**Note: For Code Optimization, We created HRV column in IBI table using temp table. It stores the HRV value for all the patient. we haven’t calculated HRV separately for the queries needed instead we used it from IBI table.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*created the hrv column in IBI Table\*\*\*\*\*\*\*\*\*\*\*\*\***

**-- Create a temporary table to store HRV calculations**

CREATE TEMP TABLE Temp\_HRV AS

SELECT

patientid,

datestamp,

rmssd\_ms,

AVG(rmssd\_ms) OVER (PARTITION BY patientid ORDER BY datestamp ASC) \* 600 AS HRV

FROM

ibi;

**--altering the IBI table with adding HRV column**

ALTER TABLE IBI

ADD COLUMN HRV numeric;

**-- Update the IBI table with HRV values**

UPDATE ibi SET HRV = Temp\_HRV.HRV

FROM Temp\_HRV

WHERE

ibi.patientid = Temp\_HRV.patientid

AND ibi.datestamp = Temp\_HRV.datestamp;

**-- Drop the temporary table**

DROP TABLE Temp\_HRV;

**-- selecting the table**

select patientid, AVG(rmssd\_ms)\*600 as hrv from ibi group by patientid

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**@@@One Other Team Member created the HRV table Seaprately and she used this**

CREATE TABLE HRV AS

SELECT

patientid,

datestamp,

rmssd\_ms,

AVG(rmssd\_ms) OVER (PARTITION BY patientid ORDER BY datestamp ASC) \* 600 AS HRV FROM ibi;

**Q1: Write a query to get a list of patients with event type of EGV and glucose (mgdl)**

**greater than 155 .**

**Query**

SELECT id, event\_type, glucose\_value\_mgdl

FROM public.eventtype WHERE event\_type = 'EGV' AND glucose\_value\_mgdl > 155;

**OUTPUT**

A screenshot of a computer

Description automatically generated

**Q2 :How many patients consumed meals with at least 20 grams of protein in it?**

**Query**

SELECT COUNT(DISTINCT patientid) as PatientCount

FROM foodlog

WHERE protein >= 20;

**OUTPUT**

A screenshot of a computer

Description automatically generated

**Q3 Who consumed maximum calories during dinner? (assuming the dinner time is after 6pm-8pm)**

**Query**

WITH total\_dinner\_calories as(

SELECT patientid,

sum(calorie) as total\_dinner\_calorie

FROM

foodlog

WHERE

EXTRACT(HOUR FROM datetime) >= 18 AND EXTRACT(HOUR FROM datetime) < 20

GROUP BY patientid

)

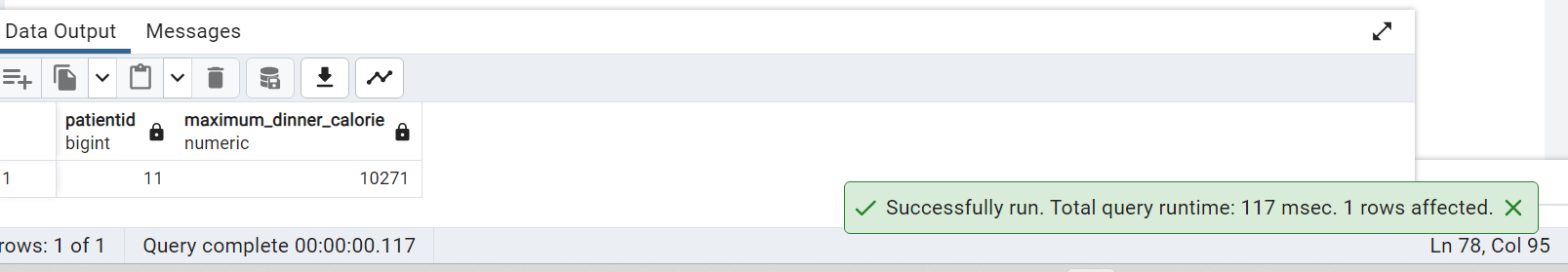
SELECT t.patientid,

t.total\_dinner\_calorie As Maximum\_Dinner\_Calorie

FROM total\_dinner\_calories t

WHERE t.total\_dinner\_calorie=(SELECT MAX(x.total\_dinner\_calorie) from total\_dinner\_calories x)

**OUTPUT**



**Q4: Which patient showed a high level of stress on most days recorded for him/her?**

**Query**

WITH StressCounts AS ( SELECT

e.patientid,

COUNT(\*) AS high\_stress\_days

FROM eda AS e WHERE

e.max\_eda > 40

GROUP BY

e.patientid

)

SELECT s.patientid FROM StressCounts s

WHERE

s.high\_stress\_days = (

SELECT MAX(high\_stress\_days) FROM StressCounts

);

**OUTPUT**

A screenshot of a computer

Description automatically generated

**Q5: Based on mean HR and HRV alone, which patient would be considered least healthy?**

**Query**

SELECT i.patientid,

(AVG(h.max\_hr) \* AVG(i.hrv)) AS health\_score

FROM ibi AS i

JOIN hr AS h ON i.patientid = h.patientid

GROUP BY i.patientid

ORDER BY health\_score ASC

LIMIT 1;

**OUTPUT**

A screenshot of a computer

Description automatically generated

**Q6: Create a table that stores any Patient Demographics as firstname, dob,patientid,hba1c of your choice as the parent table.**

**Create a child table that contains max\_EDA from EDA table and mean\_HR from HR per patient and inherits all columns from the parent table**

**Query**

drop table patient\_demographic cascade;

CREATE TABLE patient\_demographic (

patientid serial PRIMARY KEY,

firstname VARCHAR(30) ,

dob DATE ,

hba1c DECIMAL(5, 2)

);

INSERT INTO patient\_demographic (firstname, dob, hba1c)

VALUES ('Jane', '1990-01-01', 5.5);

select \* from patient\_demographic;

**OUTPUT**

A screenshot of a computer

Description automatically generated

drop table patient\_additional\_info cascade

CREATE TABLE patient\_additional\_info (

max\_eda DECIMAL(5, 2),

mean\_hr DECIMAL(5, 2),

patientid INT REFERENCES patient\_demographic (patientid)

)INHERITS (patient\_demographic);

INSERT INTO patient\_additional\_info(firstname, dob, hba1c,max\_eda,mean\_hr,patientid)

VALUES ('sita','1990-01-01',5.5,5.50,13.67,1);

select \* from patient\_additional\_info ;

**OUTPUT**

A screenshot of a computer

Description automatically generated

**Q7: What percentage of the dataset is male vs what percentage is female?**

**Query**

SELECT gender,

COUNT(\*) AS count,

Round((COUNT(\*) \* 100.0 / (SELECT COUNT(\*) FROM demographics)),2) AS percentage

FROM demographics

GROUP BY gender;

**OUTPUT**

A screenshot of a computer

Description automatically generated

**Q8: Which patient has the highest max eda?**

**Query**

With Max\_eda\_score as

(SELECT patientid,

MAX(max\_eda) AS highest\_max\_eda

FROM eda

GROUP BY patientid

)

SELECT m.patientid,

m.highest\_max\_eda

FROM Max\_eda\_score m

WHERE

m.highest\_max\_eda=(SELECT Max(x.highest\_max\_eda) from Max\_eda\_score x)

**OUTPUT**

A screenshot of a computer

Description automatically generated

**Q9: Display details of the prediabetic patients.**

**Query**

SELECT \* FROM demographics WHERE hba1c >= 5.7 AND hba1c <= 6.4;

**OUTPUT**

A screenshot of a computer

Description automatically generated

**Q10: List the patients that fall into the highest EDA category by name, gender and age**

**Query**

SELECT

pd.firstname,

pd.lastname,

pd.gender,

pd.dob

FROM

demographics AS pd

INNER JOIN ( SELECT patientid, MAX(max\_eda) AS highest\_eda

FROM eda GROUP BY patientid

) AS max\_eda\_table ON pd.patientid = max\_eda\_table.patientid

WHERE max\_eda\_table.highest\_eda = (SELECT MAX(max\_eda) FROM eda);

**OUTPUT**

A screenshot of a computer

Description automatically generated

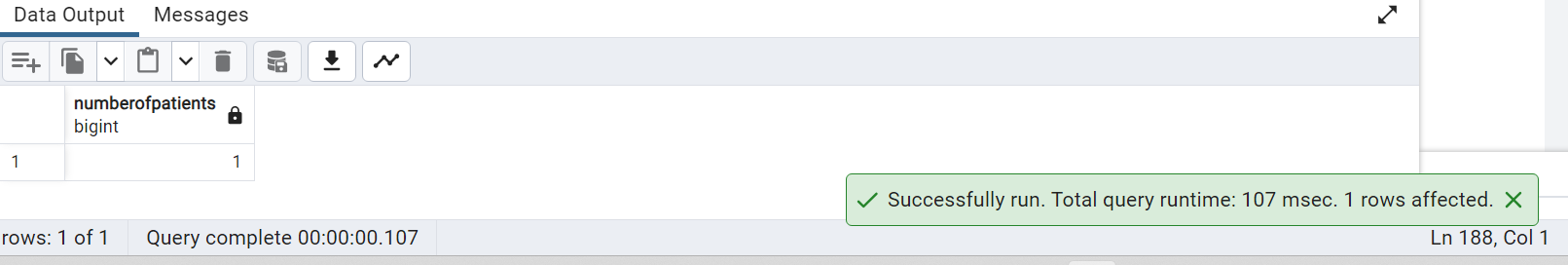
**Q11: How many patients have names starting with 'A'?**

**Query**

SELECT COUNT(\*) as numberofpatients FROM demographics

WHERE firstname LIKE 'A%';

**OUTPUT**



**Q12: Show the distribution of patients across age.**

**Query**

SELECT CASE WHEN dob IS NOT NULL THEN

EXTRACT(YEAR FROM AGE(NOW(), dob))

ELSE

NULL

END AS age,

COUNT(\*) AS patient\_count

FROM demographics GROUP BY age ORDER BY age;

**OUTPUT**

A screenshot of a computer

Description automatically generated

**Q13: Display the Date and Time in 2 seperate columns for the patient who consumed only Egg**

**Query**

SELECT

datetime AS consumption\_datetime,

CAST(datetime AS DATE) AS consumption\_date,

CAST(datetime AS TIME) AS consumption\_time,

logged\_food

FROM

foodlog

WHERE

logged\_food = 'Egg';

**OUTPUT**

A screenshot of a computer

Description automatically generated

**Q14: Display list of patients along with the gender and hba1c for whom the glucose value is null.**

**Query**

SELECT

d.patientid,

d.gender,

d.hba1c

FROM demographics AS d

LEFT JOIN dexcom AS g ON d.patientid = g.patientid

WHERE g.glucose\_value\_mgdl IS NULL;

**OUTPUT**

A screenshot of a computer

Description automatically generated

**Q15: Rank patients in descending order of Max blood glucose value per day**

**Query**

WITH MaxGlucosePerDay AS (

SELECT

patientid,

DATE(datestamp) AS observation\_date,

MAX(glucose\_value\_mgdl) AS max\_daily\_glucose

FROM dexcom

GROUP BY

patientid,

DATE(datestamp)

)

SELECT

patientid,

observation\_date,

max\_daily\_glucose,

RANK() OVER (PARTITION BY observation\_date ORDER BY max\_daily\_glucose DESC) AS rank\_per\_day

FROM MaxGlucosePerDay ORDER BY observation\_date DESC, max\_daily\_glucose DESC;

**OUTPUT**

A screenshot of a computer

Description automatically generated

**Q16 : Assuming the IBI per patient is for every 10 milliseconds, calculate Patient-wise HRV from RMSSD.**

**/\* to calculate HRV and create a column with HRV=Avg rmssd\_ms from IBI per patientid from IBI \* 600 \*/**

**Query**

-- Create a temporary table to store HRV calculations

CREATE TEMP TABLE Temp\_HRV AS

SELECT

patientid,

datestamp,

rmssd\_ms,

AVG(rmssd\_ms) OVER (PARTITION BY patientid ORDER BY datestamp ASC) \* 600 AS HRV FROM ibi;

ALTER TABLE IBI

ADD COLUMN HRV numeric;

-- Update the IBI table with HRV values

UPDATE ibi SET HRV = Temp\_HRV.HRV

FROM Temp\_HRV

WHERE

ibi.patientid = Temp\_HRV.patientid

AND ibi.datestamp = Temp\_HRV.datestamp;

-- Drop the temporary table

**OUTPUT:**

DROP TABLE Temp\_HRV;

A screenshot of a computer

Description automatically generated

**Q.17 What is the % of total daily calories consumed by patient 14 after 3pm Vs Before 3pm?**

**Query:**

SELECT patientid,

SUM(CASE WHEN EXTRACT(hour FROM datetime) >= 15 THEN calorie ELSE 0 END) AS calories\_after\_3pm,

SUM(CASE WHEN EXTRACT(hour FROM datetime) < 15 THEN calorie ELSE 0 END) AS calories\_before\_3pm,

SUM(calorie) AS total\_calories,

(SUM(CASE WHEN EXTRACT(hour FROM datetime) >= 15 THEN calorie ELSE 0 END) \* 100.0) / SUM(calorie) AS percentage\_after\_3pm,

(SUM(CASE WHEN EXTRACT(hour FROM datetime) < 15 THEN calorie ELSE 0 END) \* 100.0) / SUM(calorie) AS percentage\_before\_3pm

