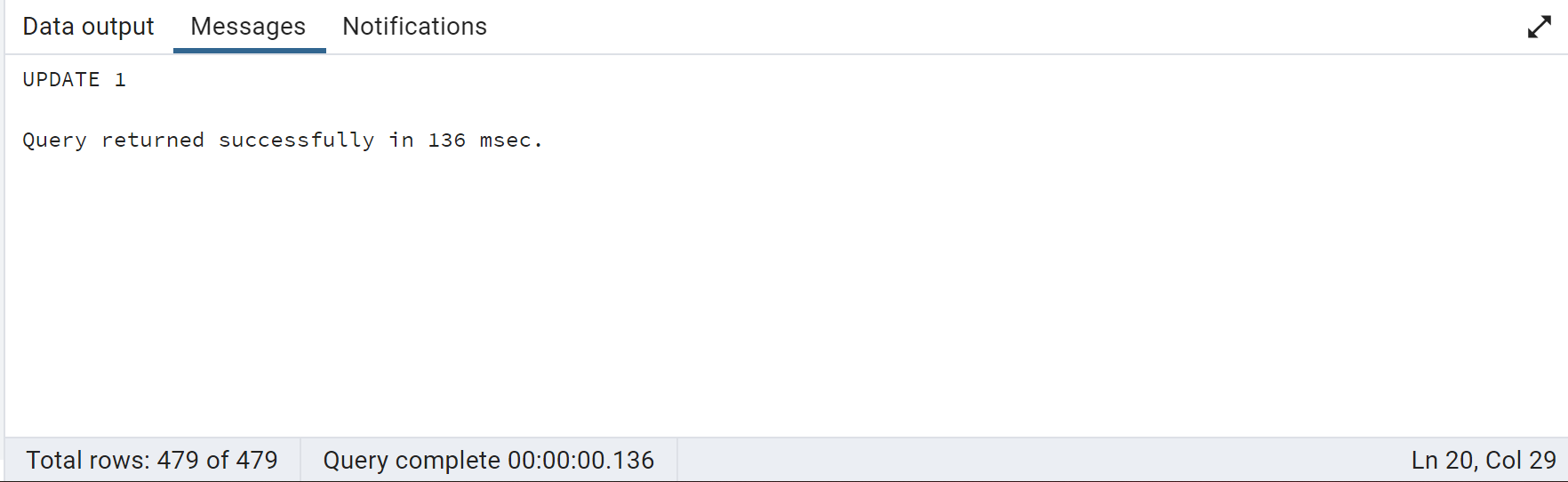
--8. Write a query to Update Service ID SID05 to Ortho--

Query:

UPDATE "Service"

SET "Service" = 'Ortho'

WHERE "Service\_ID" = 'SID05'



Query:

SELECT \* FROM "Service"

Graphical user interface, application

Description automatically generated

--9. Get list of Patient ID's whose visit type was 'Followup' and VisitdepartmentID is 5 or 6--

Query:

SELECT AV."Patient\_ID",VT."VisitType",AV."VisitDepartmentID"

FROM "AmbulatoryVisits" AV JOIN "VisitTypes" VT ON AV."AMVT\_ID" = VT."AMVT\_ID"

WHERE VT."VisitType" = 'Follow Up' AND

(AV."VisitDepartmentID" = 5 OR AV."VisitDepartmentID" = 6)

Graphical user interface, application, Word

Description automatically generated

--10. Create index ONambulatory visit by selecting columns Visit\_ID, AMVT\_ID and VisitStatus\_ID--

Query:

CREATE INDEX ambulatory\_visitid\_amvtid\_visitstatusid\_idx

ON "AmbulatoryVisits"("Visit\_ID","AMVT\_ID","VisitStatus\_ID")

Graphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generated with low confidence

-- 11. Create a trigger to execute after inserting a record into Patients table.Insert value to display result.

Query:

-- Creating Temp table to Record Trigger executionTime

CREATE TABLE Audit\_Patients(PID Integer NOT NULL PRIMARY KEY,entry\_date VARCHAR(100) NOT NULL)

--Creating Trigger Function

CREATE OR REPLACE FUNCTION auditlogfunc()

RETURNS TRIGGER AS $examp\_table$

BEGIN

INSERT INTO audit\_patients(pid,entry\_date) VALUES(NEW."Patient\_ID",CURRENT\_TIMESTAMP);

RETURN NEW;

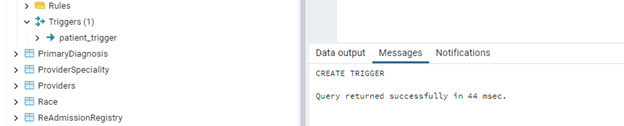
END;

$examp\_table$ LANGUAGE plpgsql;

--Creating Trigger

CREATE TRIGGER patient\_trigger AFTER INSERT ON public."Patients" FOR EACH ROW

EXECUTE PROCEDURE auditlogfunc()



--Testing By Inserting records into Patients

INSERT into "Patients"("Patient\_ID","FirstName","LastName") VALUES ('946','KKK','LLL')

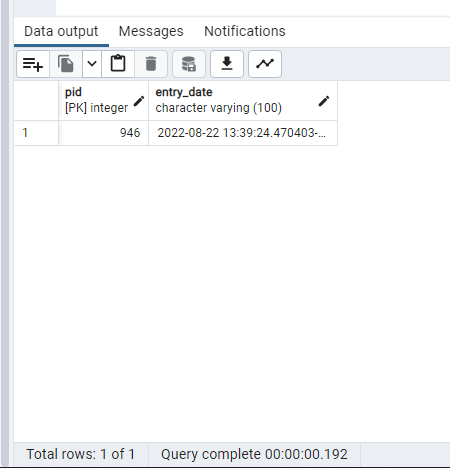
--Checking for Newly added record

SELECT \* FROM public."Patients" ORDER BY "Patient\_ID" DESC LIMIT 5

--Checking Trigger Log

SELECT \* FROM public.audit\_patients

--After inserting records Into Patients, trigger is executed and created record in Log table “audit\_patients”



-- 12.Write a query to find the ProviderName and Provider Speciality for PS\_ID ='PSID02'

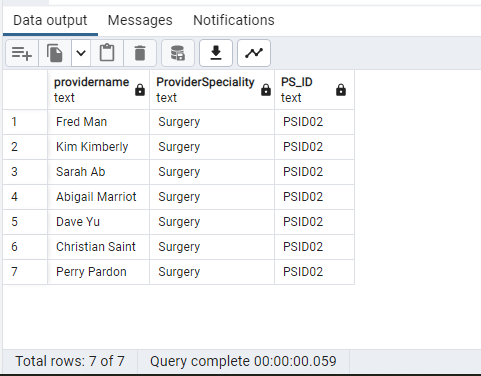
Query:

SELECT PV."ProviderName" ProviderName,PS."ProviderSpeciality",PS."PS\_ID"

FROM "ProviderSpeciality" PS JOIN "Providers" PV

ON PV."PS\_ID" = PS."PS\_ID"

WHERE PV."PS\_ID"='PSID02'

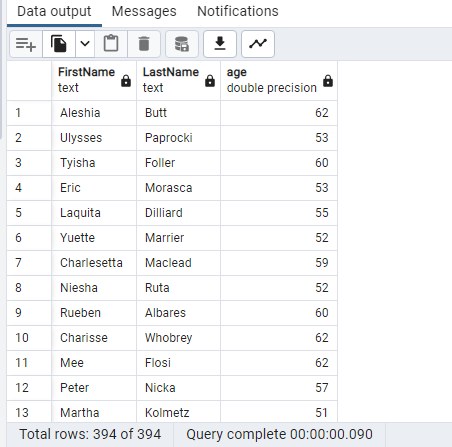


-- 13. Display the patient names and ages whose age is more than 50 years

Query:

SELECT "FirstName","Patients"."LastName",date\_part('year',age("Patients"."DateOfBirth")) as Age

FROM "Patients" WHERE (date\_part('year',age("Patients"."DateOfBirth"))) >50



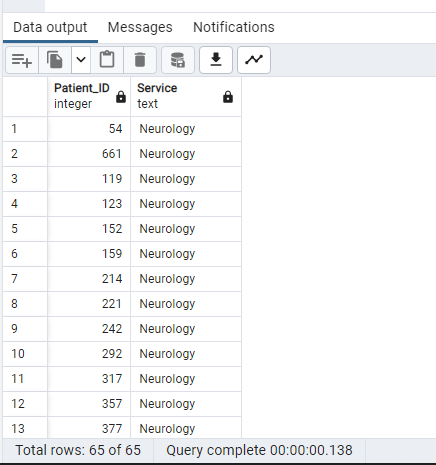
-- 14.Write a query to get list of patient ID's and service whose are in service as 'Neurology'

Query:

SELECT RG."Patient\_ID",S."Service" FROM "ReAdmissionRegistry" RG JOIN "Service" AS S

ON RG."Service\_ID" = S."Service\_ID"

WHERE "Service" = 'Neurology'



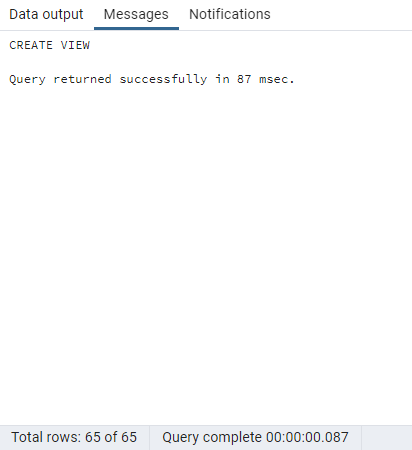
-- 15.Create view ONtable Provider table ONcolumns ProviderName and Provider\_ID

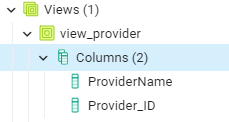
Query:

CREATE VIEW View\_Provider("ProviderName","Provider\_ID")

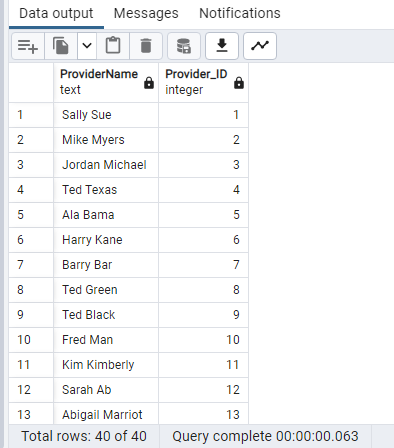
AS

SELECT "ProviderName","Provider\_ID" FROM "Providers"





SELECT \* FROM public.view\_provider

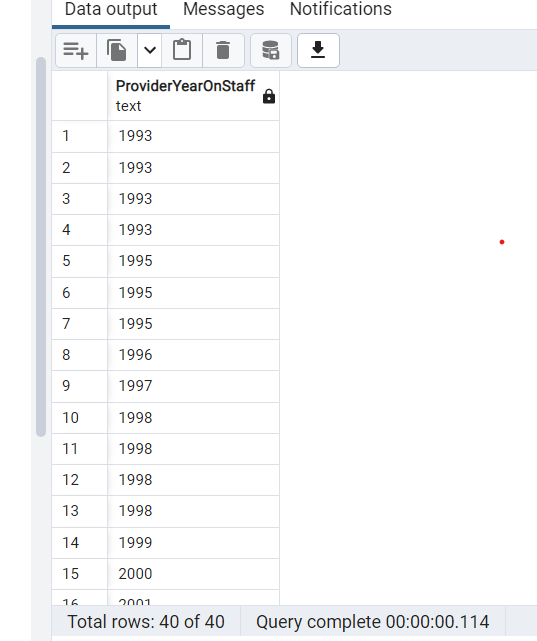


--16. Write a query to Extract Year FROM ProviderDateOnStaf

Query:

SELECT to\_char("ProviderDateOnStaff",'YYYY') as "ProviderYearOnStaff"

FROM public."Providers"



--17. Write a query to get unique Patient\_ID,race and Language of patients

whose race is White and also speak English.

