**1. What is the median in EDSS levels for all patients**

SELECT

PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY edss\_in\_cat2) AS median

FROM patient\_msdetails;

A screenshot of a computer

Description automatically generated

**2.Rank the types of MS, partioned by age group , by the sum of all comorbidities associated with each one.**

SELECT

ms\_type2,

age\_group,

sum\_comorbidities,

RANK() OVER (PARTITION BY age\_group ORDER BY sum\_comorbidities DESC) AS rank\_within\_category

FROM (

SELECT

pd.ms\_type2,

pd.age\_group,

SUM(com\_cardiovascular\_disease) + SUM(com\_chronic\_kidney\_disease) + SUM(com\_chronic\_liver\_disease) + SUM(com\_diabetes) +

SUM(com\_hypertension) + SUM(com\_immunodeficiency) + SUM(com\_lung\_disease) + SUM(com\_malignancy) + SUM(com\_neurological\_neuromuscular) AS sum\_comorbidities

FROM

patient\_details AS pd

INNER JOIN

comorbidities AS co ON pd.patient\_id = co.patient\_id

GROUP BY

pd.ms\_type2, pd.age\_group

) AS subquery

A screenshot of a computer

Description automatically generated

**3. DOnut chart showing age category and covid affected.**

CREATE MATERIALIZED VIEW covidagedistribution AS

SELECT pd.age\_group, COUNT(pd.patient\_id) FROM patient\_details pd JOIN covid\_details cd ON pd.patient\_id=cd.patient\_id

WHERE cd.covid19\_diagnosis='confirmed' GROUP BY pd.age\_group

SELECT \*FROM covidagedistribution

**A screenshot of a computer

Description automatically generated**

**----Ppsql tool----**

*\set width 80*

*\set height 35*

*\set radius 1.0*

*\set colours '''#;o:X"@+-=123456789abcdef'''*

*WITH slices AS (*

*SELECT CAST(*

*row\_number() over () AS integer) AS slice,name,value,100.0 \* value / sum(value) OVER () AS percentage,*

*2\*PI() \* sum(value) OVER (rows unbounded preceding) / sum(value) OVER () AS radians*

*FROM*

*(select age\_group,count from covidagedistribution*

*) AS data(name,value))*

*(*

*SELECT array\_to\_string(array\_agg(c),'') AS donut\_chart*

*FROM (*

*SELECT x, y,*

CASE WHEN NOT (sqrt(pow(x, 2) + pow(y, 2)) BETWEEN 0.6 AND :radius)

THEN ' '

*ELSE substring(:colours,*

*(select min(slice) from slices where radians >= PI() + atan2(y,-x)), 1)*

*END AS c*

*FROM (SELECT 2.0\*generate\_series(0,:width)/:width-1.0) AS x(x),*

*(SELECT 2.0\*generate\_series(0,:height)/:height-1.0) AS y(y)*

*ORDER BY y,x*

*) AS xy*

*GROUP BY y*

*ORDER BY y*

*)*

*UNION ALL*

*SELECT repeat(substring(:colours,slice,1), 2) || ' ' ||*

*name || ': ' ||*

*value || ' (' || round(percentage,0) || '%)'*

*FROM slices;*

A circle with small dots

Description automatically generated with medium confidence

Ref: https://www.numpyninja.com/amp/sql-pie-charts-and-doughnut-charts

**4.Data histogram shows the female and male covid confirmed cases and its percentage in age groups.**

WITH patientcount\_agegroup AS (

SELECT

age\_group AS agegroup,

sex AS gender,

COUNT(patient\_id) AS total\_count

FROM

patient\_details

GROUP BY

age\_group,

sex

),

covid\_patientcount\_agegroup AS (

SELECT

pd.age\_group AS agegroup,

pd.sex AS gender,

COUNT(cd.patient\_id) AS covid\_count

FROM

patient\_details pd

JOIN

covid\_details cd ON cd.patient\_id = pd.patient\_id AND cd.covid19\_confirmed\_case = 1