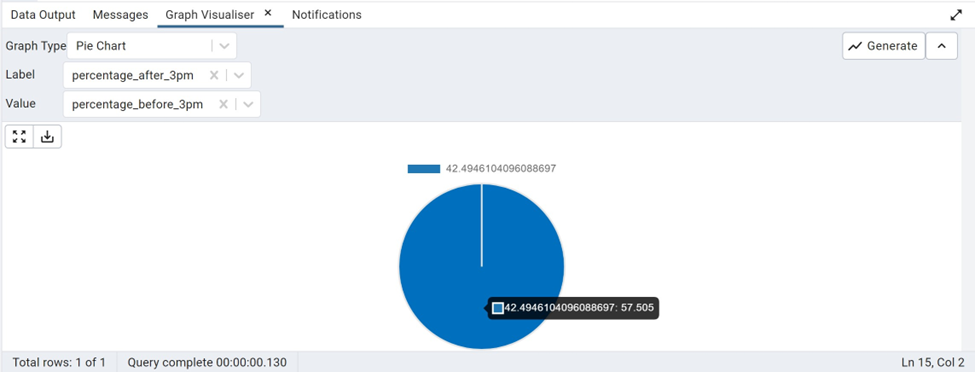
FROM foodlog

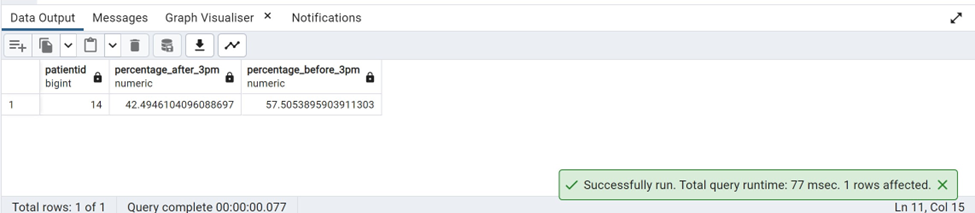
WHERE patientid = 14

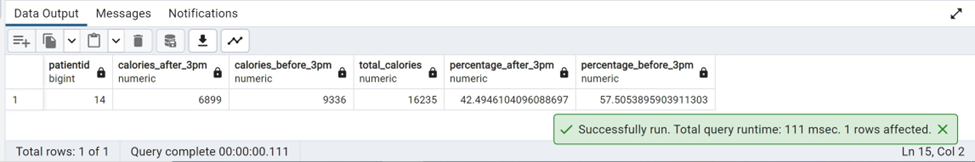
GROUP BY

patientid;

**OUTPUT**

****

****

****

**Q18 Display 5 random patients with HbA1c less than 6.**

**Query:**

SELECT CONCAT(firstname,' ',lastname)

AS patient\_name FROM demographics

WHERE hba1c < 6

ORDER BY RANDOM() LIMIT 5;

**OUTPUT**

****

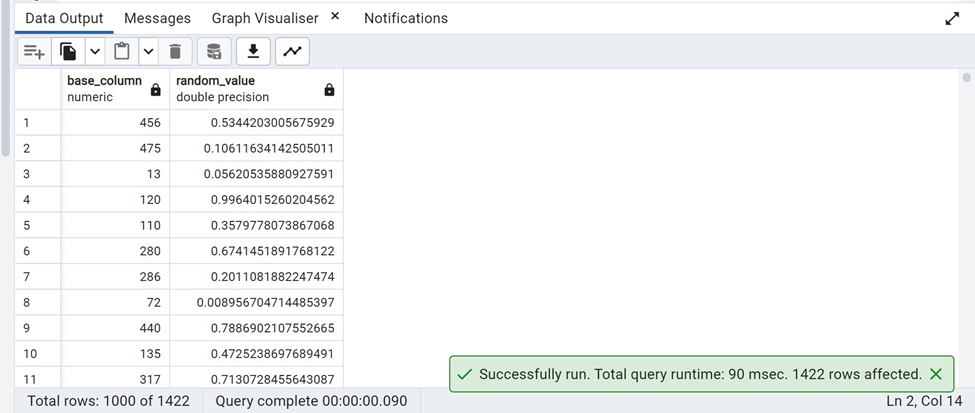
**Q19 Generate a random series of data using any column from any table as the base.**

**Query:**

SELECT calorie AS base\_column, RANDOM() AS random\_value

FROM foodlog;

**OUTPUT**

****

**Q20 Display the foods consumed by the youngest patient.**

**Query:**

SELECT f.logged\_food FROM foodlog f

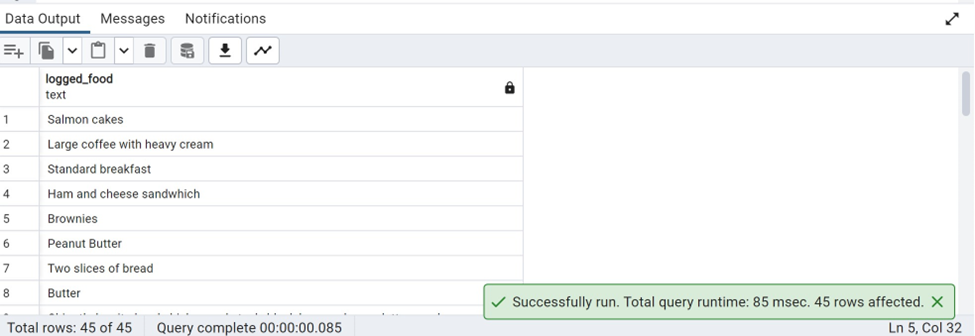
INNER JOIN (SELECT patientid FROM demographics

ORDER BY dob ASC

LIMIT 1) AS d

ON f.patientid =d.patientid;

**OUTPUT**

****

**Q21 Identify the patients that has letter ‘h’ in their first name and print the last letter of their first name.**

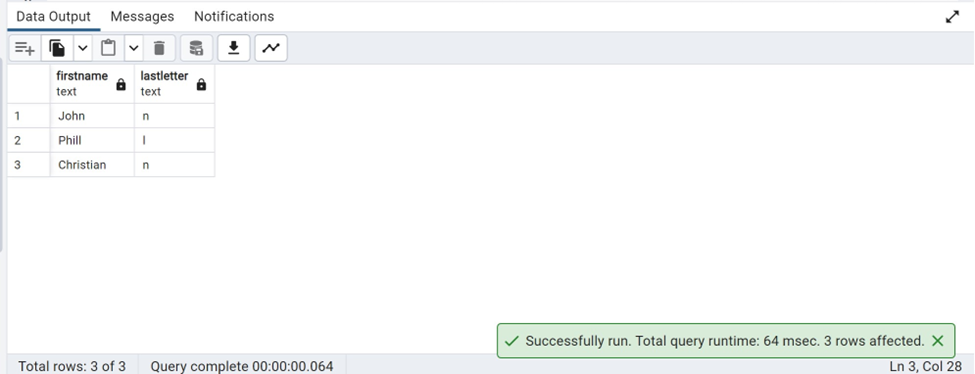
**Query:**

SELECT firstname, RIGHT(firstname,1) AS lastletter

FROM demographics

WHERE firstname LIKE '%h%';

**OUTPUT**

****

**Q22 Calculate the time spent by each patient outside the recommended blood glucose range.**

**Query:**

SELECT patientid,EXTRACT(EPOCH FROM (MAX(datestamp) - MIN(datestamp)))/3600 AS time\_outside\_range

FROM dexcom

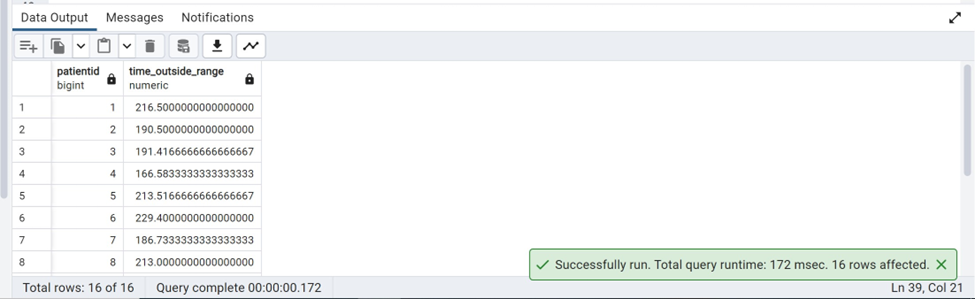
WHERE

(glucose\_value\_mgdl < 70 OR glucose\_value\_mgdl >= 126)

GROUP BY patientid

ORDER BY patientid;

**OUTPUT**

****

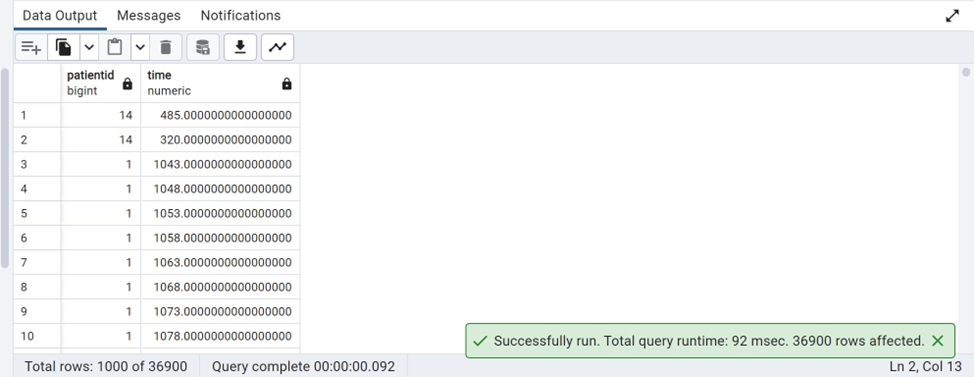
**Q23 Show the time in minutes recorded by the Dexcom for every patient.**

**Query:**

SELECT patientid, EXTRACT(EPOCH FROM datestamp::TIME)/60 AS time

FROM dexcom;

**OUTPUT**

****

**Q24 List all the food eaten by patient Phill Collins.**

**Query:**

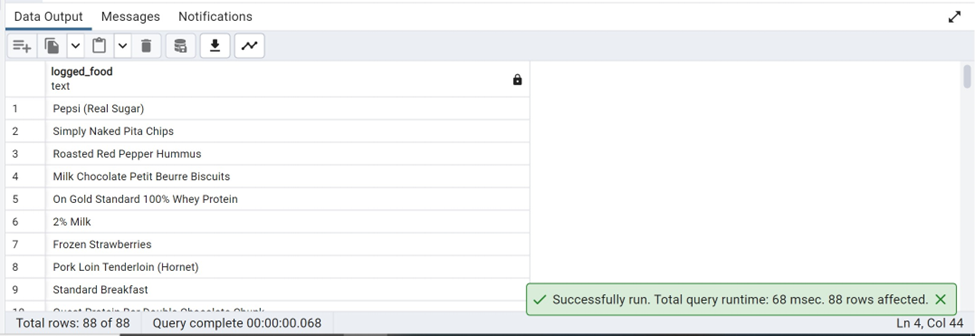
SELECT f.logged\_food FROM foodlog f

INNER JOIN (SELECT patientid ,CONCAT(firstname,' ',lastname) AS patient\_name FROM demographics) AS d

ON f.patientid = d.patientid

WHERE d.patient\_name= 'Phill Collins';

**OUTPUT**



**Q25 Create a stored procedure to delete the min\_EDA column in the table EDA.**

**Query:**

CREATE PROCEDURE delete\_column()

LANGUAGE plpgsql

AS $$

BEGIN

EXECUTE 'ALTER TABLE eda DROP COLUMN min\_EDA';

EXCEPTION

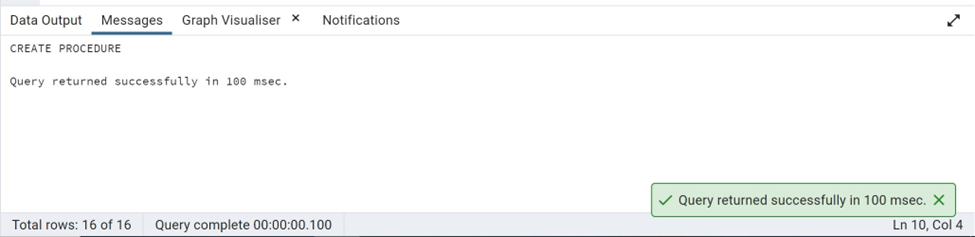
WHEN others THEN

RAISE NOTICE 'Column min\_EDA does not exist in table EDA';

END;

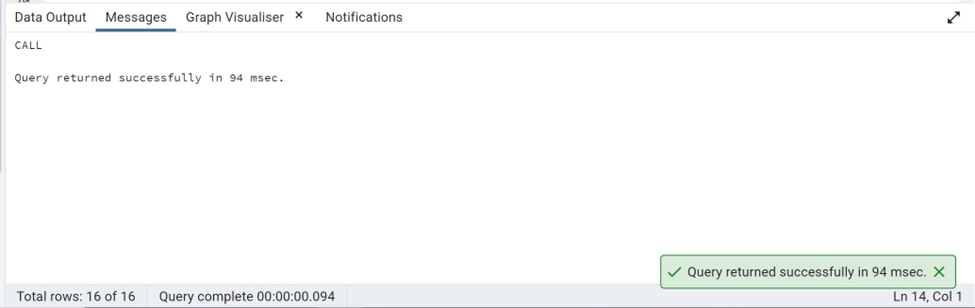
$$

**OUTPUT**

****

CALL delete\_column();

**OUTPUT**

****

SELECT \* FROM eda;