Query:

SELECT Distinct("Patient\_ID"),"Race","Language"

FROM public."Patients" as Pt

JOIN public."Race" as R

ON Pt."Race\_ID" = R."Race\_ID"

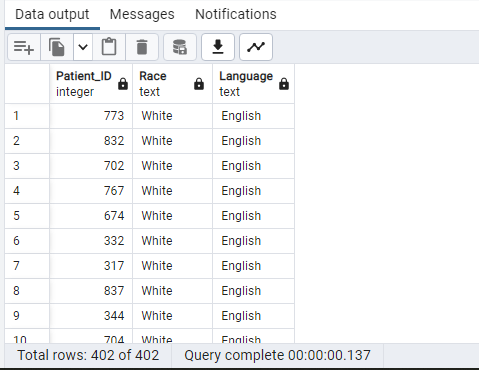
JOIN public."Language" as L

ON Pt."Language\_ID" = L."Language\_ID"

WHERE 1=1

And R."Race" = 'White'

And L."Language" = 'English'



--Q18. Get list of patient ID's whose service was 'Cardiology' and discharged to

'Home'

Query:

SELECT "Patient\_ID","Service","DischargeDisposition"

FROM "ReAdmissionRegistry" RA JOIN "DischargeDisposition" DD

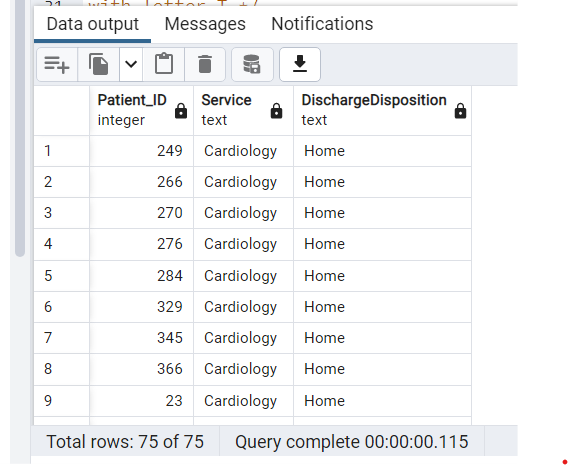
ON RA."Discharge\_ID"= DD."Discharge\_ID"

JOIN "Service" S

ON RA."Service\_ID" = S."Service\_ID"

WHERE DD."DischargeDisposition" = 'Home'

AND S."Service"='Cardiology'



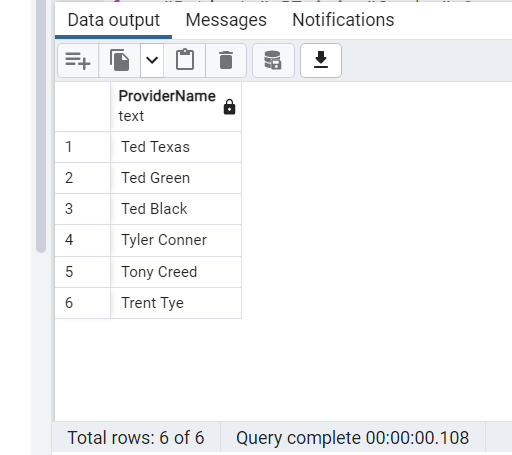
--Q19. Write a query to get list of Provider names whose Providername is starting

with letter

Query:

SELECT "ProviderName" FROM public."Providers"

WHERE "ProviderName" like 'T%'



--Q20. List female patients over the age of 40 who have undergone surgery from

January-March 2019

Query:

SELECT "FirstName" || ' ' || "LastName" as "PatientName"

,substring(G."Gender",1,1) as "Gender"

, EXTRACT(YEAR FROM AGE(CURRENT\_DATE, Pt."DateOfBirth")) as "Age"

, AV."DateofVisit"

FROM "Patients" PT JOIN "Gender" G

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

JOIN "AmbulatoryVisits" AV

ON PT."Patient\_ID" = AV."Patient\_ID"

JOIN "Providers" P

ON P."Provider\_ID" = AV."Provider\_ID"

JOIN "ProviderSpeciality" PS

ON PS."PS\_ID"= P."PS\_ID"

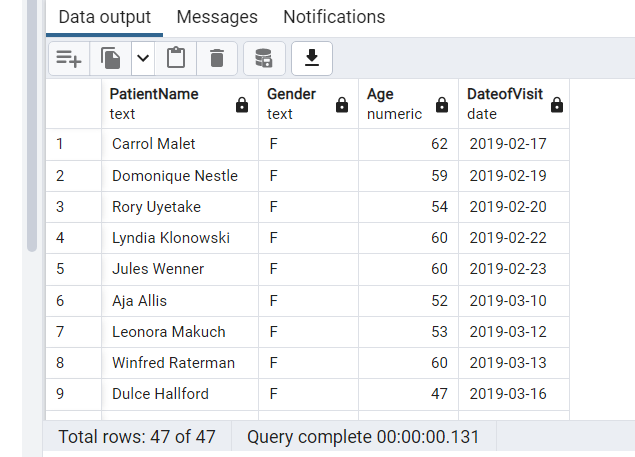
WHERE

G."Gender" = 'Female' AND

EXTRACT(YEAR FROM AGE(CURRENT\_DATE, Pt."DateOfBirth")) > 40 AND

PS."ProviderSpeciality" = 'Surgery' AND

AV."DateofVisit" between to\_date('20190101','YYYYMMDD') AND to\_date('20190331','YYYYMMDD')



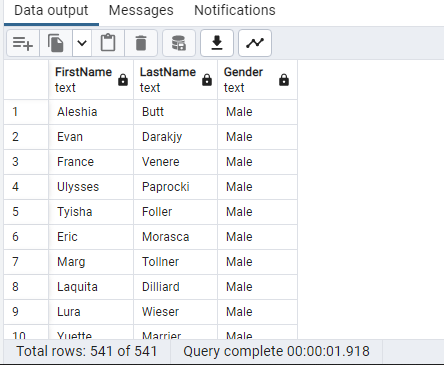
--21. Write a Query to get list of Male patients.--

Query:

SELECT p."FirstName",p."LastName",g."Gender" FROM public."Patients" as p

JOIN public."Gender" as g

ON p."Gender\_ID"= g."Gender\_ID" WHERE g."Gender"='Male';



--22. Write a query to get list of patient ID's who has discharged to home--

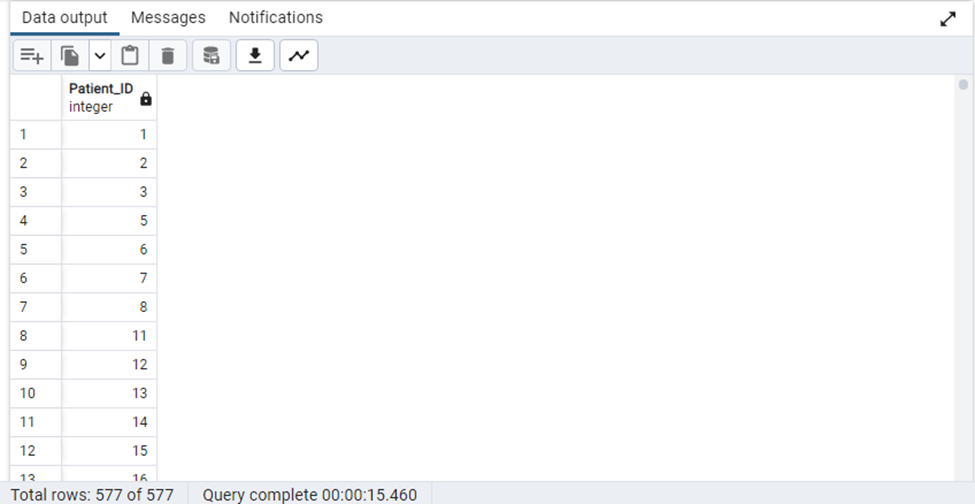
Query:

SELECT "Patient\_ID" FROM public."Discharges" as d

JOIN public."DischargeDisposition" as dd

ON d."Discharge\_ID"= dd."Discharge\_ID"

WHERE dd."DischargeDisposition"='Home';



--23. Find the category of illness(Stomach Ache or Migraine) that has maximum

number of patients--

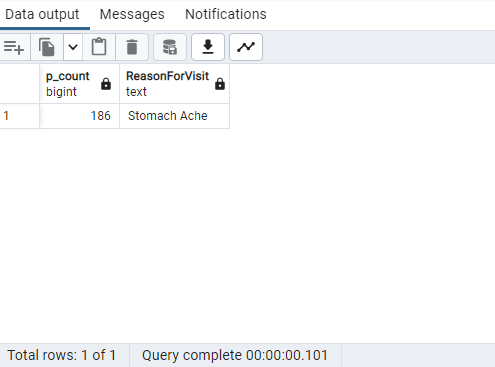
Query:

SELECT COUNT(edv."Patient\_ID") AS P\_Count,rfv."ReasonForVisit" FROM public."EDVisits" as edv

JOIN public."ReasonForVisit" as rfv ONedv."Rsv\_ID"=rfv."Rsv\_ID"

WHERE "ReasonForVisit"='Stomach Ache' OR "ReasonForVisit"='Migraine'

GROUP BY rfv."ReasonForVisit" ORDER BY P\_Count DESC LIMIT 1



--24. Write a query to get list of New Patient ID's.—

Query:

SELECT av."Patient\_ID" FROM public."AmbulatoryVisits" as av

JOIN public."VisitTypes" as vt

ON av."AMVT\_ID"= vt."AMVT\_ID"

WHERE vt."VisitType"='New';



--25. Create trigger on table Readmission registry—

Query:

SELECT \* FROM public."ReAdmissionRegistry"

INSERT INTO "ReAdmissionRegistry" VALUES (1000,1000,'2018-05-26','2018-05-26 20:40:34','DID01','SID01','PD001','12','0.23',1,3,1)

--creating temporary table--

CREATE TABLE temp\_readm(

"Entry\_Date" VARCHAR(100) NOT NULL

)

--creating Trigger Function--

CREATE OR REPLACE FUNCTION Del()