GROUP BY

pd.age\_group,

pd.sex

)

SELECT

CONCAT(p.agegroup||','||p.gender) AS GROUPNAME,

c.covid\_count,

p.total\_count,

ROUND(c.covid\_count\*100.0 / p.total\_count,2) AS percentage\_affected

FROM

patientcount\_agegroup p

JOIN

covid\_patientcount\_agegroup c ON p.agegroup = c.agegroup AND p.gender = c.gender

ORDER BY

p.agegroup,

p.gender; A screenshot of a graph

Description automatically generated

**5.Display 5 random records from patient table in age group 18-50 in the increasing order of their BMI**

--first run

WITH randomrecords AS(

SELECT \*FROM patient\_details WHERE age\_group='18-50' ORDER BY RANDOM() LIMIT 5)

SELECT \*FROM randomrecords ORDER BY bmi\_in\_cat2

A screenshot of a computer

Description automatically generated

--second run

WITH randomrecords AS(

SELECT \*FROM patient\_details WHERE age\_group='18-50' ORDER BY RANDOM() LIMIT 5)SELECT \*FROM randomrecords ORDER BY bmi\_in\_cat2

A screenshot of a computer

Description automatically generated

**6.details of first patient admitted in every month**

WITH FirstAdmissionofMonth AS (

SELECT patient\_id,

TO\_CHAR(date\_of\_onset, 'MONTH') AS Month\_name,

EXTRACT(YEAR FROM date\_of\_onset) AS yea,

FIRST\_VALUE(patient\_id) OVER (

PARTITION BY EXTRACT(YEAR FROM date\_of\_onset), EXTRACT(MONTH FROM date\_of\_onset)

ORDER BY date\_of\_onset

) AS first\_admiited\_patient,

FIRST\_VALUE(date\_of\_onset) OVER (

PARTITION BY EXTRACT(YEAR FROM date\_of\_onset), EXTRACT(MONTH FROM date\_of\_onset)

ORDER BY date\_of\_onset

) AS admission\_date

FROM covid\_details

)

SELECT DISTINCT CONCAT(yea||'-'||month\_name) as mon,first\_admiited\_patient AS First\_Admitted\_Patient\_Id,

admission\_date

FROM FirstAdmissionofMonth

ORDER BY admission\_date

A screenshot of a computer

Description automatically generated

**7.details of first patient admitted in every month**

WITH firstadmissionofthemonth AS (

SELECT patient\_id,

TO\_CHAR(date\_of\_onset, 'MONTH') AS Month\_name,

EXTRACT(YEAR FROM date\_of\_onset) AS yea,

ROW\_NUMBER() OVER (

PARTITION BY EXTRACT(YEAR FROM date\_of\_onset), EXTRACT(MONTH FROM date\_of\_onset)

ORDER BY date\_of\_onset

) AS rownumber

FROM covid\_details

),

firstadmittedpatients AS (

SELECT CONCAT(yea||'-'||month\_name) as mon,patient\_id

FROM firstadmissionofthemonth

WHERE rownumber = 1

)

SELECT fap.mon,pd.\*

FROM patient\_details pd

RIGHT JOIN firstadmittedpatients fap ON pd.patient\_id = fap.patient\_id

A screenshot of a computer

Description automatically generated

**8.barchart shows patients admitted (Jan-Dec)month**

SELECT

EXTRACT(MONTH FROM date\_of\_onset) AS month,

COUNT(patient\_id) AS no\_of\_patients

FROM

covid\_details

GROUP BY

month

ORDER BY

month;

A screenshot of a graph

Description automatically generated

**9.Update the ms\_type column in patient\_details table with title case**

UPDATE patient\_details SET ms\_type2 = INITCAP(ms\_type2) RETURNING\*

A screenshot of a computer

Description automatically generated

**10.Create a trigger to warn invalid edss score**

CREATE OR REPLACE FUNCTION warn\_invalid\_edssscore()