**OUTPUT**

****

**Q26 When is the most common time of day for people to consume spinach?**

**Query:**

SELECT datetime::TIME AS time,

COUNT(\*) AS repeatingNO

FROM foodlog

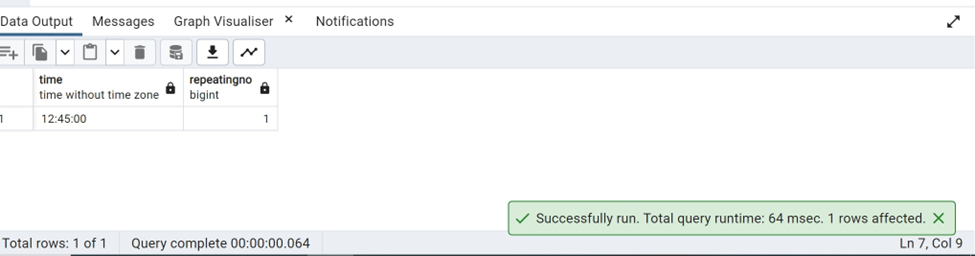
WHERE logged\_food = 'Spinach'

GROUP BY time

ORDER BY repeatingNo DESC

LIMIT 1;

**OUTPUT**

****

**Q27 Classify each patient based on their HRV range as high, low, or normal.**

**Query:**

SELECT patientid,(AVG(rmssd\_ms)\*600) AS hrv,

CASE

WHEN (AVG(rmssd\_ms) \* 600) < 20 THEN 'Low'

WHEN (AVG(rmssd\_ms) \* 600) >= 20 AND (AVG(rmssd\_ms) \* 600) <=30 THEN 'Normal'

ELSE 'High'

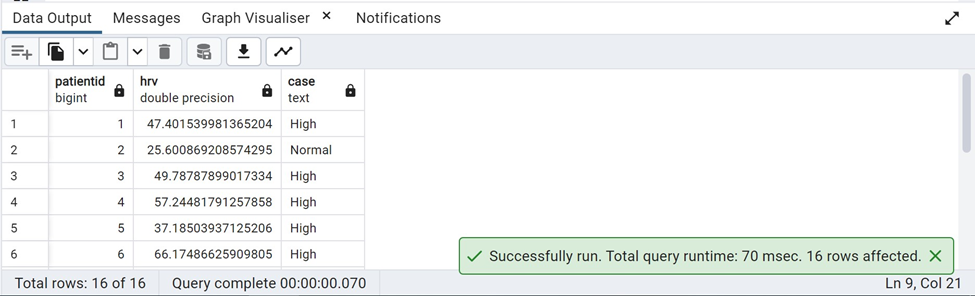
END

FROM ibi

GROUP BY patientid

ORDER BY patientid;

**OUTPUT**

****

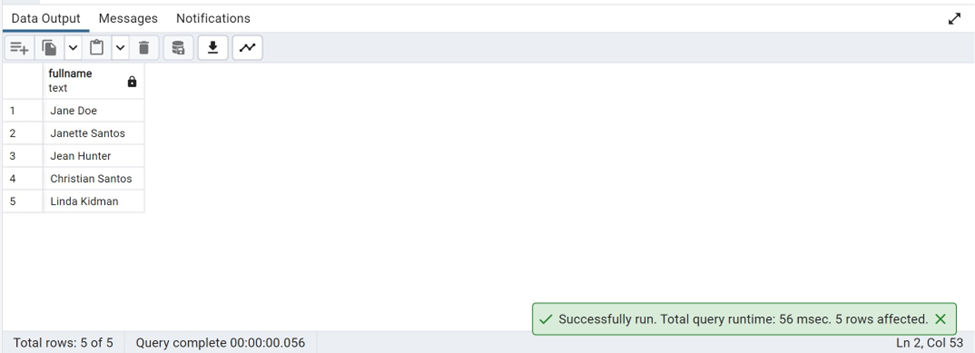
**Q28 List full name of all patients with ‘an’ in their first or last names.**

**Query:**

SELECT CONCAT(firstname,' ',lastname) AS fullname FROM demographics

WHERE firstname LIKE '%an%' OR lastname LIKE '%an%';

**OUTPUT**

****

**Q29 Display a pie chart of gender vs average HbA1c.**

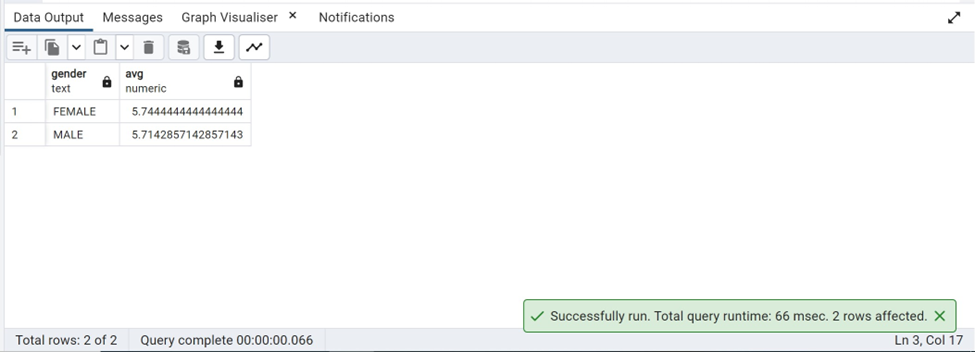
**Query:**

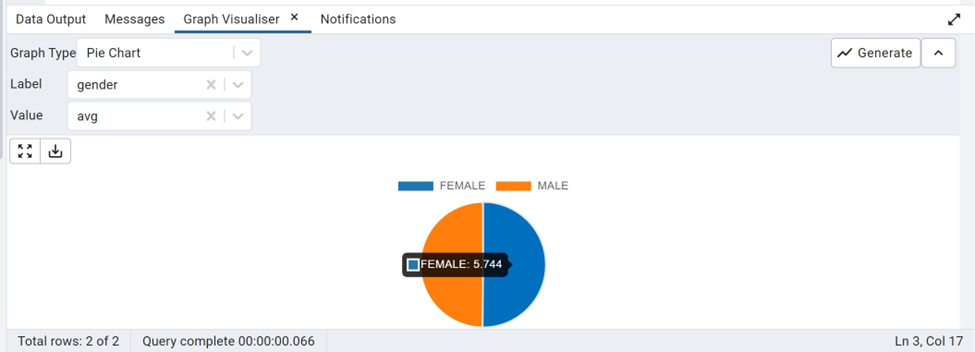
SELECT gender, AVG(HbA1c)

FROM demographics

GROUP BY gender;

**OUTPUT**

****

****

**Q30 The recommended daily allowance of fiber is approximately 25 grams a day. What % of this does every patient get on average?**

**Query:**

SELECT patientid,

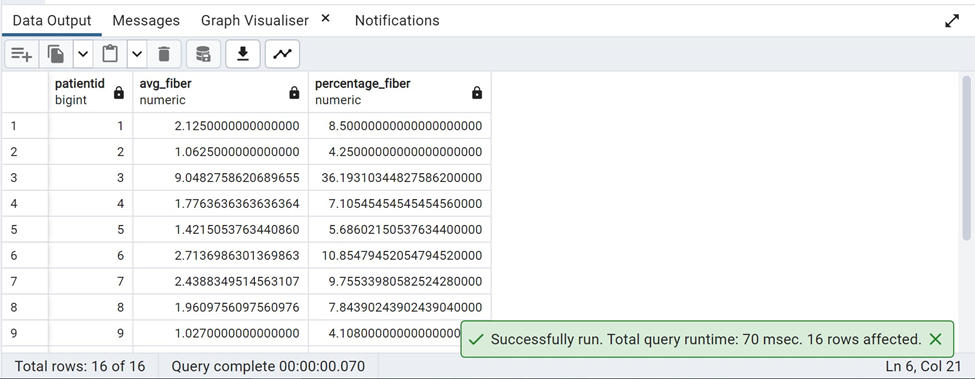
AVG(dietary\_fiber) AS avg\_fiber,

(AVG(dietary\_fiber)/25)\*100 AS percentage\_fiber

FROM foodlog

GROUP BY patientid

ORDER BY patientid;

**OUTPUT**

**Q31 What is the relationship between EDA and Mean HR?**

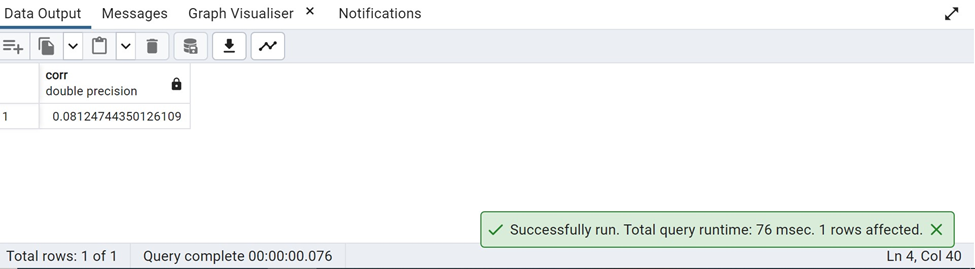
**Query:**

SELECT corr(e.mean\_eda, t2.mean\_hr)

FROM eda e

JOIN hr t2

ON e.patientid = t2.patientid;

**OUTPUT**

**Q32 Show the patient that spent the maximum time out of range.**

**Query:**

SELECT patientid,EXTRACT(EPOCH FROM (MAX(datestamp) - MIN(datestamp)))/3600 AS time\_outside\_range

FROM dexcom

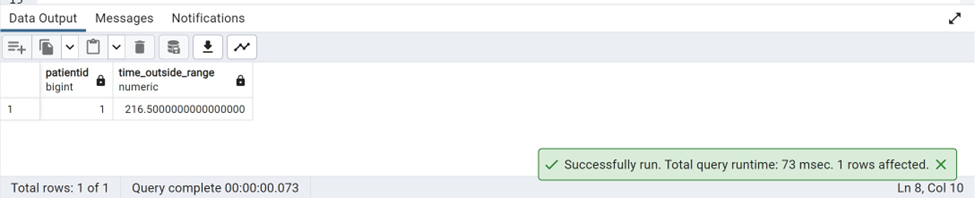
WHERE

(glucose\_value\_mgdl < 70 OR glucose\_value\_mgdl >= 126)

GROUP BY patientid

ORDER BY patientid

LIMIT 1;

**OUTPUT**

**Q33 : Create a User Defined function that returns min glucose value and patient ID for any date entered**

**Query**

DROP FUNCTION if exists minGlucose;

DROP TABLE if exists min\_glucose\_dexcom;

CREATE TABLE min\_glucose\_dexcom

( glucose\_value\_mgdl real,

patientid integer);

CREATE FUNCTION minGlucose(dts date) RETURNS min\_glucose\_dexcom AS $$

DELETE FROM min\_glucose\_dexcom;

INSERT INTO min\_glucose\_dexcom

SELECT MIN(glucose\_value\_mgdl),

patientid FROM dexcom

WHERE date(datestamp) = dts and glucose\_value\_mgdl is not null

GROUP BY patientid;

SELECT glucose\_value\_mgdl,

patientid

FROM min\_glucose\_dexcom;

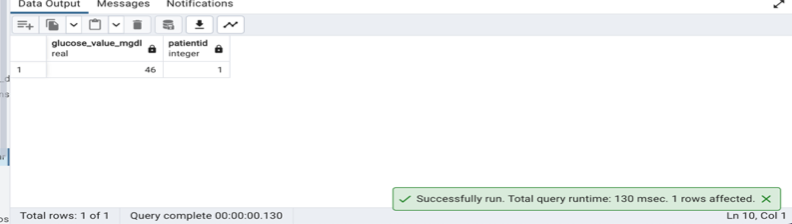
$$ LANGUAGE SQL

;

-- Call the above procedure to display the patientid and minimum glucose value for a date.