RETURNS TRIGGER AS $NEW$

BEGIN

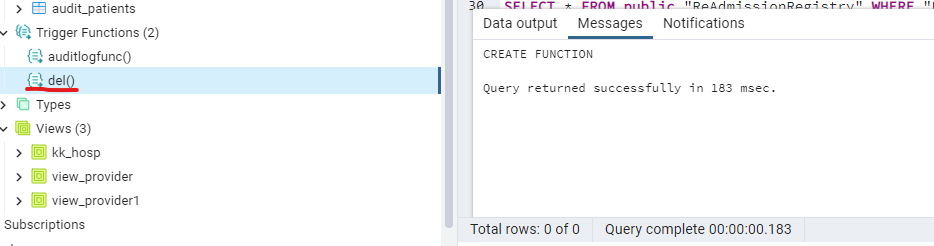
INSERT INTO temp\_readm("Entry\_Date")

VALUES(CURRENT\_TIMESTAMP);

RETURN NEW;

END;

$NEW$ LANGUAGE plpgsql;



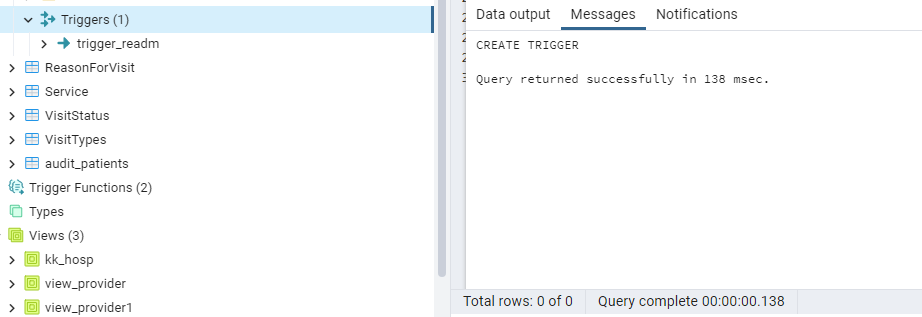
--creating trigger named Trigger\_readm--

CREATE TRIGGER Trigger\_readm

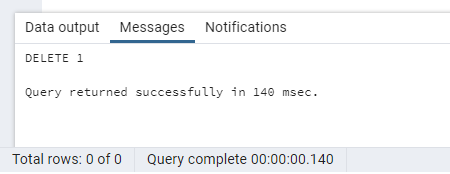
AFTER DELETE ON "ReAdmissionRegistry"

FOR EACH ROW

EXECUTE PROCEDURE Del();

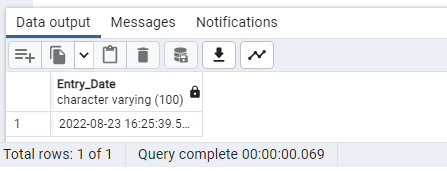


--Deleting a Record From ReAdmissionRegistry to execute Trigger



--Temp Table got updated with the time the record got deleted

SELECT \* FROM temp\_readm



----26. Select all providers with a name starting 'h' followed by any character ,

--followed by 'r', followed by any character, followed by 'y

Query:

SELECT "ProviderName"

FROM "Providers"

WHERE "ProviderName" LIKE 'H\_r\_y%'

Graphical user interface, application, Word

Description automatically generated

--27. Show the list of the patients who have cancelled their appointment--

Query:

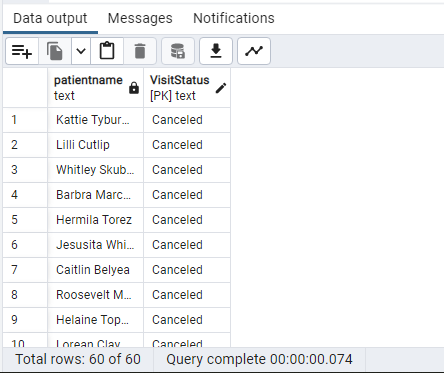
SELECT P."FirstName" || ' ' || P."LastName" AS PatientName ,"VisitStatus"

FROM "Patients" P

JOIN "AmbulatoryVisits" AV ON P."Patient\_ID" = AV."Patient\_ID"

JOIN "VisitStatus" VS ON AV."VisitStatus\_ID" = VS."VisitStatus\_ID"

WHERE VS."VisitStatus" = 'Canceled'



--28. Write a query to get list of ProviderName's with a name starting 'ted--

Query:

SELECT "ProviderName"

FROM "Providers"

WHERE "ProviderName" LIKE 'Ted%'

Graphical user interface, application, table, Word

Description automatically generated

--29. Create a view without using any schema or table and check the created view using SELECT statement--

Query:

CREATE VIEW BPLimit\_view AS

SELECT text 'BPS>130 & BPD>80' AS High\_BP,

text 'BPS<90 & BPD<60' AS Low\_BP

Graphical user interface, text, application, Word

Description automatically generated

Text

Description automatically generated

//View created ONBP Limits without using table or schema. //

Query:

SELECT \* FROM BPLimit\_view

Graphical user interface, text, application, Word, Teams

Description automatically generated

--30. Write a query to get unique list of Patient Id's whose reason for visit is car accident--

Query:

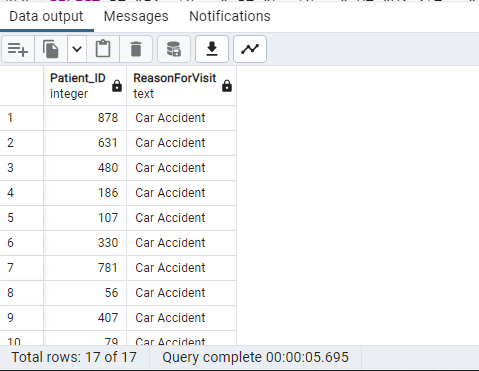
SELECT DISTINCT P."Patient\_ID","ReasonForVisit"

FROM "Patients" P

INNER JOIN "EDVisits" ED ON P."Patient\_ID" = ED."Patient\_ID"

INNER JOIN "ReasonForVisit" RV ON ED."Rsv\_ID" = RV."Rsv\_ID"

WHERE RV."ReasonForVisit" = 'Car Accident'



--31. Find which Visit type of patients are maximum in cancelling their appointment

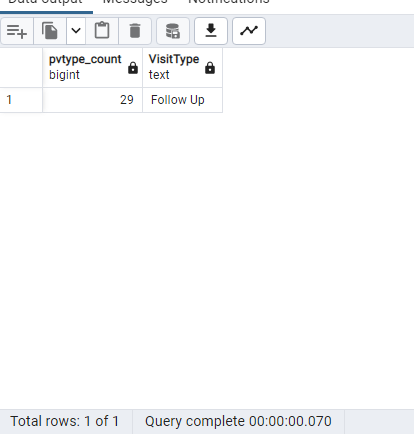
Query:

SELECT COUNT("Patient\_ID") AS PVType\_Count,"VisitType"

FROM "AmbulatoryVisits" AV JOIN "VisitTypes" V

ON AV."AMVT\_ID" = V."AMVT\_ID" JOIN "VisitStatus" VS ON AV."VisitStatus\_ID" = VS."VisitStatus\_ID"

WHERE "VisitStatus"='Canceled' GROUP BY "VisitType" ORDER BY PVType\_Count DESC LIMIT 1

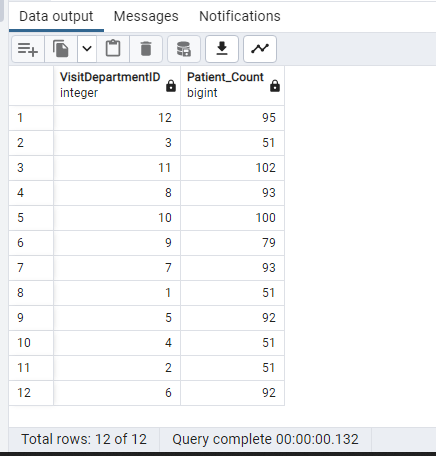


-- 32. Write a query to Count number of patients by VisitdepartmentID WHERE count greater than 50

Query:

SELECT "VisitDepartmentID",count("Patient\_ID") as "Patient\_Count" FROM "AmbulatoryVisits"

GROUP BY("VisitDepartmentID") HAVING count("Patient\_ID")> 50



-- 33. Write a query to get list of patient names whose visit type is new and visitdepartmentId is 2

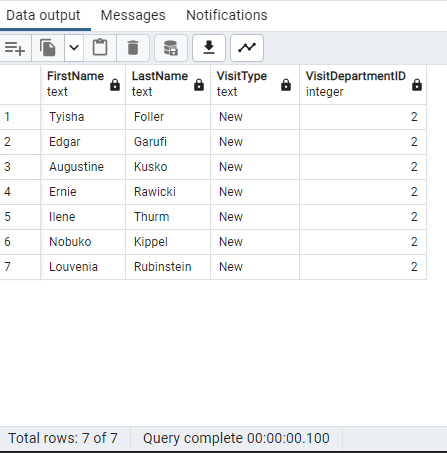
Query:

SELECT PT."FirstName",PT."LastName",VT."VisitType",AV."VisitDepartmentID" FROM "Patients" PT

JOIN "AmbulatoryVisits" AV ON PT."Patient\_ID" = AV."Patient\_ID"

JOIN "VisitTypes" VT ON VT."AMVT\_ID" = AV."AMVT\_ID"

WHERE "VisitDepartmentID" = 2 AND "VisitType"='New'



-- 34. Write a query to find the most commONreasons for hospital visit for patients between 50 and 60 years

Query:

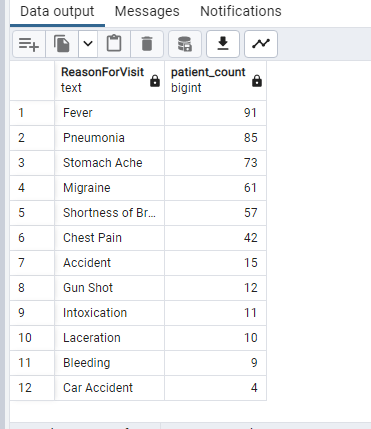
SELECT "ReasonForVisit" ,COUNT(P."Patient\_ID") AS Patient\_Count

FROM "Patients" P JOIN "EDVisits" EV ON EV."Patient\_ID" = P."Patient\_ID"

JOIN "ReasonForVisit" ON "ReasonForVisit"."Rsv\_ID" = EV."Rsv\_ID"

WHERE date\_part('year',age(P."DateOfBirth")) BETWEEN 50 AND 60

GROUP BY "ReasonForVisit" ORDER BY Patient\_Count DESC



-- 35. Get list of Patients whose gender is Male and who speak English and whose race is White

Query:

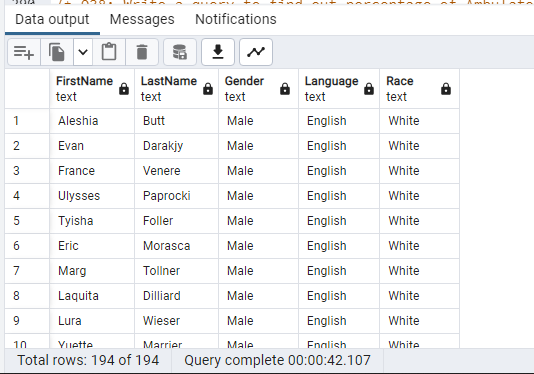
SELECT P."FirstName",P."LastName","Gender","Language","Race"

FROM "Patients" P JOIN "Gender" G ON G."Gender\_ID" = P."Gender\_ID"

JOIN "Language" L ON L."Language\_ID" = P."Language\_ID"

