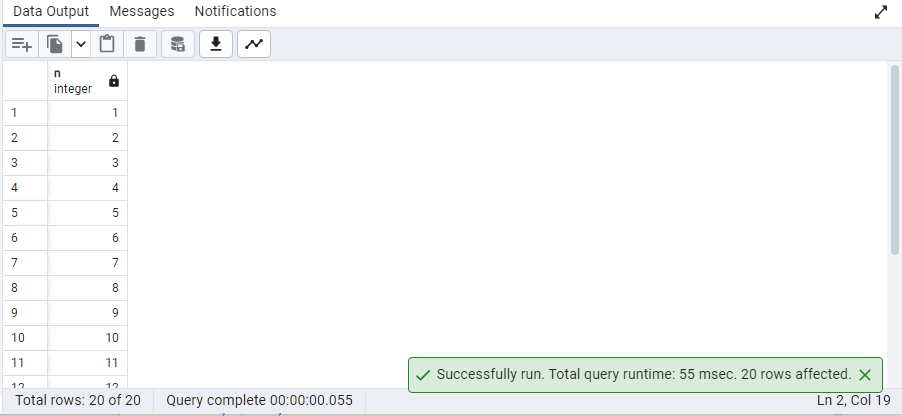
**DIABETIC DATASET ANALYSIS USING SQL**

|  |  |
| --- | --- |
| 1. **Display any 10 random DM patients.**   **QUERY :**  SELECT PA."Firstname",PA."Lastname" FROM "Patients" PA,"Group" GR where GR."Group"='DM' ORDER BY RANDOM() LIMIT 10;  **OUTPUT :** | |
| 1. **Please go through the below screenshot and create the exact output.**     **QUERY :**  SELECT CONCAT(PA."Firstname",' ',PA."Lastname") AS Full\_name  FROM "Patients" PA  WHERE PA."Lastname" LIKE 'Ma%'  **OUTPUT** **:** | |
| 1. **Write a query to get a list of patients whose RPE start is at moderate intensity.**   **QUERY :**  select "Firstname", "Lastname","Gait\_RPE\_Start " from "Walking\_Test"  inner join "Patients" using("WalkTest\_ID")  Where "Gait\_RPE\_Start " in (4,5,6);  **OUTPUT** **:** | |
| 1. **Write a query by using common table expressions and case statements to display birthyear ranges.**   **QUERY :**  WITH CTE  as (SELECT "Visit\_Date",  case  when extract(Year from "Visit\_Date") -"Age">=1928 and extract(Year from  "Visit\_Date") -"Age"<=1938 THEN '1928-1938'  when extract(Year from "Visit\_Date") -"Age">=1939 and extract(Year from  "Visit\_Date") -"Age"<=1948 THEN '1939-1948'  when extract(Year from "Visit\_Date") -"Age">=1949 and extract(Year from  "Visit\_Date") -"Age"<=1958 THEN '1949-1958'  ELSE 'not in range' end "Birth\_year\_Range"  from "Patients") select distinct "Birth\_year\_Range",count(\*) as  "No\_Of\_Patients" from CTE  group by "Birth\_year\_Range"  order by "Birth\_year\_Range"    **OUTPUT** **:** | |
| 1. **Display DM patient names with highest day MAP and night MAP (without using limit).**   **QUERY :**  SELECT PA."Patient\_ID", GR. "Group",PA."Firstname",  PA."Lastname",  cast(max(((bp."24Hr\_Day\_SBP" - bp."24Hr\_Day\_DBP")/3)  + bp."24Hr\_Day\_DBP") as decimal(16,2)) as  Highest\_Day\_MAP,  cast(max(((bp."24Hr\_Night\_SBP"-  bp."24Hr\_Night\_DBP")/3)+bp."24Hr\_Night\_DBP")as  decimal(16,2)) as highest\_night\_MAP  from "Patients" PA  JOIN "Blood\_Pressure"bp  on bp."Patient\_ID"=PA."Patient\_ID"  JOIN "Group" GR  on PA."Group\_ID"=GR."Group\_ID"  where GR."Group\_ID"='GRP\_02'  group by PA."Patient\_ID", PA."Firstname", PA."Lastname",  GR."Group\_ID"  order by Highest\_Day\_MAP desc, highest\_night\_MAP  **OUTPUT** **:** | |
| 1. **Create view on table Lab Test by selecting some columns and filter data using Where condition.**   **QUERY:**  CREATE VIEW High\_fASTING\_Glucose as  SELECT "Lab\_ID","Patient\_ID","Fasting\_Glucose"  FROM "Lab\_Test"  WHERE "Fasting\_Glucose">90;  SELECT \*  FROM High\_fASTING\_Glucose;  **OUTPUT** : | |
| 1. **Display a list of Patient IDs and their Group whose diabetes duration is greater than 10 years.**   **QUERY :**  SELECT "Patient\_ID", "Group"  FROM "Patients" p  INNER JOIN "Group" g  ON p."Group\_ID" = g."Group\_ID"  WHERE "Diabetes\_Duration">10;  **OUTPUT** **:** | |