SELECT \* FROM minGlucose ('2020-02-16');

**Output**

****

**Q34 : Write a query to find the day of highest mean HR value for each patient and display it along with the patient id.**

**Query**

SELECT datestamp,

hr.patientid,

mean\_hr

FROM hr,

(SELECT MAX(mean\_hr) AS mean\_hr2 ,

patientid from hr

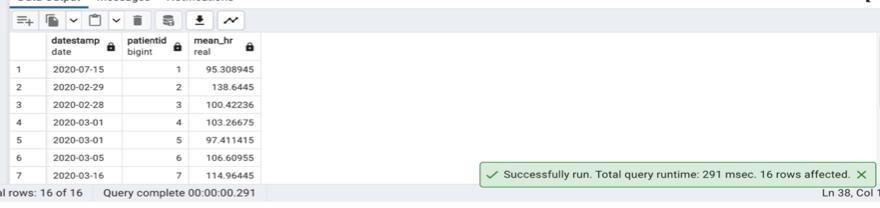
GROUP BY patientid) hr2

WHERE hr.patientid = hr2.patientid

AND hr.mean\_hr = hr2.mean\_hr2

;

**Output**

****

**Q35: Create view to store Patient ID, Date, Avg Glucose value and Patient Day to every patient, ranging from 1-11 based on every patients minimum date and maximum date (eg: Day1,Day2 for each patient)**

**Query**

CREATE OR REPLACE VIEW patient\_glucose\_daily\_summary AS

SELECT

patientid,

'Day ' || ROW\_NUMBER() OVER (PARTITION BY patientid ORDER BY reading\_date) AS patient\_day,

date(reading\_date) as reading\_date,

avg\_glucose

FROM (

select

patientid,

date\_trunc('day',datestamp) as reading\_date,

avg(glucose\_value\_mgdl) as avg\_glucose

from dexcom

group by

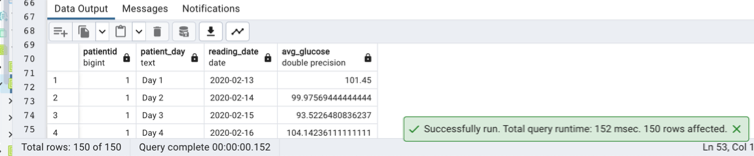
patientid,

reading\_date

order by patientid, reading\_date asc) subquery;

SELECT \* FROM patient\_glucose\_daily\_summary;

**Output**

****

**Q36 : Using width bucket functions, group patients into 4 HRV categories**

**Query**

SELECT

distinct

CASE

WHEN bucket = 1 THEN '1. Low HRV'

WHEN bucket = 2 THEN '2. Moderate HRV'

WHEN bucket = 3 THEN '3. Moderately High HRV'

WHEN bucket = 4 THEN '4. High HRV'

ELSE 'Invalid HRV'

END AS hrv\_bucket,

patientid,

hrv

FROM (

select patientid,

hrv,

width\_bucket(hrv, (select min(hrv) from hrv),(select max(hrv) from hrv),3) as bucket

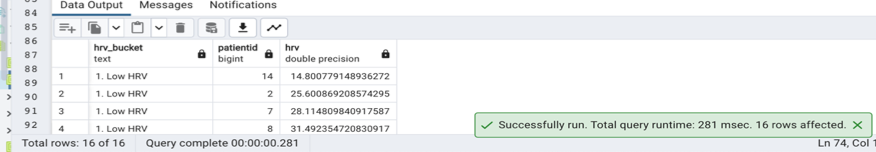
from hrv

) AS hrv\_bucketed

order by 1,3

;

**Output**

****

**Q37 : Is there a correlation between High EDA and HRV. If so, display this data by querying the relevant tables?**

**Query**

WITH HighEda AS (

SELECT

e.patientid,

e.datestamp AS eda\_timestamp,

e.max\_eda as high\_eda,

h.datestamp AS hrv\_timestamp,

h.hrv

FROM eda e

JOIN hrv h ON e.patientid = h.patientid

WHERE e.max\_eda > 40

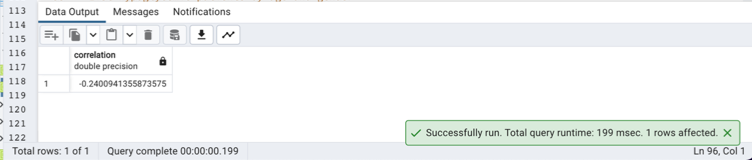
)

SELECT

CORR(HighEda.high\_eda, HighEda.hrv) AS correlation

FROM HighEda;

**Output**

****

**Q38 : List hypoglycemic patients by age and gender**

**Query**

SELECT

d.patientid,

d.gender,

date\_part('year', age(current\_date, d.dob)) AS age,

dx.glucose\_value\_mgdl

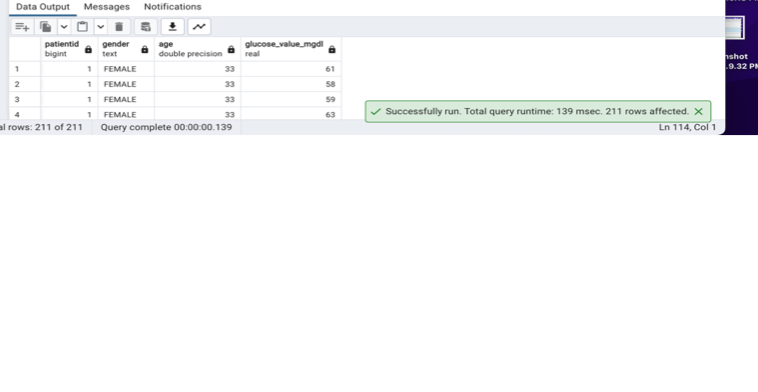
FROM demographics d

JOIN dexcom dx ON d.patientid = dx.patientid

WHERE dx.glucose\_value\_mgdl < 70

ORDER BY age, gender;

**Output**

****

**Q39 : Write a query using recursive view(use the given dataset only)**

**Query**

CREATE OR REPLACE RECURSIVE VIEW max\_mean\_hr (datestamp, patientid, mean\_hr) AS

WITH RECURSIVE recursive\_cte AS (

SELECT

DATE(datestamp) AS datestamp,

patientid,

mean\_hr

FROM hr

WHERE mean\_hr > 100

UNION ALL

SELECT

DATE(hr2.datestamp) AS datestamp,

hr2.patientid,

hr2.mean\_hr

FROM hr hr2

INNER JOIN recursive\_cte r ON hr2.patientid = r.patientid

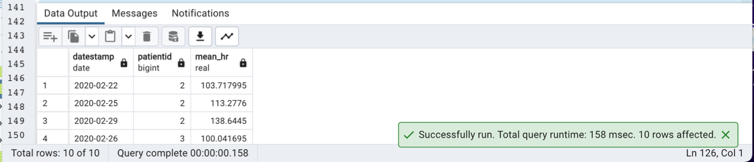
WHERE hr2.mean\_hr > 100

)

SELECT \* FROM recursive\_cte;

select \* from max\_mean\_hr limit 10;

**Output**

****

**Q40 : Create a stored procedure that adds a column to table IBI. The column should just be the date part extracted from IBI.Date**

**Query**

-- DROP Column if exists

ALTER TABLE IBI

drop column if exists date\_part;

-- Create procedure

CREATE OR REPLACE PROCEDURE add\_ibi\_date()

LANGUAGE plpgsql

AS $$

BEGIN

-- Add the new column 'date\_part' to the 'IBI' table

ALTER TABLE IBI ADD COLUMN date\_part date;

-- Update the 'date\_part' column with the date part from 'Date'

UPDATE IBI SET date\_part = DATE(datestamp);

END;

$$;

CALL add\_ibi\_date();

select \* from ibi;

**Output**

****

**Q 41 : Fetch the list of Patient ID's whose sugar consumption exceeded 30 grams on a meal from FoodLog table.**

**Query**

SELECT DISTINCT patientid, MAX(sugar) as max\_sugar

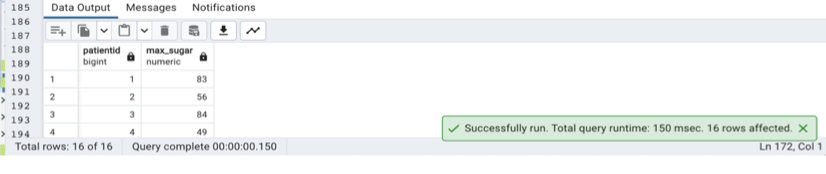
FROM foodlog

WHERE sugar > 30

GROUP BY foodlog.patientid

ORDER BY patientid ASC;

**Output**

****

**Q42 : How many patients are celebrating their birthday this month?**

**Query**

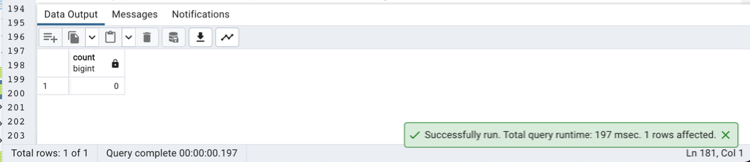
SELECT COUNT(patientid)

FROM demographics

WHERE

EXTRACT(MONTH FROM demographics.DOB)= EXTRACT(MONTH FROM current\_date);

**Output**

****

**Q43 : How many different types of events were recorded in the Dexcom tables? Display counts against each Event type**

**Query**

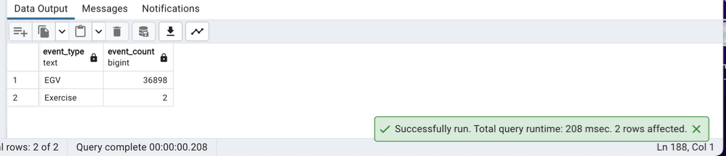
SELECT et.event\_type, COUNT(\*) AS event\_count

FROM dexcom AS d

JOIN eventtype AS et ON d.eventid = et.id

GROUP BY et.event\_type;

**Output**

****

**Q44 : How many prediabetic/diabetic patients also had a high level of stress?**

**Query**

select count(diabetics.patientid)

from

(select patientid,

hba1c,

case when hba1c >= 6.5 then 'diabetic'

when hba1c > 5.7 and hba1c < 6.5 then 'prediabetic'

else 'normal' end

as diabetic\_level

from demographics) diabetics,

(select patientid

-- ,max\_eda

from eda

where max\_eda > 40

union

select patientid

-- ,hrv

from hrv

where hrv < 20

union

select patientid

-- ,mean\_hr

from hr

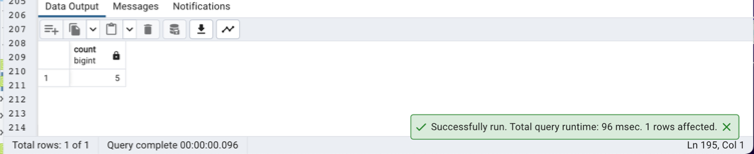
where mean\_hr > 100

order by patientid) high\_stress

where high\_stress.patientid = diabetics.patientid

and diabetics.diabetic\_level in ('prediabetic','diabetic');

**Output**

****

**Q45 : List the food that coincided with the time of highest blood sugar for every patient.**

**Query**

WITH HighSugarFood AS (

SELECT

patientid,

logged\_food,

ROW\_NUMBER() OVER (PARTITION BY PatientID ORDER BY sugar DESC) AS RowNum

FROM

foodlog

)

SELECT

patientid,

logged\_food

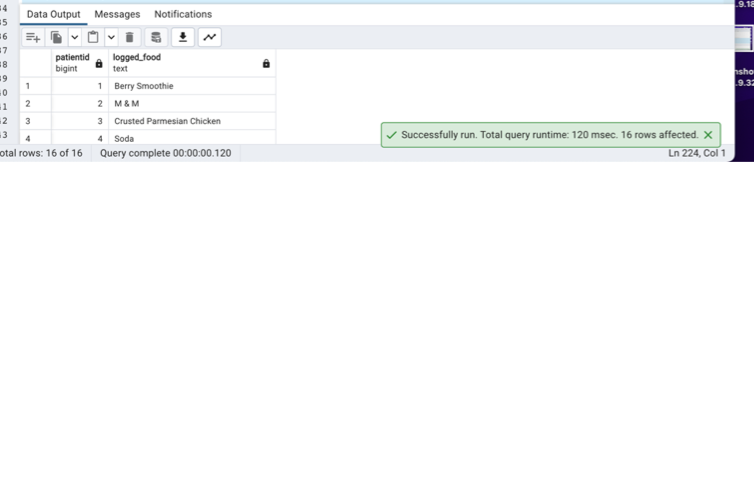
FROM

HighSugarFood

WHERE

RowNum = 1;

**Output**

**Q46 : How many patients have first names with length >7 letters?**

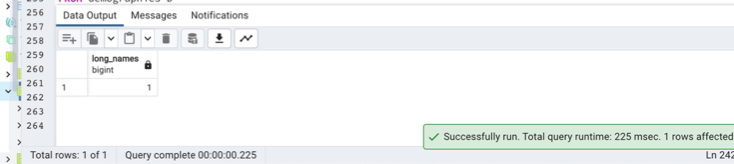
**Query**

SELECT COUNT(\*) AS long\_names

FROM demographics

WHERE LENGTH(firstname) > 7;

**Output**

****

**Q47 : List all foods logged that end with 'se'. Ensure that the output is in Title Case.**

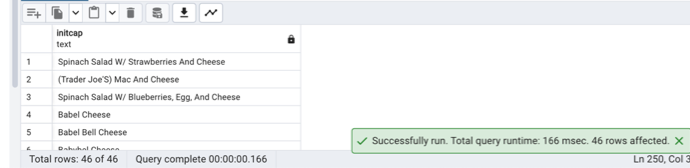
**Query**

SELECT INITCAP(logged\_food)

FROM foodlog

WHERE logged\_food LIKE '%se';

**Output**

****

**Q48 List the patients who had a birthday the same week as their glucose or IBI readings.**

**Query**

SELECT DISTINCT D.patientid, D.DOB

FROM demographics D

INNER JOIN

dexcom Dx ON D.PatientID = Dx.PatientID AND EXTRACT(WEEK FROM D.DOB) = EXTRACT(WEEK FROM Dx.datestamp)

UNION

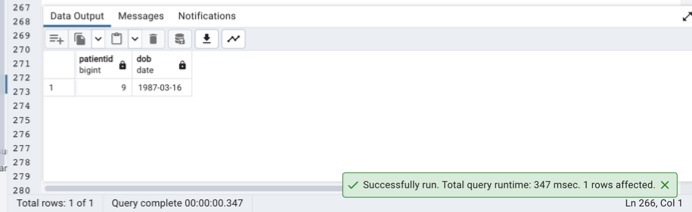
SELECT DISTINCT D.patientid, D.DOB

FROM demographics D

INNER JOIN

ibi I ON D.PatientID = I.PatientID AND EXTRACT(WEEK FROM D.DOB) = EXTRACT(WEEK FROM I.datestamp);

