1. Number of patients in ICU for different Primary Diagnosis using first\_value() Window Function

we are using first\_value() Window Function.

Query:

SELECT

"PrimaryDiagnosis",Count("Patient\_ID"),

FIRST\_VALUE("Service")

OVER(

ORDER BY Count("Patient\_ID") desc

) Service

FROM

public."ReAdmissionRegistry" rvis

join public."Service" ser on ser."Service\_ID" = rvis."Service\_ID"

join public."PrimaryDiagnosis" pd on pd."Diagnosis\_ID" = rvis."Diagnosis\_ID"

where "Service" = 'ICU'

group by "PrimaryDiagnosis",ser."Service";

Table

Description automatically generated

1. Find the percentage of Kidney Failure as per gender

Query:

--To create a function which returns the count of the patients with the given DiagnosisID.

create function get\_PatientCount\_with( diagId text)

returns int

language plpgsql

as

$$

declare

PatientCount integer;

begin

select count("Patient\_ID")

into PatientCount

from "Discharges"

where "Diagnosis\_ID" = diagId;

return PatientCount;

end;

$$;

select ('Kidney Failure in ' || g."Gender" ),count(d."Patient\_ID") as "NoOfPatients",

(count(d."Patient\_ID")\*100)::float/(select get\_PatientCount\_with('PD011')) as "Percentage%"

from "Discharges" as d

join "Patients" as p on d."Patient\_ID" = p."Patient\_ID"

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

where "Diagnosis\_ID" = 'PD011'

group by "Gender"

Graphical user interface, text, application, email

Description automatically generated

3. Get the 3rd ProviderSpeciality based on number of patients.

Query:

With "cte\_specialityRank" as (

select av."Patient\_ID",av."Provider\_ID",p."PS\_ID",ps."ProviderSpeciality"

from "AmbulatoryVisits" as av

join "Providers" as p on av."Provider\_ID" = p."Provider\_ID"

join "ProviderSpeciality" as ps on p."PS\_ID" = ps."PS\_ID"

order by p."PS\_ID")

select \* from (select count("Patient\_ID") as "NumberOfPatients","ProviderSpeciality",

ROW\_NUMBER() over(order by count("Patient\_ID") desc ) as "OrderOfSpeciality"

from "cte\_specialityRank"

group by "ProviderSpeciality") as subtable

where "OrderOfSpeciality" = 3

Graphical user interface, application

Description automatically generated

4. Create view to get the size of HospitalDB\_New database

Query:

--create view

Create or replace view hospDBDetails as

SELECT PG\_SIZE\_PRETTY (PG\_DATABASE\_SIZE ('HospitalDB\_New'));

--Select View

select \* from hospDBDetails;

Graphical user interface, text, application

Description automatically generated

5. Without using any function ,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 public."Providers" where "ProviderName" ilike ('h\_r\_y%');

Graphical user interface, text, application

Description automatically generated

6.Find the major Primary diagnosis faced by patients over age 50.

Query:

Select T."PatientCount",T."PrimaryDiagnosis" from

(Select(count(P1."Patient\_ID")) as "PatientCount",P."PrimaryDiagnosis" from

"Patients" as P1

Left Join "Discharges" D ON P1."Patient\_ID"= D."Patient\_ID"

Left Join "PrimaryDiagnosis" P ON D."Diagnosis\_ID" = P."Diagnosis\_ID"

Where (date\_part('year',AGE(D."AdmissionDate",P1."DateOfBirth"))) > 50

Group by P."PrimaryDiagnosis" order by "PatientCount" desc) T limit 1

Graphical user interface, application

Description automatically generated

7.Write a Query to list all the Telemedicine Patients in the hospitaldb

Query:

Select distinct(p."Patient\_ID"),p."FirstName",p."LastName",V."VisitType"

from public."Patients" p

join public."AmbulatoryVisits" A on A."Patient\_ID" = p."Patient\_ID"

join public."VisitTypes" V on V."AMVT\_ID" = A."AMVT\_ID"

where "VisitType" = 'Telemedicine'

order by p."Patient\_ID"

Table

Description automatically generated