| 1. **Write a query to list male patient ids and their names who are above 40 years of age and less than 60 years and have Day BloodPressureSystolic above 120 and Day BloodPressureDiastolic above 80.**   **QUERY :**  SELECT "Firstname", "Lastname", "Age","24Hr\_Day\_SBP", "24Hr\_Day\_DBP","Gender",pt."Patient\_ID"  FROM "Patients"pt  INNER join "Gender" using("Gender\_ID")  INNER join "Blood\_Pressure" using("BP\_ID")  WHERE "Gender" ='Male' and "Age" >40 and "Age"<60 AND "24Hr\_Day\_DBP" >80  and "24Hr\_Day\_SBP">120  **OUTPUT** **:** | |
|  | |
| 1. **Use a function to calculate the percentage of patients according to the lab visited per month.**   **QUERY :**  WITH lvm AS (SELECT DISTINCT TO\_CHAR("Lab\_Visit\_Date", 'Mon')AS "Month Name",count(\*) AS "Number of Patients visited"  ,COUNT(DISTINCT "Lab\_visit\_ID") AS pat\_visited  FROM "Lab\_Visit"  GROUP BY "Month Name")  SELECT  "Month Name",round(pat\_visited \* 100.0 /(SELECT COUNT(\*) FROM "Lab\_Visit"),2) AS "percentage of Patients"  FROM lvm;  **OUTPUT** **:** | |
|  | 1. **Count of patients by first letter of firstname.**   **QUERY :**  SELECT SUBSTR("Firstname",1,1)AS Firstletter,  COUNT(\*) AS Patients  FROM "Patients"  GROUP BY Firstletter  ORDER BY Firstletter ;  **OUTPUT :** |
| 1. **Write a query to get the list of patients whose lipid test value is null.**   **QUERY :**  SELECT  pt."Firstname",  pt."Lastname"  FROM  "Patients" AS pt  INNER JOIN "Lipid\_Lab\_Test" AS lipid ON pt."Patient\_ID" = lipid."Patient\_ID"  WHERE  (  "Fasting\_Cholestrol" IS NULL  OR "Fasting\_Triglyc" IS NULL  OR "Fasting\_HDL" IS NULL  OR "Fasting\_LDL" IS NULL )  **OUTPUT** : | |
| 1. **Create a stored procedure to make user ids for the given patient id.**   **QUERY :**  CREATE OR REPLACE PROCEDURE makeuserid( patient\_id IN text, result inout refcursor )  LANGUAGE 'plpgsql'  AS $$  BEGIN  OPEN result FOR  SELECT floor(random()\*(100-10+1))+10 AS "UserID" ,"Patient\_ID"  FROM "Patients" WHERE "Patient\_ID" = patient\_id;  END;  $$;    CALL makeuserid ('S0030','result');  FETCH ALL IN "result";  **OUTPUT** **:** | |
| 1. **Show the position of the letter 'r' in the Patient's name.**   **QUERY :**    with cte\_fname as ( SELECT "Patient\_ID", "Firstname","Lastname",  STRING\_AGG(pos\_r\_fn, ', ' ) as "Position\_of\_r\_Firstname"  FROM ( SELECT DISTINCT pt."Patient\_ID", pt."Firstname", pt."Lastname",  CAST(uf."the\_pos" AS VARCHAR(10)) as pos\_r\_fn  FROM "Patients" AS pt JOIN UNNEST(string\_to\_array(pt."Firstname", NULL))  WITH ORDINALITY uf(elem, the\_pos) ON uf.elem = 'r'  WHERE (pt."Firstname" LIKE '%r%')) AS a  GROUP BY "Patient\_ID","Lastname","Firstname"),  cte\_lname as ( SELECT "Patient\_ID", "Firstname", "Lastname",  STRING\_AGG(pos\_r\_ln, ', ' ) as "Position\_of\_r\_Lastname"  FROM(SELECT DISTINCT pt."Patient\_ID", pt."Firstname", pt."Lastname",  CAST(uf."the\_pos" AS VARCHAR(10)) as pos\_r\_ln  FROM "Patients" AS pt JOIN UNNEST(string\_to\_array(pt."Lastname", NULL))  WITH ORDINALITY uf(elem, the\_pos) ON uf.elem = 'r'  WHERE (pt."Lastname" LIKE '%r%')) AS a  GROUP BY "Patient\_ID","Firstname","Lastname")  SELECT COALESCE(f."Firstname") as FirstName,  COALESCE(f."Position\_of\_r\_Firstname",'0') as Position\_of\_r\_Firstname ,  COALESCE(l."Lastname",f."Lastname") as LastName,  COALESCE(l."Position\_of\_r\_Lastname",'0') as Position\_of\_r\_Lastname  FROM cte\_fname f FULL outer join cte\_lname l on f."Patient\_ID" = l."Patient\_ID"  **OUTPUT** **:** | |