**Output**

****

**Q 49: Assuming breakfast is between 8 am and 11 am. How many patients ate a meal with bananas in it?**

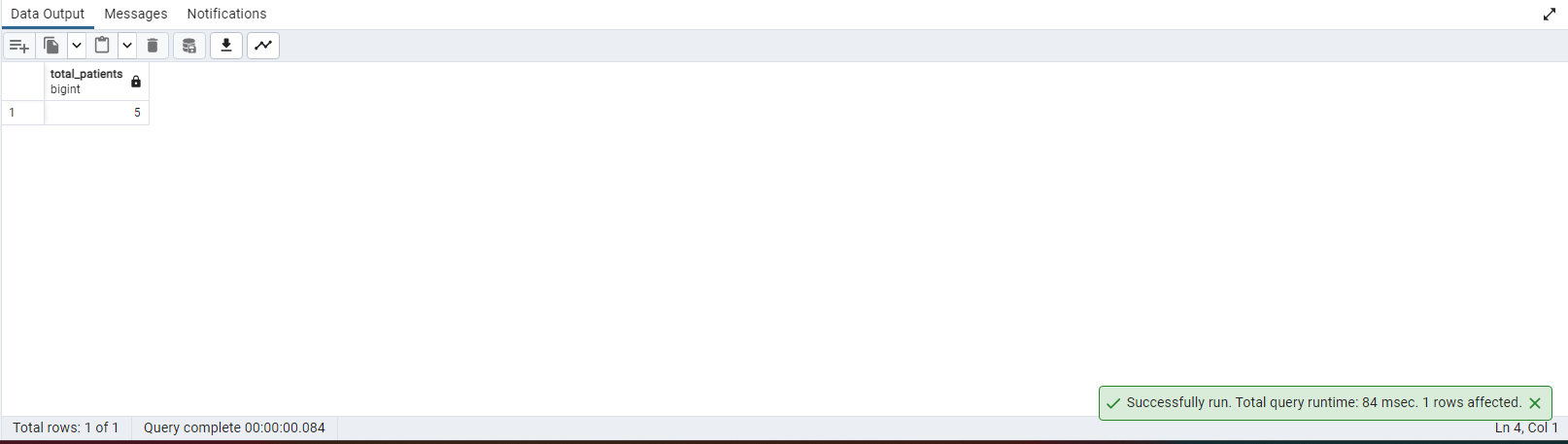
**Query:**

SELECT count( distinct patientid) as total\_patients from foodlog

where Extract(hour from datetime)

between 8 and 11 and logged\_food = 'Banana'

**OUTPUT**



**Q 50: Create a User defined function that returns the age of any patient based on input**

**Query**

create or replace function get\_age(inputid bigint)

returns int

language plpgsql

as

$$

declare

age int;

begin

select DATE\_PART('YEAR', age(current\_date, dob))

into age

from demographics

where patientid=inputid;

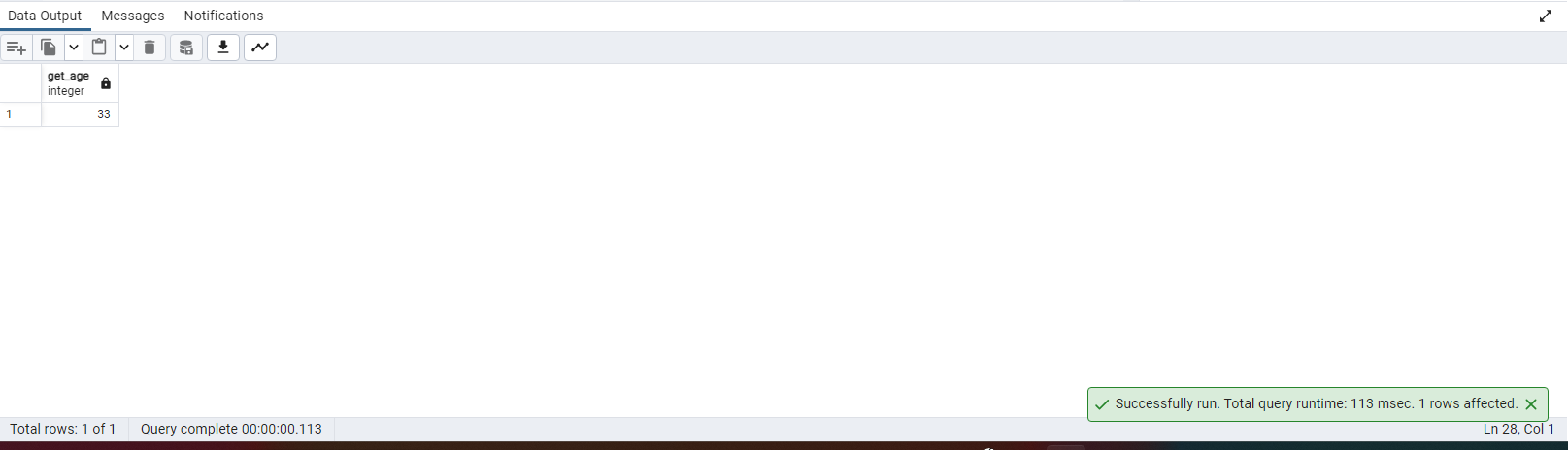
return age;

end;

$$

select get\_age(1)

**OUTPUT**



**Q 51:Based on Number of hyper and hypoglycemic incidents per patient, which patient has the least control over their blood sugar**

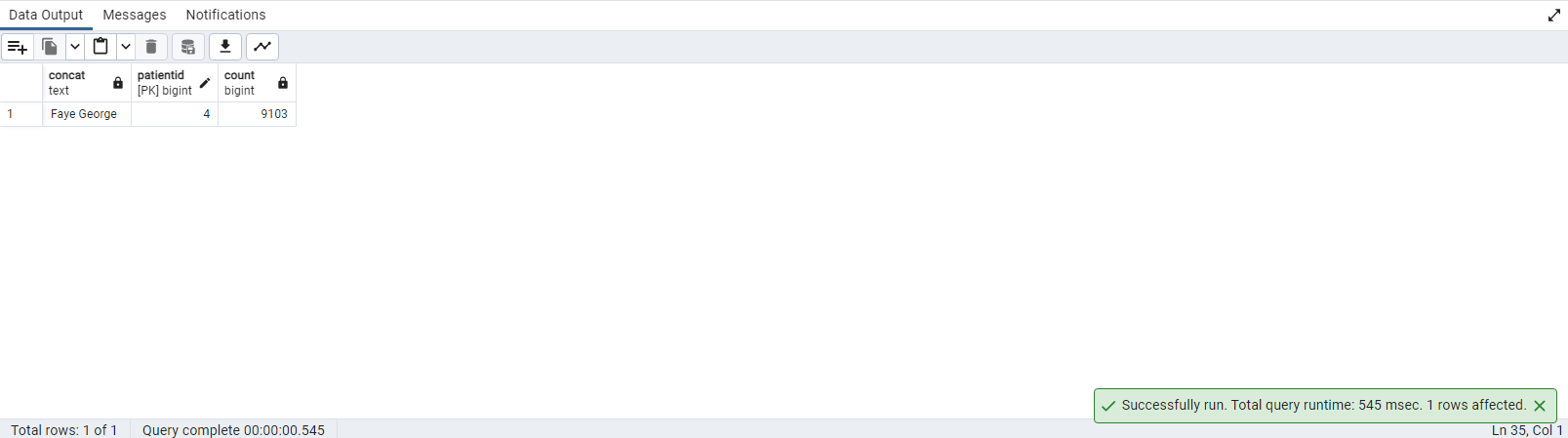
**Query**

select concat(firstname,' ',lastname),demographics.patientid,count( demographics.patientid )

from dexcom, demographics where dexcom.glucose\_value\_mgdl<70 or dexcom.glucose\_value\_mgdl>=126

group by demographics.patientid order by count(demographics.patientid) desc limit 1

**OUTPUT**



**Q 52: Display patients details with event details and minimum heart rate**

**Query**

select concat(firstname,' ',lastname),demographics.patientid,eventtype.event\_type,

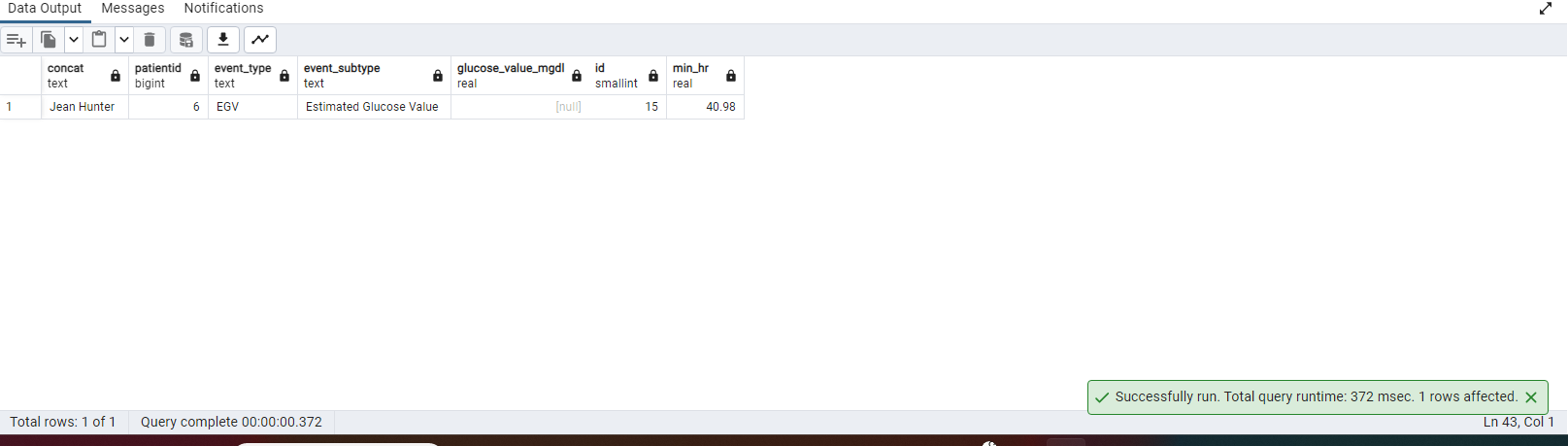
eventtype.event\_subtype,eventtype.glucose\_value\_mgdl,eventtype.id,hr.min\_hr

from demographics inner join dexcom on demographics.patientid= dexcom.patientid

inner join eventtype on dexcom.eventid= eventtype.id inner join hr on dexcom.patientid=hr.patientid

order by hr.min\_hr asc limit 1

**OUTPUT**



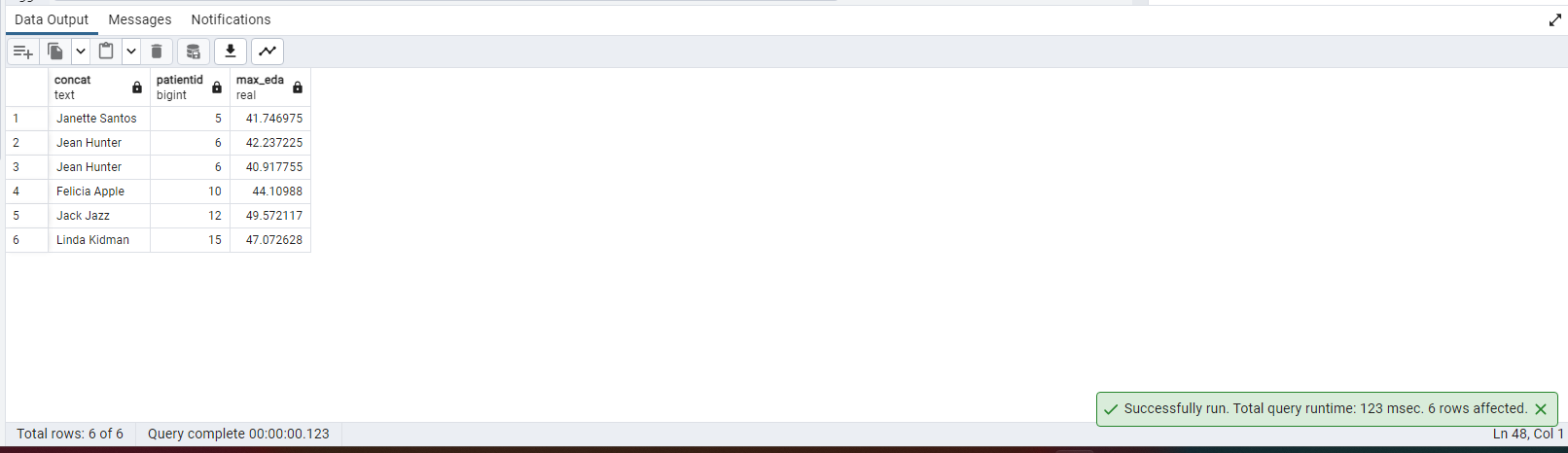
**Q 53: Display a list of patients whose daily max\_eda lies between 40 and 50**

**Query**

select concat(firstname,' ',lastname),demographics.patientid, eda.max\_eda from demographics

inner join eda on demographics.patientid = eda.patientid where eda.max\_eda between 40 and 50