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

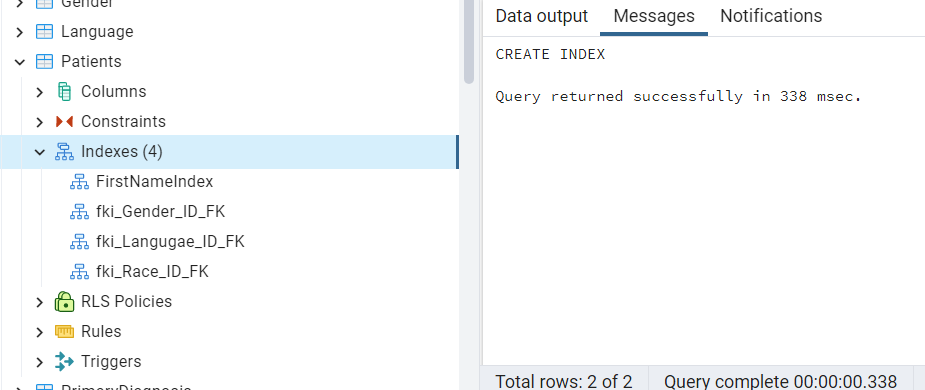
Where (G."Gender"='Male' AND L."Language" = 'English' AND R."Race" ='White')



--Q36: Create index ONPatient table

Query:

CREATE index "FirstNameIndex" ON"Patients"("FirstName")



--Q37: Write a query to get list of Provider ID's WHERE ProviderDateOnStaff year is

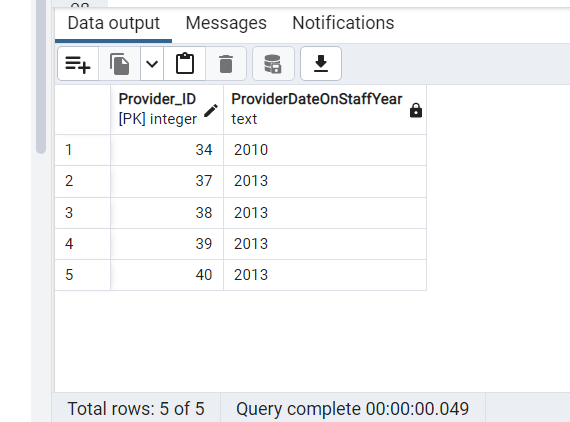
2013 and 2010

Query:

SELECT "Provider\_ID", to\_char("ProviderDateOnStaff",'YYYY') as "ProviderDateOnStaffYear"

FROM "Providers"

WHERE to\_char("ProviderDateOnStaff",'YYYY') in('2013','2010')



--Q38: Write a query to find out percentage of Ambulatory visits by visit type.\*/

//Using sub query

Query:

SELECT VT."VisitType" as "Visit\_Type"

,count(AV."Visit\_ID") "Visit\_Count"

,(SELECT count("Visit\_ID") FROM "AmbulatoryVisits") "Total\_Count"

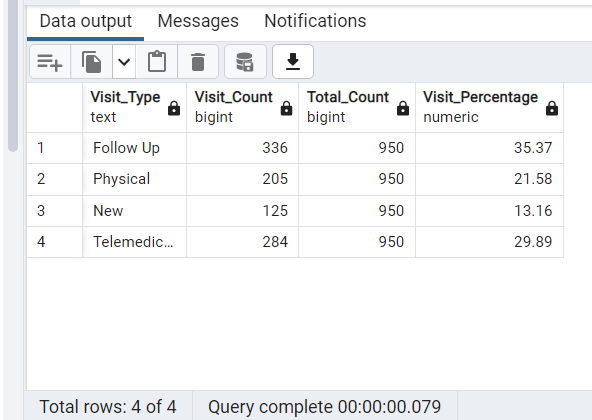
,round(count(AV."Visit\_ID")\*100.0/(SELECT count("Visit\_ID") FROM "AmbulatoryVisits"),2) as "Visit\_Percentage"

FROM "AmbulatoryVisits" AV

JOIN "VisitTypes" VT

ON AV."AMVT\_ID" = VT."AMVT\_ID"

Group By "VisitType"



--Q39: Write a query to get list of patient names who has discharged.

Query:

SELECT "FirstName" || ' ' || "LastName" as "DischargedPatientName" ,"DischargeDisposition"

FROM "Patients" PT JOIN "Discharges" D

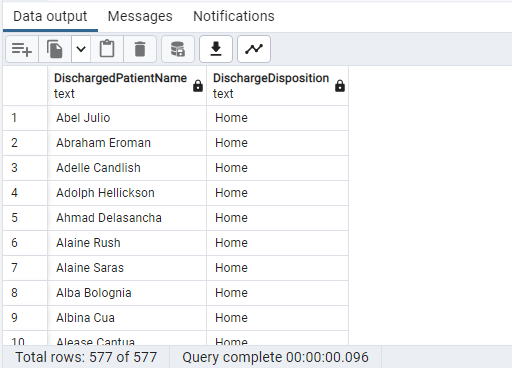
ON PT."Patient\_ID" = D."Patient\_ID"

JOIN "DischargeDisposition" DD

ON DD."Discharge\_ID" = D."Discharge\_ID"

WHERE DD."DischargeDisposition" = 'Home'

ORDER BY "DischargedPatientName"



.

--Q40: Create view on table EdVisit by selecting some columns and filter data

using Where condition

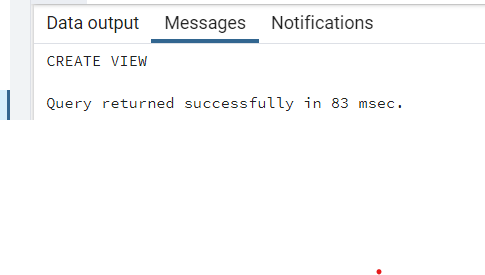
Query:

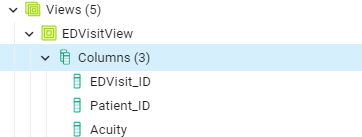
Create View "EDVisitView" as

SELECT "EDVisit\_ID", "Patient\_ID", "Acuity"

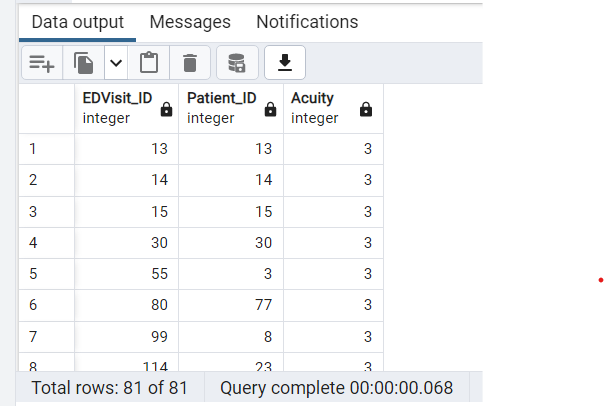
FROM "EDVisits"

WHERE "Acuity" = 3





SELECT \* FROM "EDVisitView"



--41. Get list of patient names whose primary diagnosis as 'Spinal Cord injury'

having Expected LOS is greater than 15--

Query:

SELECT p."FirstName",p."LastName",pd."PrimaryDiagnosis",rr."ExpectedLOS"

FROM public."Patients" as p

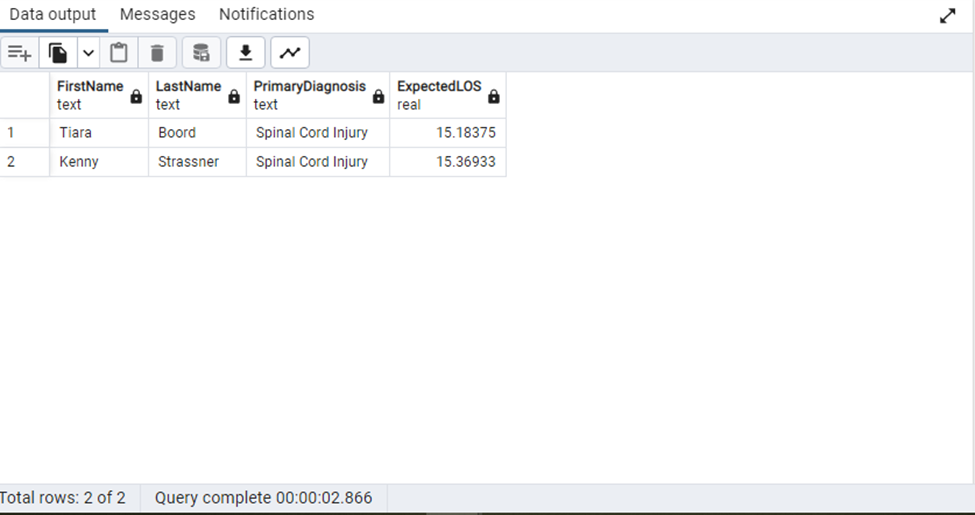
JOIN public."ReAdmissionRegistry" as rr

ON p."Patient\_ID"=rr."Patient\_ID"

inner JOIN public."PrimaryDiagnosis" as pd

ON rr."Diagnosis\_ID"=pd."Diagnosis\_ID" WHERE pd."PrimaryDiagnosis"='Spinal Cord Injury'

AND rr."ExpectedLOS" >'15';



--42. Write a query to get list of Patient names who haven't discharged—

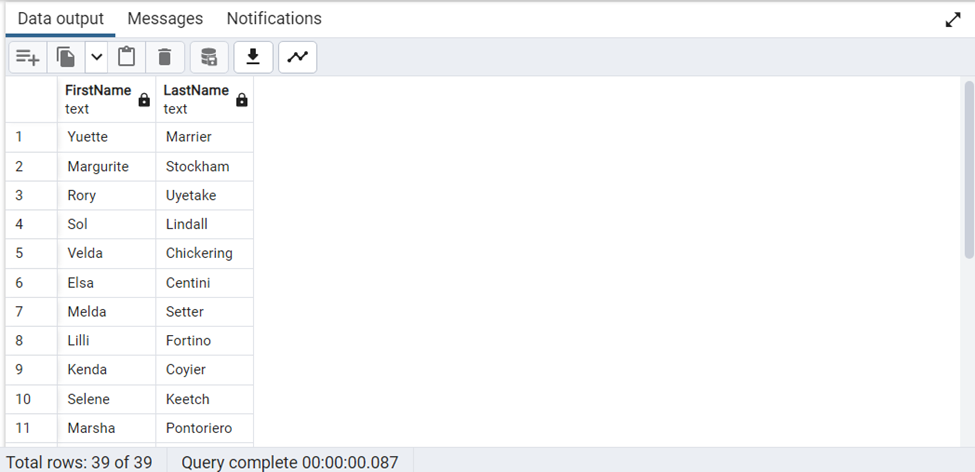
Query:

SELECT p."FirstName",p."LastName",dd."DischargeDisposition" FROM public."Patients" as p

JOIN public."Discharges" as d ON d."Patient\_ID"=p."Patient\_ID"

JOIN public."DischargeDisposition" as dd ON d."Discharge\_ID"= dd."Discharge\_ID"

WHERE dd."DischargeDisposition"='Transfer';