| 1. **Calculate the patient's birth year in descending order.**   **QUERY :**  SELECT  "Firstname",  "Lastname",  EXTRACT(year from "Visit\_Date") - "Age" AS "BirthYear"  FROM  "Patients"  ORDER BY  "BirthYear" DESC;  **OUTPUT** **:** | |
| 1. **Find the patients that have eye damage due to diabetes.**   **QUERY :**  SELECT  pt."Patient\_ID",  pt."Firstname",  pt."Lastname"  FROM  "Patients" AS pt  INNER JOIN "Opthalmology" AS op ON pt."Opthal\_ID" = op."Opthal\_ID"  WHERE  "Diabetic\_Retinopathy" > 0 OR "Macular\_Edema" > 0;  **OUTPUT** **:** | |
| 1. **Query to classify Gait RPE End into 5 categories as per the intensity. (Hint: Use of CASE statement)**   **QUERY :**  SELECT "Patient\_ID","Gait\_RPE\_End ",  CASE  WHEN "Gait\_RPE\_End "=0 then 'Rest'  WHEN "Gait\_RPE\_End " between 1 and 3 then 'Easy Intensity'  WHEN "Gait\_RPE\_End " between 4 and 6 then 'Moderate Intensity'  WHEN "Gait\_RPE\_End " between 7 and 9 then 'Hard Intensity'  WHEN "Gait\_RPE\_End "=10 then 'Max Effort intensity'  END intensity  FROM "Walking\_Test"  **OUTPUT :** | |
| 1. **Create view on patient table with check constraint condition.**   **QUERY :**  CREATE VIEW "Age\_equal\_lessthan\_50" as  select "Patient\_ID" from "Patients"  where "Age"<=50  with check option;    **OUTPUT :** | |
| 1. **Calculate the patient's current age.**   **QUERY :**  select "Firstname", "Lastname", "Age"+ date\_part('year',Age(CURRENT\_DATE, "Visit\_Date")) as current\_age from "Patients";  **OUTPUT :** | |
| 1. **Write a query to display Mr. or Ms. as prefix to patients’ names with respect to gender.**   **QUERY :**  Select "Patient\_ID",  case "Gender"  when 'Male' then 'Mr. '||"Firstname"||' '|| "Lastname"  when 'Female' then 'Ms. '||"Firstname"||' ' ||"Lastname"  end Full\_name  from "Patients"  INNER JOIN "Gender" using ("Gender\_ID") ;  **OUTPUT :** | |
| 1. **Write a query to get DM patient names whose distance is greater than**   **400 and speed is greater than 1.**  **QUERY :**  Select "Firstname", "Lastname"  from "Patients"  INNER JOIN "Walking\_Test" using ("WalkTest\_ID")  inner join "Group" using("Group\_ID")  where "Group" = 'DM' and "Gait\_DT\_Distance" > 400 and "Gait\_DT\_Speed" >1 ;  **OUTPUT :** | |
| 1. **Create a trigger to raise notice and prevent the deletion of a record from a view.**   **Query:**  CREATE OR REPLACE VIEW "vw\_Delete" AS SELECT \* FROM "Group";  SELECT \* FROM "vw\_Delete"  CREATE OR REPLACE FUNCTION prevent\_delete()RETURNS TRIGGER AS  $$  BEGIN  RAISE NOTICE 'Deleting records are not allowed from this view';  RETURN null;  END;  $$ LANGUAGE plpgsql;  CREATE OR REPLACE TRIGGER vw\_prevent\_trigger INSTEAD OF DELETE ON "vw\_Delete"  FOR EACH ROW EXECUTE FUNCTION prevent\_delete();  DELETE FROM "vw\_Delete" WHERE "Group\_ID"='GRP\_02';  **Output:** | |
| 1. **Select the patient's full name with a name starting with 's' followed by any character, followed by 'r', followed by any character, followed by b.**   **Query:**  SELECT CONCAT("Firstname",' ',"Lastname") AS Fullname  FROM "Patients" WHERE CONCAT("Firstname",' ',"Lastname") like 's%r%b%' ;  **Output:** | |