**OUTPUT**



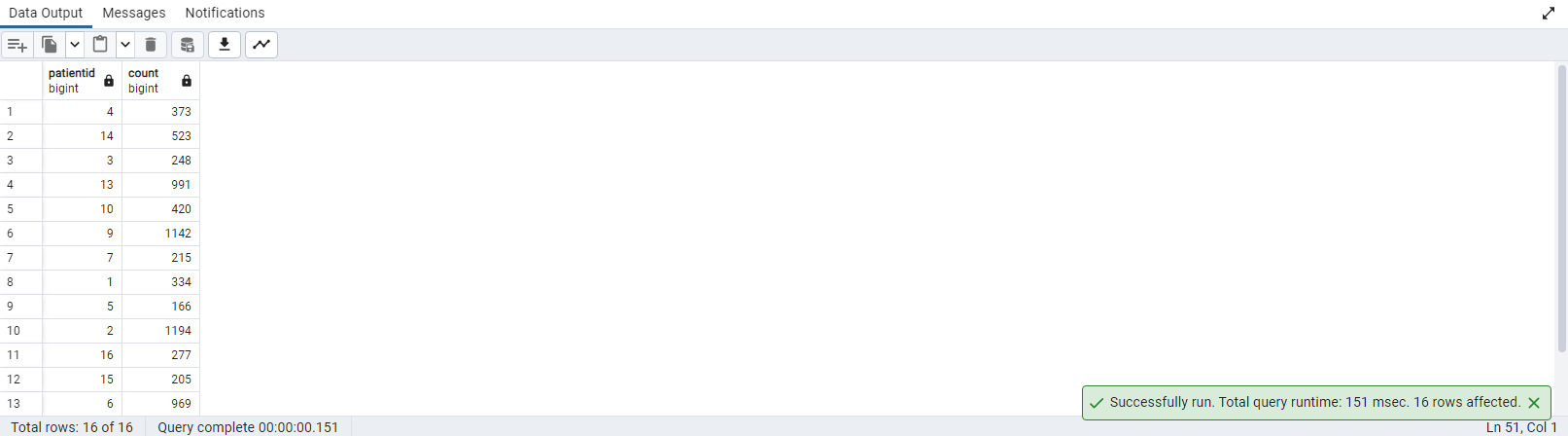
**Q 54:Count the number of hyper and hypoglycemic incidents per patient**

**Query**

select patientid, count(glucose\_value\_mgdl) from dexcom where glucose\_value\_mgdl<70 or glucose\_value\_mgdl>=126

group by patientid

**OUTPUT**

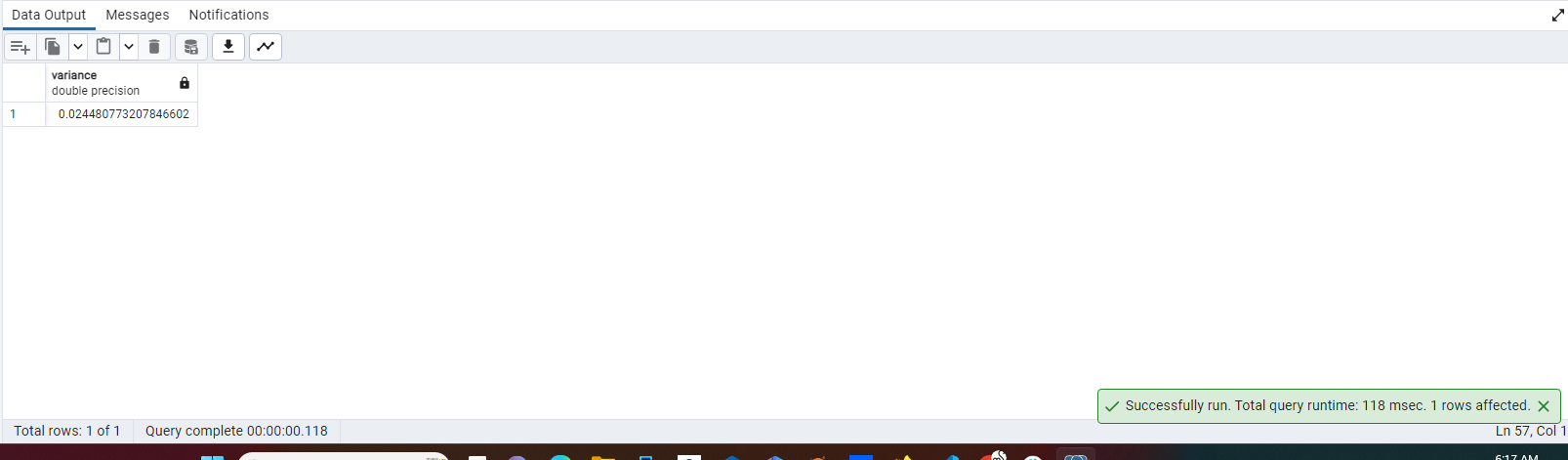


**Q 55: What is the variance from mean for all patients for the table IBI?**

**Query**

select variance(mean\_ibi\_ms) from ibi

**OUTPUT**



**Q 56:Create a view that combines all relevant patient demographics and lab markers into one. Call this view ‘Patient\_Overview’**

**Query**

create or replace view Patient\_Overview as select concat(firstname,' ',lastname),

demographics.patientid, demographics.gender , dexcom.glucose\_value\_mgdl,

eda.mean\_eda,

hr.mean\_hr

from demographics, dexcom, eda, hr

where

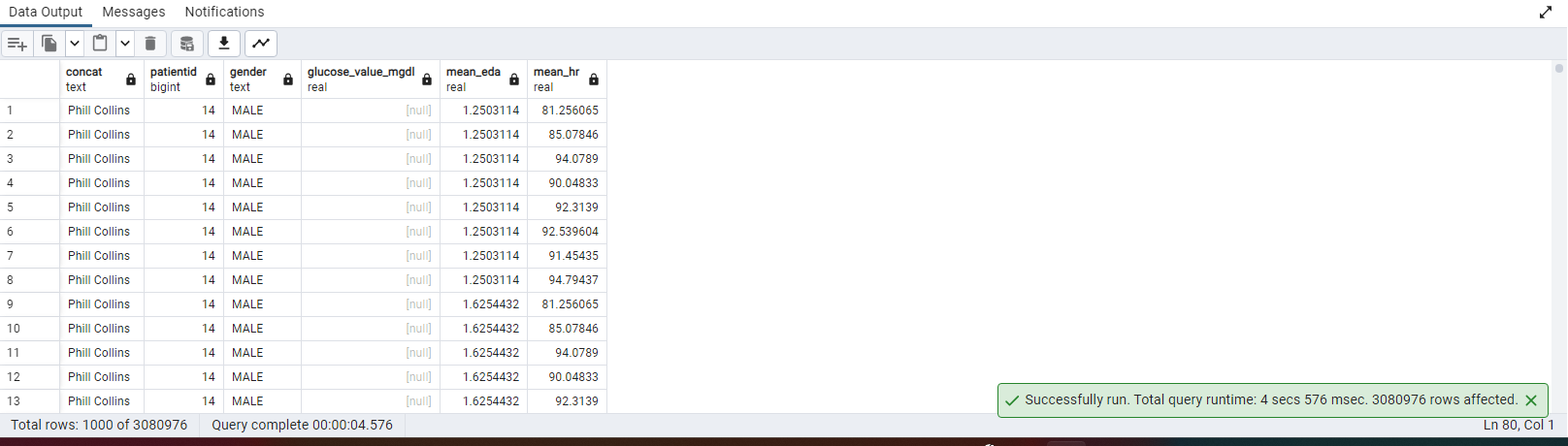
dexcom.patientid = demographics.patientid

and eda.patientid =demographics.patientid

and hr.patientid = demographics.patientid

select \* from Patient\_Overview

**OUTPUT**



**Q 57:Create a table that stores an array of biomarkers: Min(Glucose Value), Avg(Mean\_HR), Max(Max\_EDA) for every patient. The result should look like this: (Link in next cell)**

**Query**

create table newtable(

pid int,

biomarkers numeric []

)

insert into newtable

select dexcom.patientid,array[ min(dexcom.glucose\_value\_mgdl),avg(hr.mean\_hr),max(eda.max\_eda)]

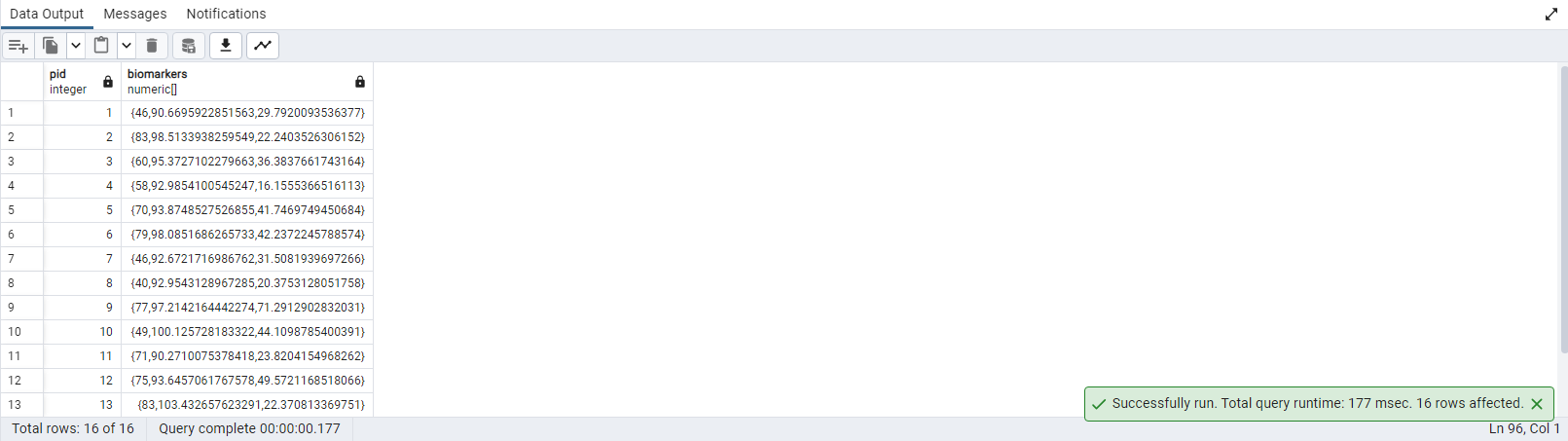
from dexcom inner join hr on dexcom.patientid = hr.patientid inner join eda on hr.patientid=eda.patientid

group by dexcom.patientid

select \* from newtable order by pid

drop table newtable

**OUTPUT**



**Q 58:Assuming lunch is between 12pm and 2pm. Calculate the total number of calories consumed by each patient for lunch on "2020-02-24"**

**Query**

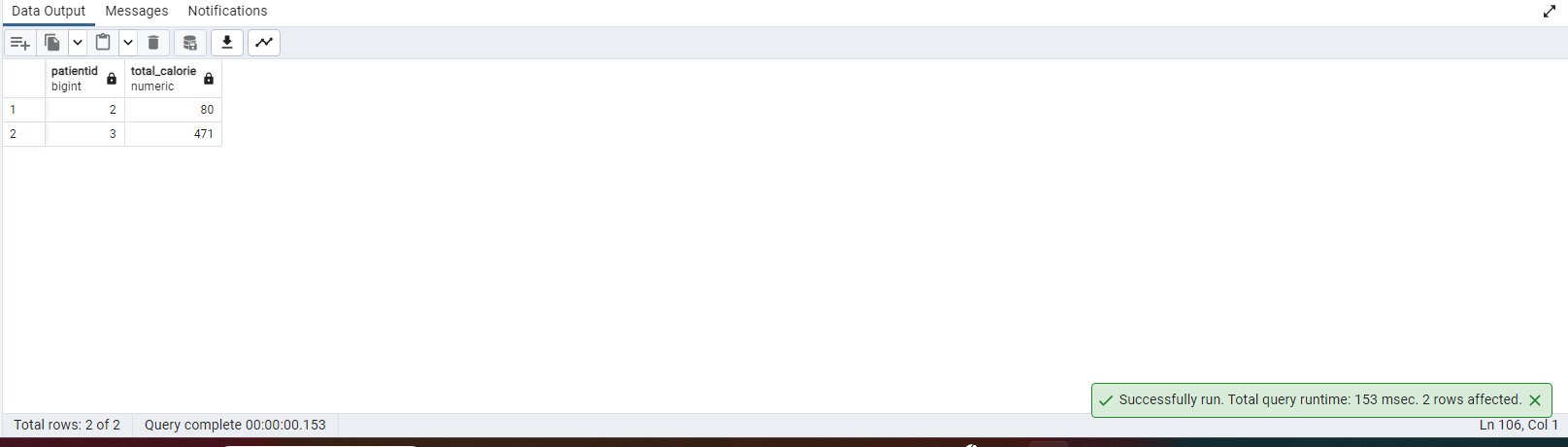
select patientid, sum(calorie) as total\_calorie from foodlog

where

((Extract(hour from datetime) between 12 and 14) and cast(datetime as date) = '2020-02-24')

group by patientid

**OUTPUT**



**Q 59:What is the total length of time recorded for each patient(in hours) in the Dexcom table?**

**Query**

SELECT

patientid,

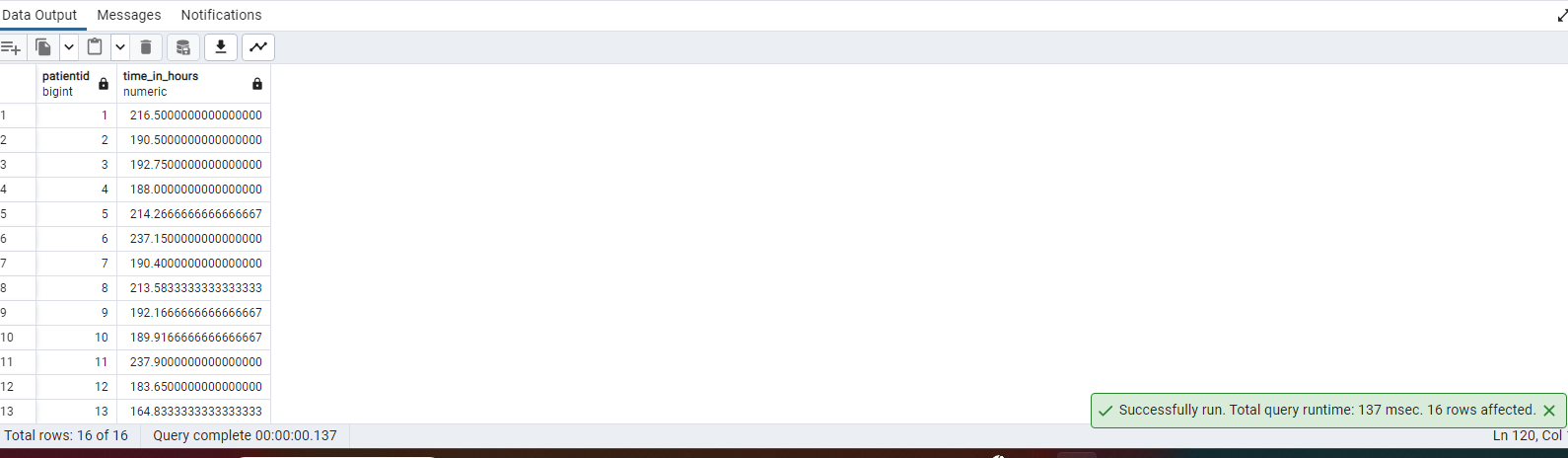
EXTRACT(EPOCH FROM (MAX(datestamp) - MIN(datestamp))) / 3600 AS time\_in\_hours