RETURNS TRIGGER AS $$

BEGIN

IF NEW.edss\_in\_cat2>10

THEN RAISE WARNING 'WARNING: INVALID EDSS SCORE, CAN NOT INSERT THE VALUE';

END IF;

END; $$

LANGUAGE plpgsql;

-- Create Trigger

CREATE OR REPLACE TRIGGER warn\_invalid\_edss

BEFORE INSERT

ON patient\_msdetails

FOR EACH ROW

EXECUTE FUNCTION warn\_invalid\_edssscore();

-- Insert a row

INSERT INTO patient\_msdetails(patient\_id, edss\_in\_cat2)

VALUES('P\_1', 13);

A screenshot of a computer

Description automatically generated

**11.show a histogram of covid symptoms distribution among all patients**

select count(case when covid19\_sympt\_chills = 1 then 1 end) as chills,

  count(case when covid19\_sympt\_dry\_cough = 1 then 1 end) as dryCough,

  count(case when covid19\_sympt\_fatigue = 1 then 1 end) as fatigue,

  count(case when covid19\_sympt\_fever = 1 then 1 end) as fever,

  count(case when covid19\_sympt\_loss\_smell\_taste = 1 then 1 end) as lost\_smell\_taste,

  count(case when covid19\_sympt\_nasal\_congestion = 1 then 1 end)as nasal\_cong,

  count(case when covid19\_sympt\_pain = 1 then 1 end) as pain,

  count(case when covid19\_sympt\_pneumonia = 1 then 1 end) as pnuemonia,

  count(case when covid19\_sympt\_shortness\_breath= 1 then 1 end) as shortness\_breadth,

  count(case when covid19\_sympt\_sore\_throat = 1 then 1 end) as sore\_throat

from covid\_symptoms

A screenshot of a bar chart

Description automatically generated

**--QS12.Highlight only confirmed and hospitilzed patiets by appending \*\*\* at the end patinet id**

SELECT

CASE

WHEN covid19\_outcome\_levels\_2 ='Confirmed & Hospitalized' THEN RPAD(patient\_id,length(patient\_id)+3,'\*\*\*')

ELSE patient\_id

END AS hospitalized\_patients FROM covid\_details

A screenshot of a computer

Description automatically generated

**QS13. Replace c with clinician and p with patient in patient id**

SELECT

CASE

WHEN patient\_id LIKE 'P%' THEN OVERLAY(patient\_id PLACING 'Patient - ' FROM 1 FOR 2)

WHEN patient\_id LIKE 'C%' THEN OVERLAY(patient\_id pLACING 'Clinician - ' FROM 1 FOR 2)

END AS patientid

FROM patient\_details;

A screenshot of a computer

Description automatically generated

**QS14.what is the most common treatment duration years amongst this MS patient dataset?using inbuilt mode function**

**--without using groupby**

select mode() within group(order by duration\_treatment\_cat2) as treatment\_dur\_mode from patient\_msdetails

A screenshot of a computer

Description automatically generated

**QS15.Regression analysis on edss category and duration of treatment.**

SELECT regr\_slope(edss\_in\_cat2,duration\_treatment\_cat2) AS slope,

regr\_intercept(edss\_in\_cat2,duration\_treatment\_cat2) AS intercept

FROM

patient\_msdetails

A screenshot of a computer

Description automatically generated

**QS16.Find the number of patients who have all of the following symptoms together: nasal congestion, dry cough, pain, and shortness of breath by using BIT\_AND**

SELECT COUNT(\*)

FROM covid\_symptoms

WHERE

(covid19\_sympt\_nasal\_congestion::bit & covid19\_sympt\_dry\_cough::bit &

covid19\_sympt\_pain::bit & covid19\_sympt\_shortness\_breath::bit)::integer = 1;)

A screenshot of a computer

Description automatically generated