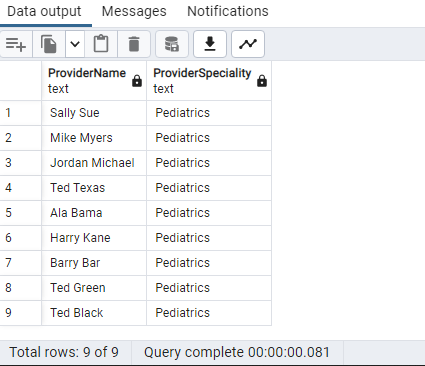
--43. Write a query to get list of Provider names whose ProviderSpecialty is Pediatrics.--

Query:

SELECT p."ProviderName",ps."ProviderSpeciality" FROM public."Providers" as p

JOIN public."ProviderSpeciality" as ps ON p."PS\_ID"=ps."PS\_ID"

WHERE ps."ProviderSpeciality"='Pediatrics';



--44. Write a query to get list of patient ID's who has admitted ON1/7/2018 and

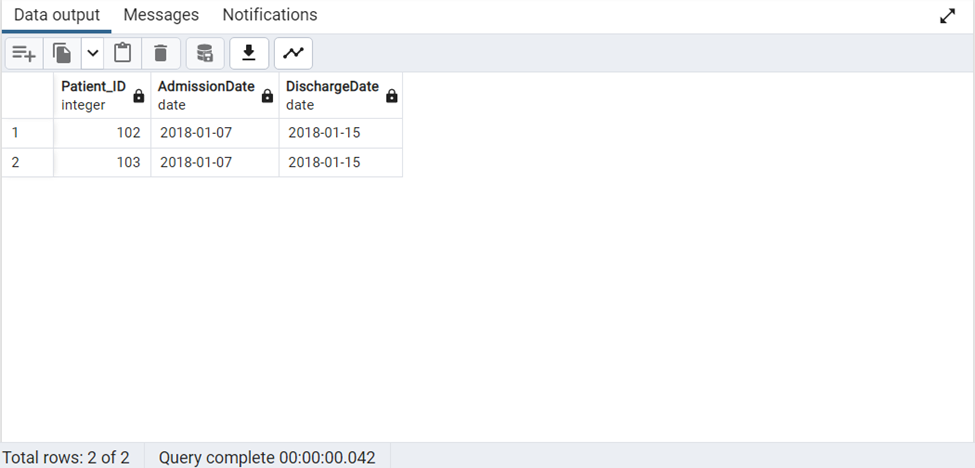
discharged on 1/15/2018--

Query:

SELECT d."Patient\_ID",d."AdmissionDate", cast(d."DischargeDate" as date )

FROM public."Discharges" as d

WHERE d."AdmissionDate" = '2018-01-07' and cast (d."DischargeDate" as date) = '2018-01-15 ' ;



--45. Write a query to find outpatients vs inpatients by monthwise (hint:

consider readmission/discharges and ambulatory visits table for inpatients

and outpatients)—

Query:

- CREATE extension tablefunc;

SELECT \* FROM

CROSSTAB(

$$

with cte as

(

SELECT count(av."Patient\_ID") counts,

to\_char(av."DateofVisit",'month')mns,date\_part('year',av."DateofVisit")yrs,

'outpatients' typ

FROM public."AmbulatoryVisits" av

GROUP BY mns,yrs

UNION ALL

SELECT count(rr."Patient\_ID") counts,

to\_char(rr."AdmissionDate",'month') mns,date\_part('year',rr."AdmissionDate")yrs,

'inpatients' typ

FROM public."ReAdmissionRegistry" rr

GROUP BY mns,yrs )

SELECT mns,yrs,typ,counts FROM

cte $$,

$$ values

( 'inpatients' ),

('outpatients')

$$)

AS (

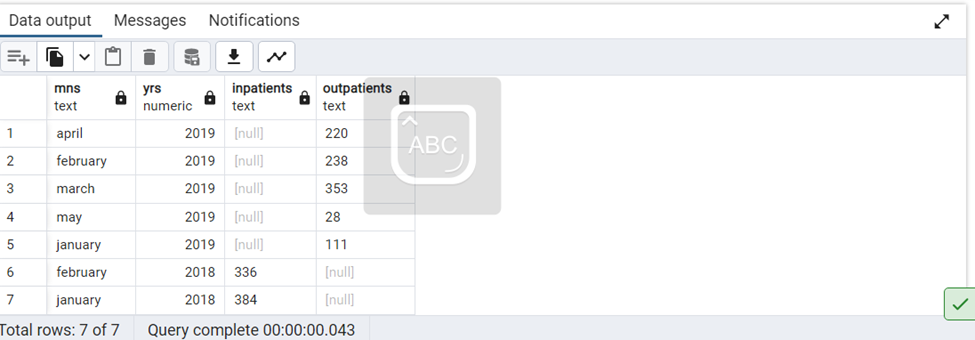
mns text,

yrs numeric,

inpatients numeric,

outpatients numeric

) ;



Note: Created a Common Table Expression(CTE) for Pivoting the table.

Used Create extension tablefunc for table function crosstab.

Used CROSSTAB Function for pivoting the table columns.

--46. Write a query to get list of Number of Ambulatory Visits by --

--Provider Speciality per month--

Query:

SELECT COUNT(AV."Visit\_ID")AS NumberOfAmbulatoryVisits,PS."ProviderSpeciality",

TO\_CHAR(AV."DateofVisit",'Month') AS Month,EXTRACT(year FROM AV."DateofVisit") AS Year

FROM (("AmbulatoryVisits" AV

JOIN "Providers" PR ON AV."Provider\_ID" = PR."Provider\_ID")

JOIN "ProviderSpeciality" PS ON PR."PS\_ID" = PS."PS\_ID")

GROUP BY PS."ProviderSpeciality",Month,Year

ORDER BY PS."ProviderSpeciality",Month,Year

Graphical user interface, application

Description automatically generated

--47.Write a query to find Average age for admissiONby service--

Query:

SELECT ROUND(AVG(EXTRACT('year' FROM AGE(P."DateOfBirth")))) AS Average\_Age, S."Service"

FROM (("Patients" P

JOIN "Discharges" D ON P."Patient\_ID" = D."Patient\_ID" )

JOIN "Service" S ON D."Service\_ID" = S."Service\_ID")

GROUP BY S."Service"

Graphical user interface, application, Word, Teams

Description automatically generated

--48. Write a query to get list of patient with their full names whose names contains "Ma"--

Query:

SELECT P."FirstName" || ' ' || P."LastName" AS PatientName

FROM "Patients" P

WHERE (P."FirstName" LIKE 'Ma%' OR P."FirstName" LIKE '%Ma') OR

(P."LastName" LIKE 'Ma%' OR P."LastName" LIKE '%Ma')

Graphical user interface, application, Word

Description automatically generated

--49. Update Visit Timestamp column in EDVisits table by selecting data type as--

--timestamp with timezone--

Query:

ALTER TABLE "EDVisits"

ALTER COLUMN "VisitTimestamp"

SET DATA TYPE timestamptz

Graphical user interface, text, application, Word

Description automatically generated

Table

Description automatically generated

--50. Create a trigger function on AmbulatoryVisits by selecting any two columns.--

--creating temporary table--

Query:

CREATE TABLE ambupdate(

"Pat\_ID" INTEGER NOT NULL,

"VisitDate" DATE,

"ScheduledDate" TIMESTAMP

)

//Creating a table ambupdate, containing three columns Patient ID, VisitDateD and Scheduled date. This table gets updated, whenever DateofVisit and DateScheduled FROM AmbulatoryVisits gets updated.//

TRIGGER ON UPDATE

--creating Trigger Function--

CREATE OR REPLACE FUNCTION update()

RETURNS TRIGGER AS $NEW$

BEGIN

INSERT INTO ambupdate("Pat\_ID","VisitDate","ScheduledDate")

VALUES(NEW."Patient\_ID",NEW."DateofVisit",NEW."DateScheduled");

RETURN NEW;

END;

$NEW$ LANGUAGE plpgsql;

//update() is the trigger functiONcreated.//

--creating trigger named update\_ambulatory--

CREATE TRIGGER update\_ambulatory

AFTER UPDATE ON "AmbulatoryVisits"

FOR EACH ROW

EXECUTE PROCEDURE update();

//Whenever the DateofVisit and DateScheduled FROM AmbulatoryVisits table gets updated, trigger update\_ambulatory calls ONtrigger functiONupdate() to insert the updated column values into temp table ambupdate.//

--update--

UPDATE "AmbulatoryVisits"

SET "DateofVisit" = '2019-01-15' , "DateScheduled" = '2018-12-30 9:18:24'

WHERE "Patient\_ID" = 5

//Screenshot of trigger and trigger function.//

Graphical user interface, text, application

Description automatically generated with medium confidence

Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

//Screenshot of table ambupdate with the updated record.//

Graphical user interface, application, Teams

Description automatically generated

-- 51. Insert number of days for ReadmissiONin DaysToReadmissiONColumn for patient ID's FROM 737 to 742

QUERY :

UPDATE "ReAdmissionRegistry" SET "DaysToReadmission" ='6' WHERE "Patient\_ID" ='737'

UPDATE "ReAdmissionRegistry" SET "DaysToReadmission" ='8' WHERE "Patient\_ID" ='738'

UPDATE "ReAdmissionRegistry" SET "DaysToReadmission" ='9' WHERE "Patient\_ID" ='739'

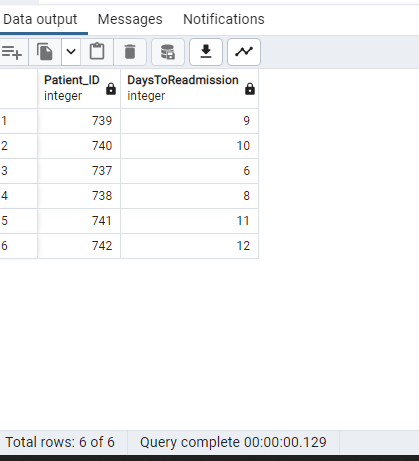
UPDATE "ReAdmissionRegistry" SET "DaysToReadmission" ='10' WHERE "Patient\_ID" ='740'

UPDATE "ReAdmissionRegistry" SET "DaysToReadmission" ='11' WHERE "Patient\_ID" ='741'

UPDATE "ReAdmissionRegistry" SET "DaysToReadmission" ='12' WHERE "Patient\_ID" ='742'

//Checking the Updated Records

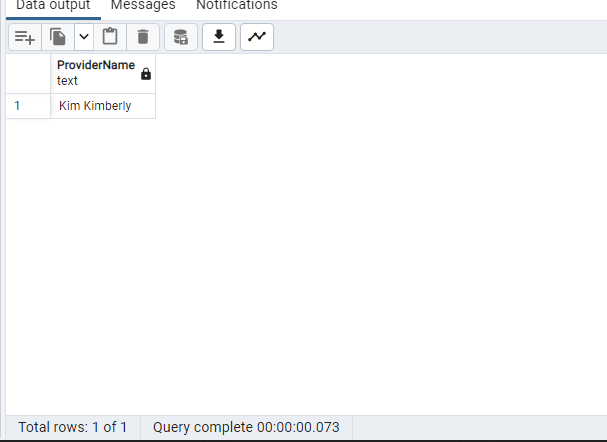
SELECT "Patient\_ID","DaysToReadmission" FROM public."ReAdmissionRegistry" WHERE "Patient\_ID" BETWEEN '737' and '742'