FROM dexcom

GROUP BY patientid

ORDER BY patientid ASC;

**OUTPUT**



**Q 60:Display the first, last name, patient age and max glucose reading in one string for every patient**

**Query**

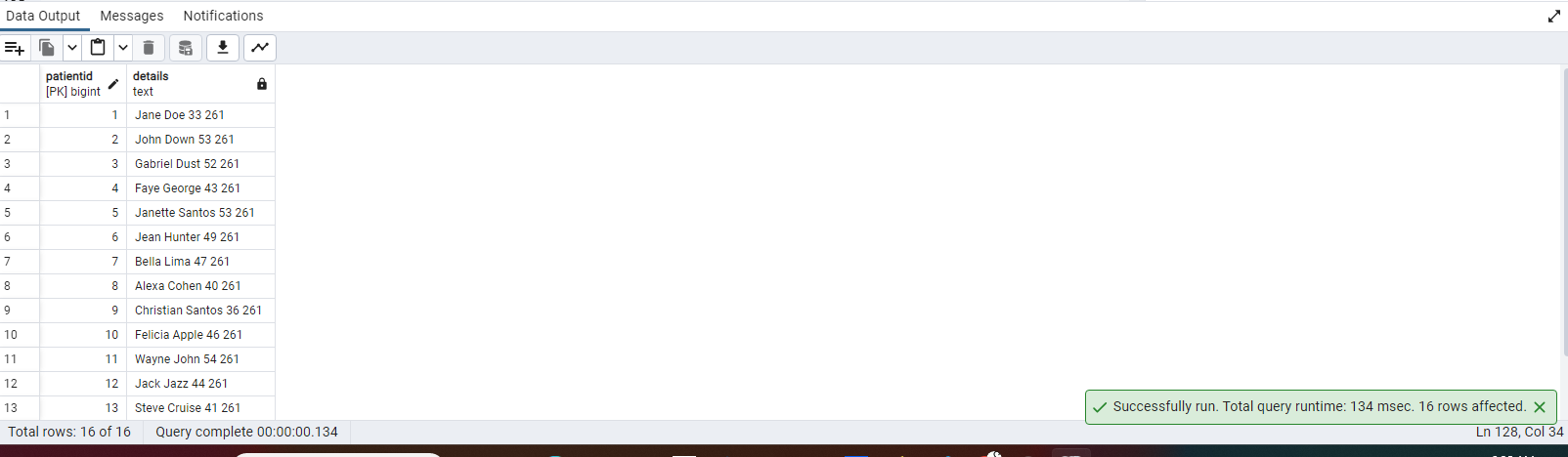
select demographics.patientid, demographics.firstname||' '||demographics.lastname||' '||

DATE\_PART('YEAR', age(current\_date, dob))||' '||max(glucose\_value\_mgdl) as details

from demographics, dexcom

group by demographics.patientid

**OUTPUT**



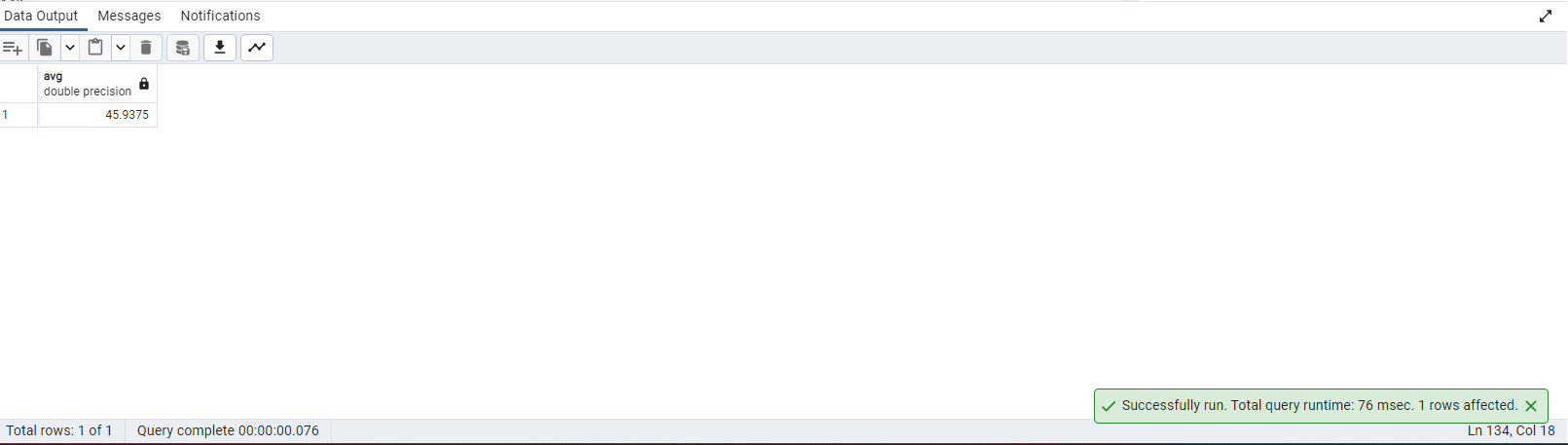
**Q 61:What is the average age of all patients in the database?**

**Query**

select avg(DATE\_PART('YEAR', age(current\_date, dob)))

from demographics

**OUTPUT**



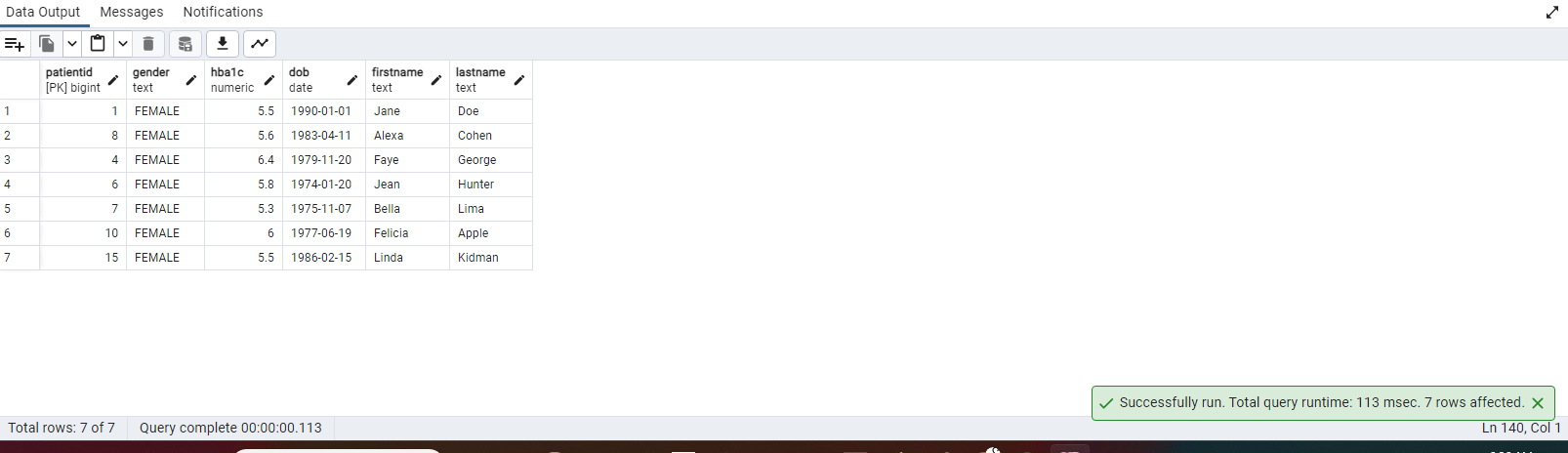
**Q 62:Display All female patients with age less than 50?**

**Query**

select \* from demographics

where gender= 'FEMALE' and DATE\_PART('YEAR', age(current\_date, dob))< 50

**OUTPUT**



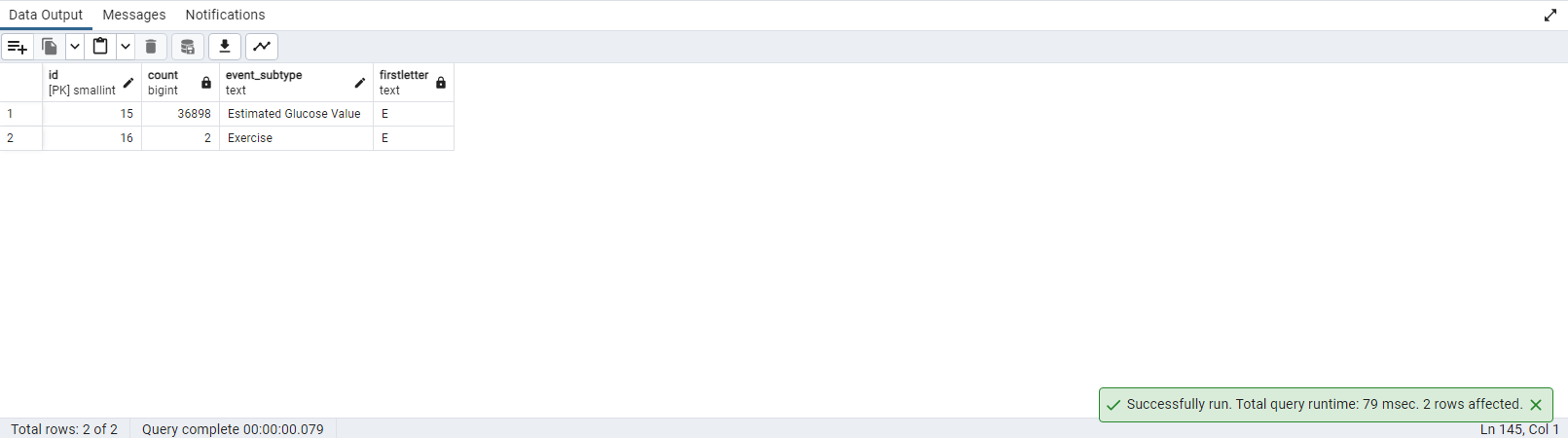
**Q 63:Display count of Event ID, Event Subtype and the first letter of the event subtype. Display all events**

Query

select eventtype.id,count(eventtype.id), eventtype.event\_subtype, left(eventtype.event\_subtype,1) as firstletter

from eventtype inner join dexcom on eventtype.id=dexcom.eventid group by eventtype.id

**OUTPUT**



**Q 64:List the foods consumed by the patient(s) whose eventype is "Estimated Glucose Value**

**Query**

with cte as(

SELECT

dexcom.patientid, eventtype.event\_subtype as eventtype

FROM

dexcom

INNER JOIN eventtype ON dexcom.eventid = eventtype.id

WHERE

eventtype.event\_subtype = 'Estimated Glucose Value')