-- 52. Get list of Provider names whose name is starting with K and ending with y(Hint:K-Upper, Y-Lower)

Query :

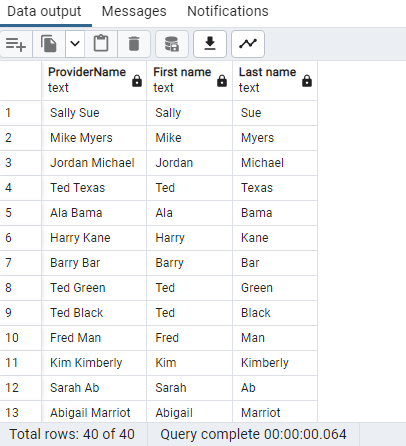
SELECT "ProviderName" FROM "Providers" WHERE "ProviderName" LIKE 'K%y'



-- 53. Write a query to Split provider First name and Last name into different column

Query :

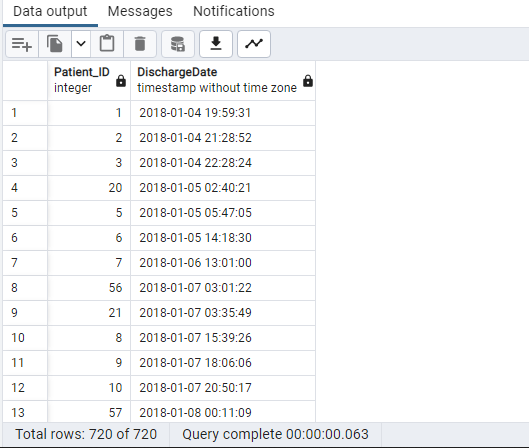
SELECT "Providers"."ProviderName",SPLIT\_PART("ProviderName",' ',1) AS "First name" , SPLIT\_PART("ProviderName",' ',2) AS "Last name" FROM "Providers"



-- 54. Get list of Patient ID's ORDER BY Discharge date

Query:

SELECT "Patient\_ID","DischargeDate" FROM "Discharges" ORDER BY "DischargeDate"



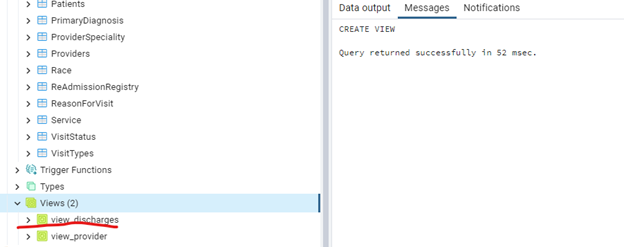
-- 55. Write a query to drop View by creating view ONtable Discharge by selecting columns

//Created View View\_Discharges

CREATE VIEW View\_Discharges("ExpectedLOS","Discharge\_ID")

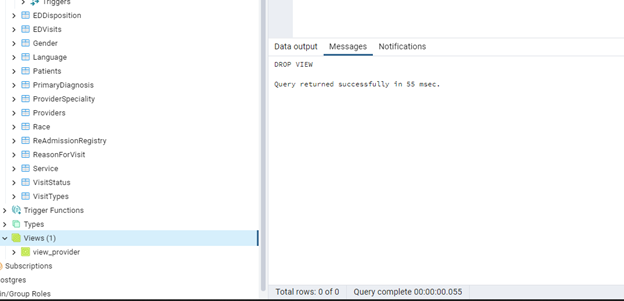
AS

SELECT "ExpectedLOS","Discharge\_ID" FROM "Discharges"



//Dropping View

DROP VIEW view\_discharges



--Q56: Write a query to get list of Patient ID's WHERE Visitdepartment ID is 1 and

BloodPressureSystolic is between 123 to 133 \*/

Query:

SELECT PT."Patient\_ID", AV."VisitDepartmentID", AV."BloodPressureSystolic"

FROM "Patients" PT JOIN "AmbulatoryVisits" AV

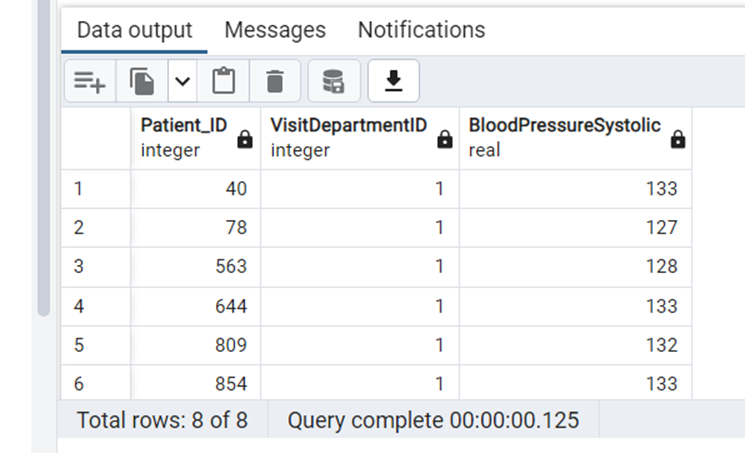
ON PT."Patient\_ID"= AV."Patient\_ID"

where

"VisitDepartmentID" = 1

And

AV."BloodPressureSystolic" between 123 And 133

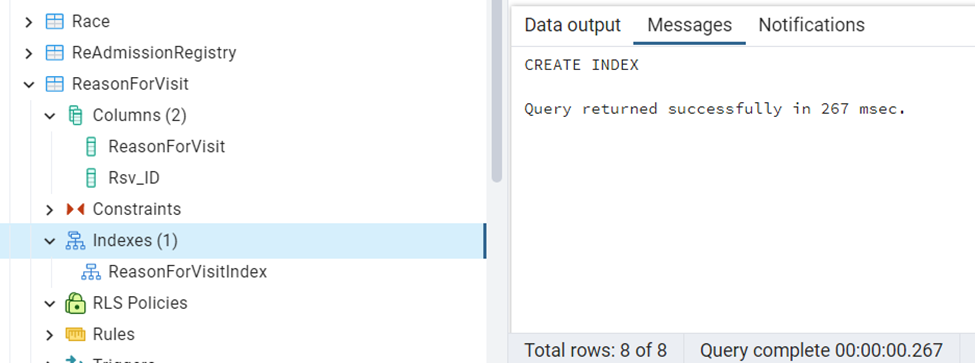


--Q57: Write the query to CREATE Index ONtable ReasonForVisit by selecting a

column and also write the query drop same index\*/

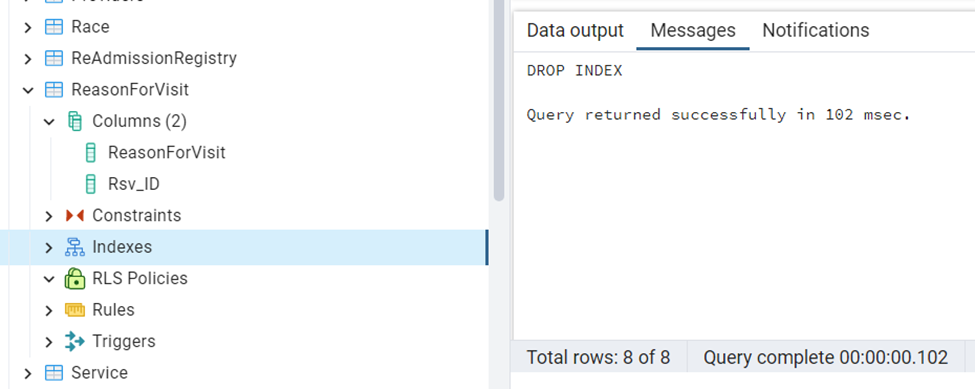
Query :

CREATE Index "ReasonForVisitIndex" ON"ReasonForVisit"("ReasonForVisit")



// Query to drop the index

DROP INDEX "ReasonForVisitIndex"



--Q58: Write a query to Count number of unique patients EDDispositiONwise.\*/

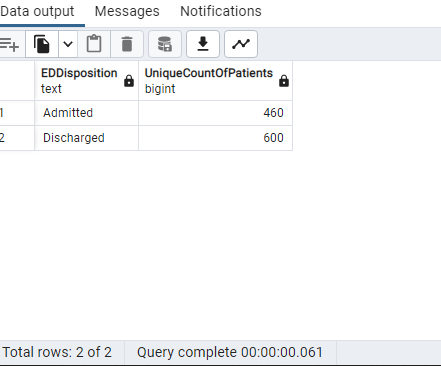
Query:

SELECT EDD."EDDisposition", count(distinct("Patient\_ID")) as "UniqueCountOfPatients"

FROM "EDVisits" EDV JOIN "EDDisposition" EDD

ON EDV."EDD\_ID"= EDD."EDD\_ID"

group by EDD."EDDisposition"



--Q59: Write a query to get list of Patient ID's WHERE Visitdepartment ID is 5 or

BloodPressureSystolic is NOT NULL

Query:

SELECT PT."Patient\_ID", AV."VisitDepartmentID", AV."BloodPressureSystolic"

FROM "AmbulatoryVisits" AV JOIN "Patients" PT

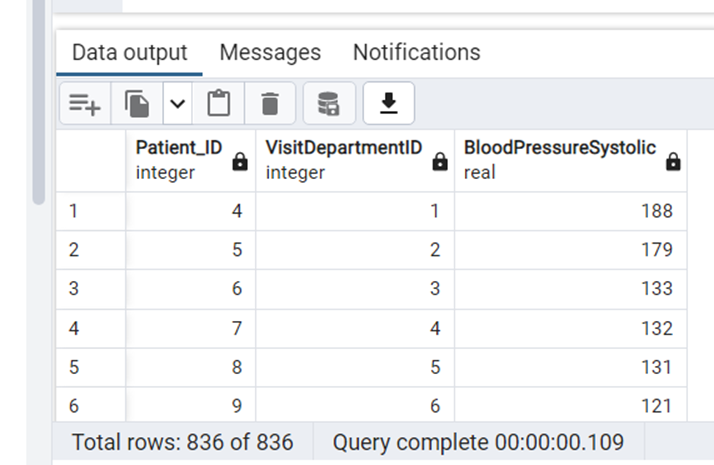
ON PT."Patient\_ID" = AV."Patient\_ID"

WHERE

AV."VisitDepartmentID" = 5

OR

AV."BloodPressureSystolic" IS NOT NULL



--Q60: Query to find the number of patients readmitted by Service

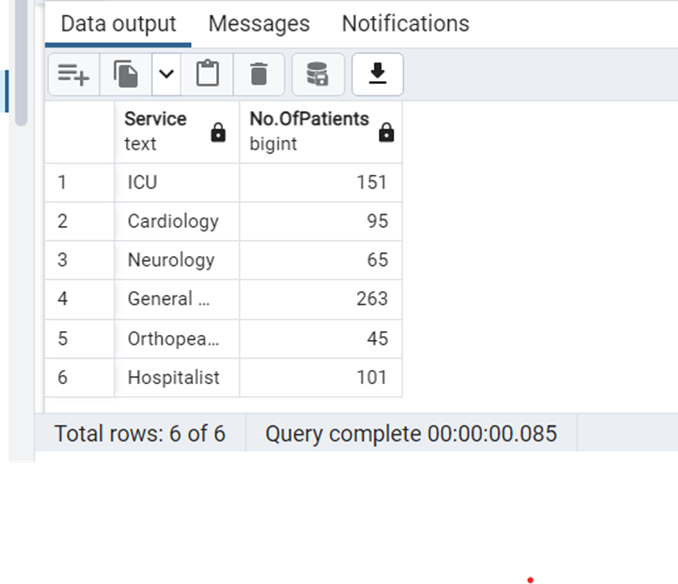
Query:

SELECT S."Service", count(RA."Patient\_ID") as "No.OfPatients"

FROM "Service" S JOIN "ReAdmissionRegistry" RA

ON S."Service\_ID" = RA."Service\_ID"

group by "Service"



--61. 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 BloodPressureSystolic above

120 and BloodPressureDiastolic above 80--

Query:

SELECT p."Patient\_ID",p."FirstName",p."LastName",

date\_part('year',age("DateOfBirth")) as Patient\_Age , "Gender" , "BloodPressureSystolic","BloodPressureDiastolic"

FROM public."Patients" as p

JOIN public."Gender" as g

ON p."Gender\_ID"= g."Gender\_ID"

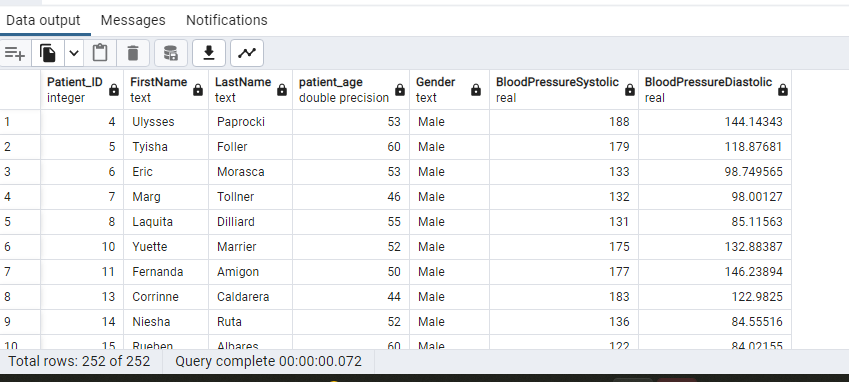
JOIN public."AmbulatoryVisits" as av

ON p."Patient\_ID"=av."Patient\_ID"

WHERE g."Gender"='Male' and

av."BloodPressureSystolic">'120' and av."BloodPressureDiastolic">'80' and

date\_part('year',age("DateOfBirth")) between 40 and 60 ;



--62. Query to find the number of outpatients who have visited month wise(use month names)--

Query:

select count(p."Patient\_ID") AS OutPatients\_Count,to\_char(av."DateofVisit",'month') month\_visit,date\_part('year',av."DateofVisit") year\_visit

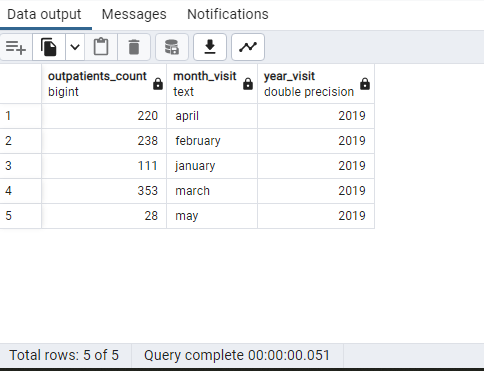
FROM public."Patients" p

inner JOIN public."AmbulatoryVisits" av

ON p."Patient\_ID"= av."Patient\_ID"

group by to\_char(av."DateofVisit",'month'), date\_part('year',av."DateofVisit")

ORDER BY to\_char(av."DateofVisit",'month');



--63. Write a query to get list of patient ID's whose BloodPressureSystolic is

131,137,138--

Query:

SELECT av."Patient\_ID",av."BloodPressureSystolic" FROM public."AmbulatoryVisits" as av

WHERE av."BloodPressureSystolic" in ('131','137','138');

with cte as

(

SELECT count(av."Patient\_ID") counts,

to\_char(av."DateofVisit",'month')month\_visit,date\_part('year',av."DateofVisit")yrs,

'outpatients' type

FROM public."AmbulatoryVisits" av

GROUP BY month\_visit,yrs

UNION ALL

SELECT count(rr."Patient\_ID") counts,

to\_char(rr."AdmissionDate",'month') month\_visit,date\_part('year',rr."AdmissionDate")yrs,

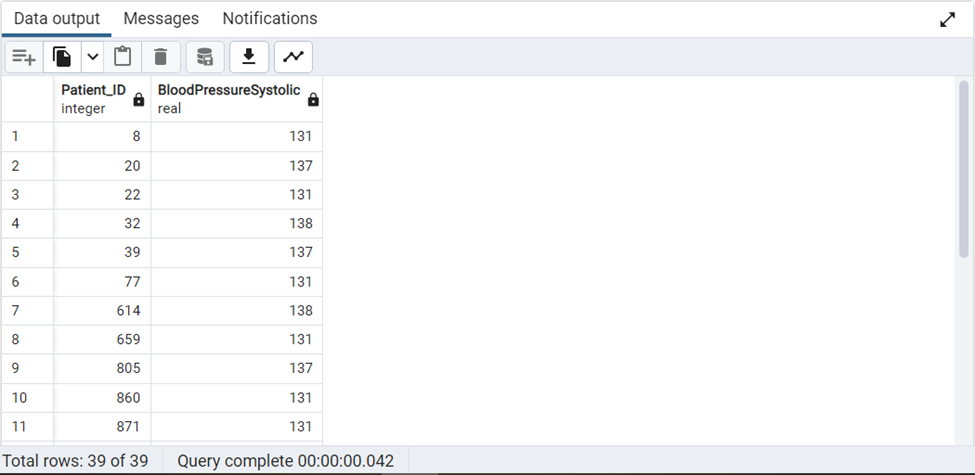
'inpatients' type

FROM public."ReAdmissionRegistry" rr

--WHERE av."Patient\_ID"= rr."Patient\_ID"

GROUP BY month\_visit,yrs

)



Note: Used Common Table Expression for combining 2 queries.

--64. Query to classify expected LOS into 3 categories as per the duration. (Hint:

Use of CASE statement)--

Query:

SELECT "Patient\_ID","ExpectedLOS",

case

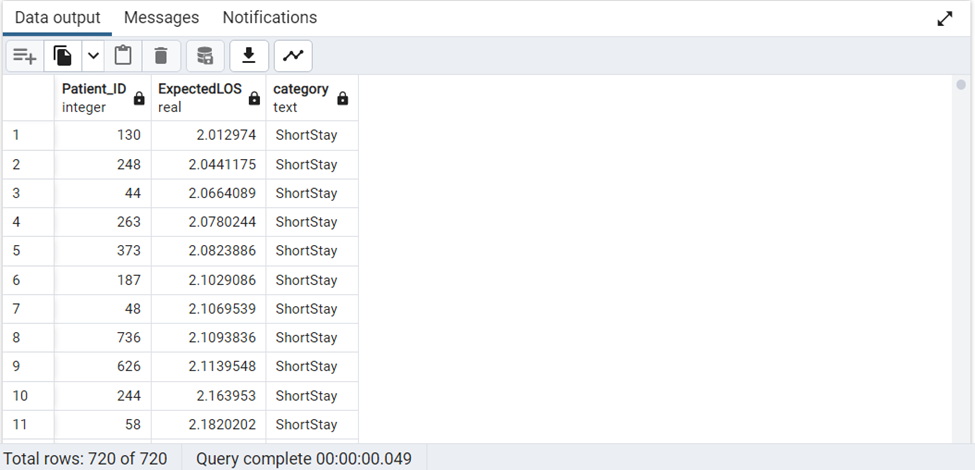
when "ExpectedLOS" <'2' then 'NormalStay'

when "ExpectedLOS" >'2' and "ExpectedLOS"<'5' then 'ShortStay'

when "ExpectedLOS" >'5' then 'LongStay'

else 'NoStay' end category

FROM public."Discharges" ORDER BY "ExpectedLOS";



--65. Write a query to CREATE a table to list the names of patients whose date of

birth is later than 1st jan 1960.Name the table as “Persons”--

Query:

--Creating Table "Persons"

SELECT "FirstName","LastName","DateOfBirth"

into "Persons" /\*Temptable\*/

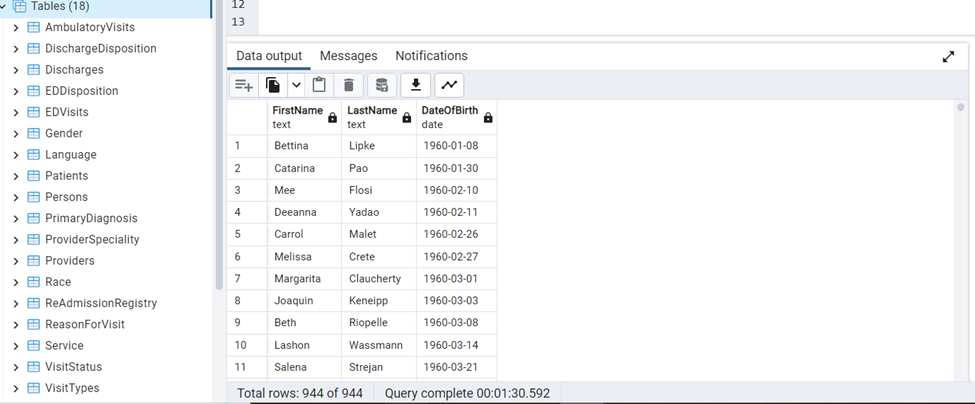
FROM public."Patients"

WHERE "DateOfBirth" > '1960-01-01'

ORDER BY "DateOfBirth";

--Checking Table "Persons"

SELECT \* FROM "Persons"



--66. Write a query to Count number of patients who has discharged after march3rd 2018

Query:

SELECT COUNT(D."Patient\_ID")

FROM "Discharges" D

WHERE D."DischargeDate" > '2018-03-03 23:59:59'

Text

Description automatically generated with low confidence

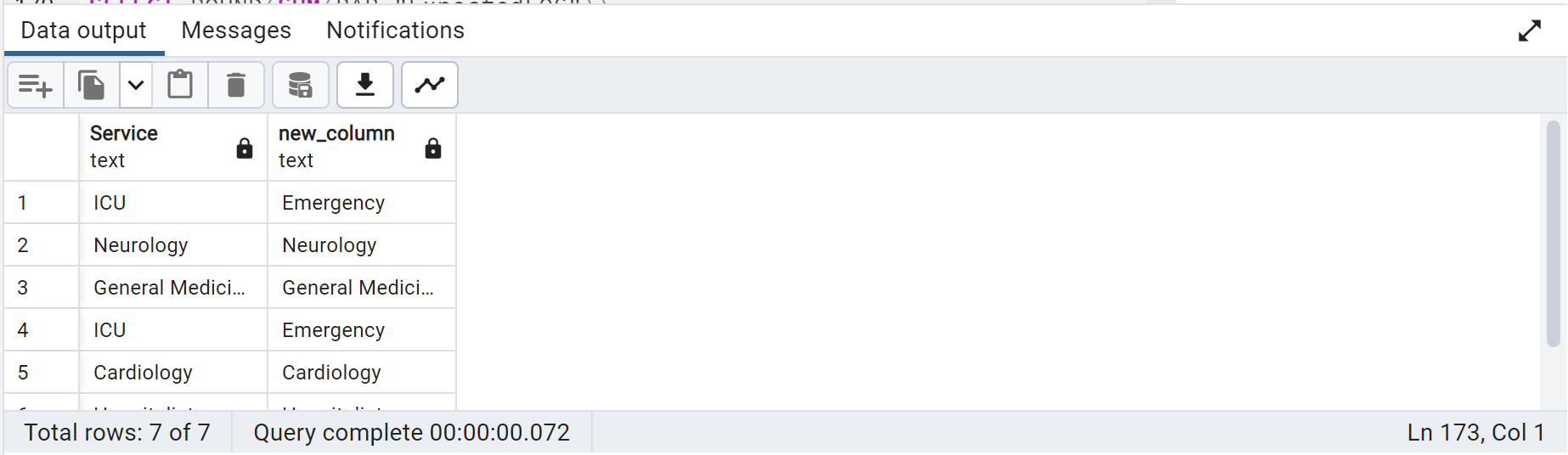
--67. Replace ICU with emergency (Hint: Do not update or alter the table)—

Query:

SELECT S."Service" ,

REPLACE(S."Service", 'ICU','Emergency') AS New\_Column

FROM "Service" S



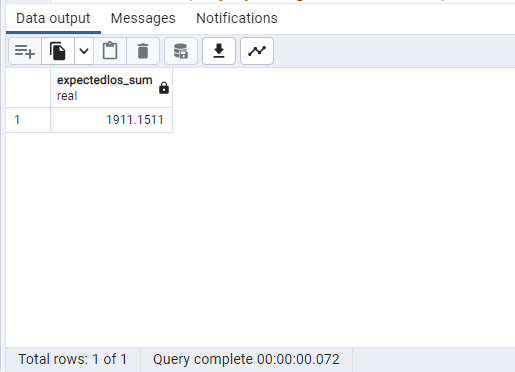
--68. Write a query to get Sum of ExpectedLOS for Service\_ID 'SID01'—

Query:

SELECT SUM("ExpectedLOS") AS ExpectedLOS\_Sum

FROM "ReAdmissionRegistry"

WHERE "Service\_ID" = 'SID01'



--69. Create index ONtable Provider by selecting a column and filter by using WHERE condition

Query:

CREATE INDEX psid\_idx

ON "Providers"("PS\_ID")

WHERE "ProviderDateOnStaff" > '1999-12-31'

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

--70. List down all triggers in our HealthDB database

Query:

SELECT event\_object\_table AS table\_name,trigger\_name

FROM information\_schema.triggers

GROUP BY table\_name,trigger\_name

ORDER BY table\_name,trigger\_name

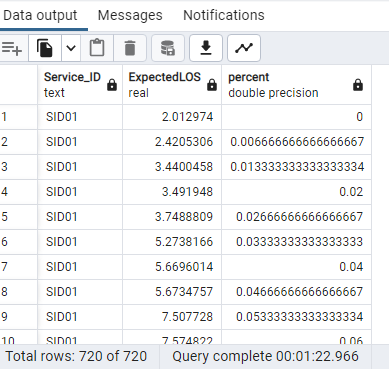
Graphical user interface, application, Word, Teams

Description automatically generated

-- 71. Partition the table according to Service\_ID and use windows function to calculate percent rank. Order by ExpectedLOS

Query :

SELECT "Service\_ID","ExpectedLOS",PERCENT\_RANK() OVER ( PARTITION BY "Service\_ID" ORDER BY "ExpectedLOS") AS Percent FROM "Discharges"



-- 72. Write a query by using commONtable expressions and case statements to display birthyear ranges

Query :

WITH DOBYears AS

(

Select

"Patient\_ID",

"DateOfBirth",

EXTRACT('Year' FROM "DateOfBirth") AS DOBYear

FROM public."Patients"

)

SELECT "Patient\_ID", "DateOfBirth",

CASE

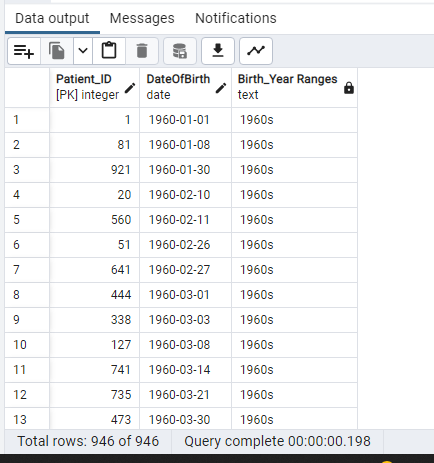
WHEN DOBYear BETWEEN 1960 AND 1969 THEN '1960s'

WHEN DOBYear BETWEEN 1970 AND 1979 THEN '1970s'

WHEN DOBYear BETWEEN 1980 AND 1989 THEN '1980s'

END AS "Birth\_Year Ranges"

FROM DOBYears ORDER BY "DateOfBirth" ASC

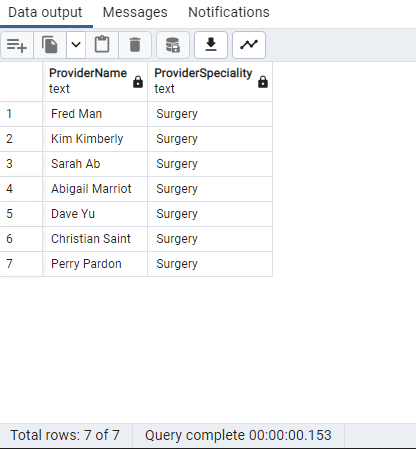


-- 73. Get list of Provider names whose ProviderSpeciality is Surgery

Query :

SELECT "ProviderName" , "ProviderSpeciality" FROM "Providers" P JOIN "ProviderSpeciality" PS

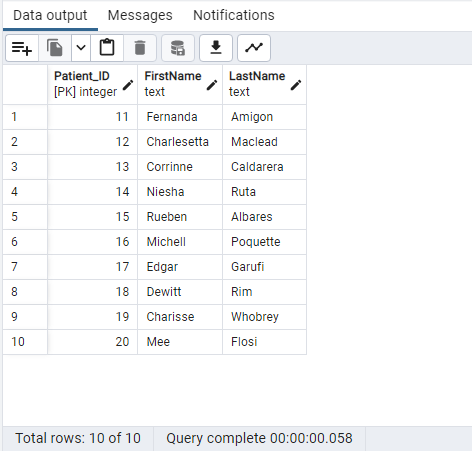
ON PS."PS\_ID" = P."PS\_ID" WHERE PS."ProviderSpeciality" = 'Surgery'



-- 74. List of patient FROM rows 11-20 without using WHERE condition.

Query :

SELECT "Patient\_ID","FirstName","LastName" FROM "Patients" ORDER BY "Patient\_ID" LIMIT 10 OFFSET 10



-- 75. Give a query how to find triggers FROM table AmbulatoryVisits

Query:

SELECT event\_object\_table AS table\_name ,trigger\_name

FROM information\_schema.triggers

WHERE event\_object\_table ='AmbulatoryVisits'

GROUP BY table\_name , trigger\_name

ORDER BY table\_name ,trigger\_name

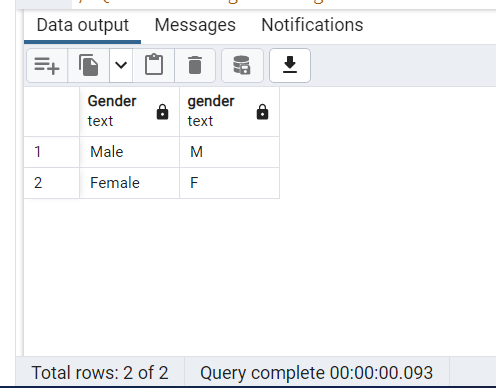
Graphical user interface, application, Word

Description automatically generated

-- Q76:Recreate the below expected output using Substring.\*/

Query :

SELECT "Gender" , substring("Gender",1,1) as "gender" FROM "Gender"



/\* Q77: Obtain the below output by grouping the patients.\*/

Query:

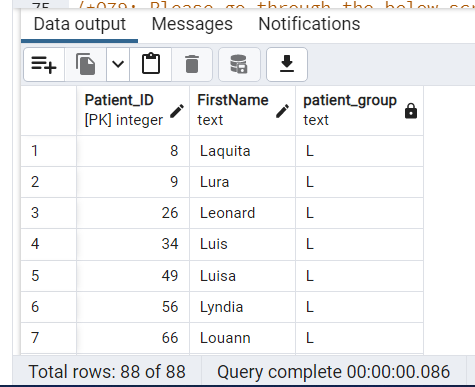
SELECT "Patient\_ID"

,"FirstName"

,substring("FirstName",1,1) as "patient\_group"

FROM "Patients"

WHERE "FirstName" Like 'L%'



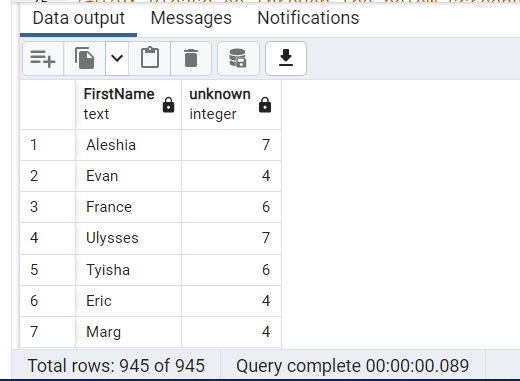
--Q78: Please go through the below screenshot and CREATE the exact output.\*/

Query:

SELECT "FirstName"

,length("FirstName") as unknown

FROM "Patients"



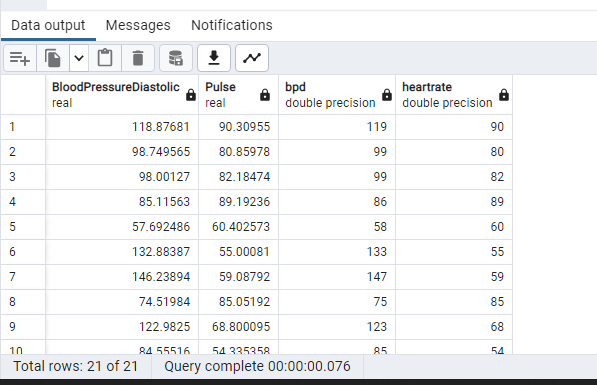
--Q79: Please go through the below screenshot and CREATE the exact output \*/

Query:

SELECT "BloodPressureDiastolic","Pulse"

, ceil("BloodPressureDiastolic") as "bpd"

, Floor("Pulse") as "heartrate" FROM "AmbulatoryVisits" offset 1 Limit 21



--Q80: Please go through the below screenshot and CREATE the exact output \*/

Query:

SELECT "BloodPressureSystolic"

, 'The Systolic Blood Pressure is ' || round(cast("BloodPressureSystolic" as Decimal),2) as "Message"

FROM "AmbulatoryVisits"