SELECT

distinct foodlog.patientid,

foodlog.logged\_food,

cte.eventtype

FROM

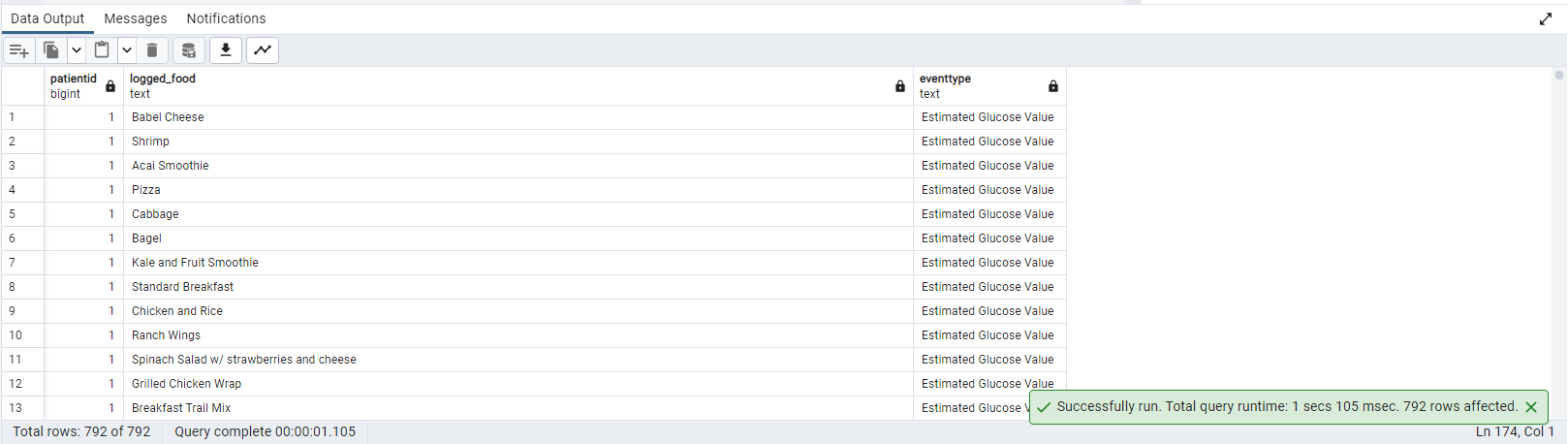
foodlog, cte

WHERE

foodlog.patientid = cte.patientid

order by patientid asc

**OUTPUT**



**Q65: Rank the patients' health based on HRV score and Control of blood sugar(AKA min time spent out of range)**

**Query:**

WITH HRV\_Scores AS (

SELECT

patientid,

ROUND(hrv::numeric, 2) AS hrv\_score

FROM ibi

GROUP BY patientid,hrv

),

Blood\_Sugar\_Control AS (

SELECT

d.patientid,

SUM(CASE

WHEN glucose\_value\_mgdl <55 OR glucose\_value\_mgdl >200 THEN 1

ELSE 0

END) AS time\_out\_of\_range

FROM demographics d

JOIN dexcom g ON d.patientid = g.patientid

GROUP BY d.patientid

)

SELECT

d.patientid,

CONCAT(d.firstname, ' ', d.lastname) AS patients\_full\_name,

COALESCE(h.hrv\_score, 0) AS hrv\_score,

COALESCE(b.time\_out\_of\_range, 0) AS time\_out\_of\_range,

(COALESCE(h.hrv\_score, 0) + COALESCE(b.time\_out\_of\_range, 0)) AS combined\_score,

CASE

WHEN (COALESCE(h.hrv\_score, 0) + COALESCE(b.time\_out\_of\_range, 0)) > 50 THEN 'poor health'

ELSE 'better health'

END AS health\_status,

ROW\_NUMBER() OVER (ORDER BY (COALESCE(h.hrv\_score, 0) + COALESCE(b.time\_out\_of\_range, 0)) ASC) AS health\_rank

FROM demographics d

LEFT JOIN HRV\_Scores h ON d.patientid = h.patientid

LEFT JOIN Blood\_Sugar\_Control b ON d.patientid = b.patientid

ORDER BY combined\_score,patientid ;

**OUTPUT:**

A screenshot of a computer

Description automatically generated

**Q66. Create a trigger on the food log table that warns a person about any food logged that has more than 20 grams of sugar. The user should not be stopped from inserting the row. Only a warning is needed**

**Query:**

**Trigger function:**

CREATE OR REPLACE FUNCTION check\_sugar\_content()

RETURNS TRIGGER AS $$

BEGIN

IF NEW.sugar > 20 THEN

RAISE NOTICE 'Warning: High sugar content([%]) detected in the logged food ', NEW.sugar;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

**OUTPUT:**

A white rectangular object with a white background

Description automatically generated

**Trigger :**

CREATE OR REPLACE TRIGGER foodlog\_sugar\_check

BEFORE INSERT ON foodlog

FOR EACH ROW

EXECUTE FUNCTION check\_sugar\_content();

**OUTPUT**:

A white rectangular object with a white background

Description automatically generated

**-- trying to insert a record with less sugar content should trigger the trigger function**

insert into foodlog(datetime,logged\_food,calorie, total\_carb, dietary\_fiber, sugar,protein,total\_fat,patientid) values(CURRENT\_TIMESTAMP,'Berry juice',456,90,2,45,23,3,1)

**OUTPUT**:

A computer screen with a white background

Description automatically generated

**Q67.Display all the patients with high heart rate and prediabetic Query:**

SELECT

d.patientid,

d.gender,

d.hba1c ,

case

when d.hba1c <5.7 then 'normal'

when d.hba1c >5.7 and d.hba1c<6.4 then 'prediabetic'

when d.hba1c>6.4 then 'diabetic'

else 'invalid range'

end range,

h.mean\_hr AS heart\_rate

FROM

demographics d

JOIN

hr h ON d.patientid = h.patientid

WHERE

h.mean\_hr > 100

AND

d.hba1c BETWEEN 5.7 AND 6.4; -- prediabetic range is between 5.7% and 6.4% as mentioned ----in data definition

**OUTPUT:**

A close-up of a computer screen

Description automatically generated

**Q68. Display patients information who have tachycardia HR and a glucose value greater than 200. Query:**

SELECT

d.patientid,

d.gender,

h.mean\_hr as tachycardia\_HR,

glucose\_value\_mgdl as glucose\_level

FROM

demographics d

JOIN

hr h ON d.patientid=h.patientid

JOIN

dexcom dx ON dx.patientid=d.patientid

WHERE

h.mean\_hr>100 and dx.glucose\_value\_mgdl >200;

**OUTPUT:**

A white rectangular object with a black border

Description automatically generated

**Q69. Calculate the number of hypoglycemic incident per patient per day where glucose drops under 55 Query:**

SELECT

d.patientid,

DATE(dx.datestamp) AS day,

SUM(CASE WHEN dx.glucose\_value\_mgdl < 55 THEN 1 ELSE 0 END) AS hypoglycemic\_incidents

FROM

demographics d

JOIN

dexcom as dx ON d.patientid = dx.patientid

GROUP BY

d.patientid, DATE(dx.datestamp)

HAVING

SUM(CASE WHEN dx.glucose\_value\_mgdl < 55 THEN 1 ELSE 0 END) > 0

ORDER BY

d.patientid, day;

**OUTPUT:**

A close-up of a computer screen

Description automatically generated

**Q70. List the day wise calories intake for each patient.**

**Query:**

WITH UniqueDates AS (

SELECT DISTINCT patientid, datetime::DATE AS unique\_date,

sum(calorie)::NUMERIC AS cal

FROM foodlog

group by patientid, unique\_date

order by patientid

)

SELECT

f.patientid,

ud.unique\_date,

'Day' || ROW\_NUMBER() OVER (PARTITION BY ud.patientid ORDER BY ud.unique\_date) AS patient\_day,

ud.cal AS total\_calories

FROM

uniquedates ud

JOIN

foodlog f ON f.patientid = ud.patientid

GROUP BY

f.patientid,ud.unique\_date,ud.patientid,ud.cal

ORDER BY

f.patientid;

**OUTPUT**:

A screenshot of a computer

Description automatically generated

**Q71. Display the demographic details for the patient that had the maximum time below recommended blood glucose range**

**Query:**

WITH time\_below\_range AS (

SELECT

patientid,

SUM(CASE WHEN glucose\_value\_mgdl<60 THEN 1 ELSE 0 END) AS count\_below\_range -- count(patientid) AS time\_below\_range

FROM

dexcom

GROUP BY

patientid

)

SELECT

d.patientid,

d.gender,

d.hba1c,

d.dob

FROM

time\_below\_range tbr

JOIN

demographics d ON tbr.patientid = d.patientid

WHERE

tbr.count\_below\_range = (SELECT MAX(count\_below\_range) FROM time\_below\_range);

**OUTPUT:**

A white rectangular object with a black border

Description automatically generated

**Q72. How many patients have a minimum HR below the medically recommended level? Query:**

SELECT count(\*) as minimum\_hr\_patient\_count

FROM

hr h

JOIN

demographics d ON d.patientid=h.patientid

WHERE

h.min\_hr <60

ORDER BY

minimum\_hr\_patient\_count asc

**OUTPUT:**

A white rectangular object with a black border

Description automatically generated

**Q73. Create a trigger to raise notice and prevent the deletion of a record from ‘Patient\_Overview’ .**

**Create View:**

CREATE or replace VIEW Patient\_Overview AS

SELECT

d.patientid,

concat (d.firstname,' ',d.lastname),

d.gender,

d.hba1c,

d.dob,

ROUND(CAST(h.mean\_hr AS NUMERIC), 2) AS average\_heart\_rate,

coalesce(g.glucose\_value\_mgdl,0) AS last\_glucose\_reading

FROM demographics d

LEFT JOIN hr h ON d.patientid = h.patientid

LEFT JOIN dexcom g ON d.patientid = g.patientid

A white rectangular object with a white border

Description automatically generated

**Trigger Function:**

-- Create an INSTEAD OF DELETE trigger on the 'demographics' table

CREATE OR REPLACE FUNCTION prevent\_patient\_overview\_deletion() RETURNS TRIGGER AS $$

BEGIN

RAISE NOTICE 'Deletion from the "Patient\_Overview" view is not allowed. Delete records from the underlying tables.';

RETURN NULL;

END;

$$ LANGUAGE plpgsql;

**OUTPUT:**

A white rectangular object with a black border

Description automatically generated

**Trigger**

-- Create the INSTEAD OF DELETE trigger

CREATE or REPLACE TRIGGER prevent\_patient\_overview\_delete\_trigger

INSTEAD OF DELETE ON Patient\_Overview

FOR EACH ROW

EXECUTE FUNCTION prevent\_patient\_overview\_deletion();

**OUTPUT:**

A white rectangular object with a blue border

Description automatically generated

**Delete Event:**

delete from patient\_overview where patientid=2

**OUTPUT:**

A screenshot of a computer

Description automatically generated

**Q74. What is the average heart rate, age and gender of the every patient in the dataset?**

**Query**:

SELECT

d.patientid,

d.gender,

date\_part('year',age(d.dob)) as patient\_age,

-- EXTRACT(year FROM age(current\_date,d.dob)) :: int as age-- other way to extract age

round(cast(AVG(h.mean\_hr) as NUMERIC), 2) AS patient\_average\_heart\_rate

FROM

demographics d

JOIN

hr h ON d.patientid = h.patientid

GROUP BY

d.patientid, d.gender,patient\_age

ORDER BY

d.patientid;

**OUTPUT:**

A screenshot of a computer

Description automatically generated

**Q75. What is the daily total calories consumed by every patient?**

**Query**

WITH UniqueDates AS (

SELECT DISTINCT patientid, datetime::DATE AS unique\_date,

sum(calorie)::NUMERIC AS cal

FROM foodlog

group by patientid, unique\_date

order by patientid

)

SELECT

f.patientid,

ud.unique\_date,

'Day' || ROW\_NUMBER() OVER (PARTITION BY ud.patientid ORDER BY ud.unique\_date) AS patient\_day,

ud.cal ||' '||'calories' AS total\_calories

FROM

uniquedates ud

JOIN

foodlog f ON f.patientid = ud.patientid

GROUP BY

f.patientid,ud.unique\_date,ud.patientid,ud.cal

ORDER BY

f.patientid;

**OUTPUT:**

A white rectangular object with a black border

Description automatically generated

**Q76. Write a query to classify max EDA into 5 categories and display the number of patients in each category.**

**Query**:

-- Category Table for EDA

WITH EDA\_Category AS (

SELECT eda.patientid, eda.max\_eda,

CASE

WHEN eda.max\_eda <= 10 THEN 'Low'

WHEN eda.max\_eda > 10 AND eda.max\_eda <= 20 THEN 'Moderate'

WHEN eda.max\_eda > 20 AND eda.max\_eda <= 30 THEN 'High'

WHEN eda.max\_eda > 30 AND eda.max\_eda <= 40 THEN 'Very High'

ELSE 'Extreme'

END AS EDAcategory

FROM eda

)

-- Number of patients in each category display

SELECT EDAcategory, COUNT(patientid) AS number\_of\_patients

FROM EDA\_Category

GROUP BY EDAcategory

ORDER BY EDAcategory;

**output:**

A white rectangular object with a black border

Description automatically generated

**Q77.List the daily max HR for patient with event type Exercise.**

**Query:**

SELECT

h.patientid,

e.event\_type,

h.datestamp AS date,

MAX(h.max\_hr) AS daily\_max\_hr

FROM

hr h

JOIN

demographics d ON h.patientid = d.patientid

JOIN

dexcom dx ON dx.patientid=d.patientid

JOIN

eventtype e ON e.id=dx.eventid

WHERE

e.event\_type = 'Exercise'

GROUP BY

h.patientid, e.event\_type, h.datestamp

ORDER BY

h.patientid, h.datestamp;

**OUTPUT:**

A white rectangular object with a black border

Description automatically generated

**Q78.What is the standard deviation from mean for all patients for the table HR?**

**Query:**

SELECT STDDEV(mean\_hr) AS standard\_deviation

FROM hr h

**OUTPUT:**

A white rectangular object with a white border

Description automatically generated

**Q79.Give the demographic details of the patient with event type ID of 16.**

**Query**:

SELECT d.patientid, d.gender, d.hba1c, d.dob, d.firstname, d.lastname

FROM demographics d

JOIN dexcom dx ON d.patientid = dx.patientid

JOIN eventtype e ON dx.eventid = e.id

WHERE e.id = 16;

**OUTPUT:**

A white rectangular object with a black border

Description automatically generated

**Q80.Display list of patients along with their gender having a tachycardia mean HR.**

**Query**

SELECT d.patientid,d.firstname, d.lastname,d.gender, h.mean\_hr as heartrate\_greaterthen100

FROM demographics d

JOIN hr h ON d.patientid=h.patientid

WHERE h.mean\_hr>100 and h.mean\_hr<250

**OUTPUT:**

A white rectangular object with a white border

Description automatically generated