| 1. **write a query to get which race has the maximum number of Diabetic patients.**   **Query:**  SELECT r."Race",COUNT(\*) FROM "Race" r JOIN "Patients" p  ON r."Race\_ID"= p."Race\_ID" GROUP BY r."Race"  HAVING COUNT(\*)=(SELECT MAX(p1.cnt)  FROM(SELECT p."Race\_ID",COUNT(\*)cnt  FROM "Patients" p GROUP BY p."Race\_ID")p1);  **Output:** | |
| 1. **Identify the patient count by Gender and Race combination.**   **QUERY :**  SELECT COUNT("Patient\_ID") AS Patient\_Count,  "Gender","Race" FROM "Patients" p  JOIN "Gender" g ON p."Gender\_ID"=g."Gender\_ID"  JOIN "Race" r ON p."Race\_ID"=r."Race\_ID"  GROUP BY "Gender","Race"  ORDER BY "Race";  **OUTPUT :** | |
| 1. **Get the number of patients in the year 2005 in each of the Genesis and Cultivate labs.**   **Query:**  SELECT COUNT(\*) AS Patients,l."Lab\_names" FROM "Patients"p  JOIN "Link\_Reference" lr ON p."Link\_Reference\_ID"=lr."Link\_Reference\_ID"  JOIN "Lab\_Visit"l ON lr."Lab\_visit\_ID"=l."Lab\_visit\_ID"  WHERE EXTRACT(YEAR FROM "Lab\_Visit\_Date")=2005  AND "Lab\_names" IN('Cultivate Lab','Genesis Lab') GROUP BY l."Lab\_names"  **Output:** | |
| 1. **Write a query to get a list of patient IDs' and their Fasting Cholesterol in February 2006.**   **QUERY :**  SELECT PA."Firstname",PA."Lastname" FROM "Patients" PA  JOIN "Lipid\_Lab\_Test" LL ON LL."Patient\_ID"=PA."Patient\_ID"  JOIN "Link\_Reference" LR ON LR."Lipid\_ID"=LL."Lipid\_ID"  JOIN "Lab\_Visit" LV ON LV."Lab\_visit\_ID"=LR."Lab\_visit\_ID"  WHERE LV."Lab\_Visit\_Date">'2006-02-01' AND  LV."Lab\_Visit\_Date"<='2006-02-28'    **OUTPUT** | |
| 1. **Write a query to get a list of patients whose first names is starting with the letter T.**   **QUERY :**  SELECT CONCAT(PA."Firstname",' ',PA."Lastname") AS NAME FROM "Patients"  PA WHERE PA."Firstname" LIKE ('T%')    **OUTPUT** | |
|  | |
| 1. **Find a list of Male patients whose age is more than 60 whose, BMI is more than 18.5, and whose height is more than e 1.5 M.**     **QUERY :**  SELECT DISTINCT PA."Firstname",PA."Lastname", PA."Age",PA."BMI",PA."Height" FROM  "Patients" PA, "Gender" GE WHERE PA."Age">60 AND PA."BMI">18.5 AND  PA."Height">1.5 AND GE."Gender\_ID"='G001'        **OUTPUT** | |
| 1. **Display the patient names and ages whose age is more than 60 years.**     **QUERY :**  SELECT CONCAT(PA."Firstname",' ',PA."Lastname") AS NAME,PA."Age"  FROM "Patients" PA WHERE PA."Age">60 ORDER BY PA."Age"  **OUTPUT :**    **30.Write a query to get the number of patients who visited the Lab between 9 am to 12 am.**    select Count("Lab\_visit\_ID") as patients\_visit\_9am\_12am from "Lab\_Visit"  where extract('hour' from "Lab\_Visit\_Date") between 9 and 24; | |
| **31. Write a trigger that calls a function, for checking space and case for two**  **columns or more before you add new data to a table.**  **Query:**  CREATE OR REPLACE FUNCTION space\_case\_func()RETURNS TRIGGER AS  $$  BEGIN  IF s."Patient\_ID" ~'\s' THEN  RAISE EXCEPTION 'Patient\_ID contains space';  END IF;  IF c."Firstname"!=LOWER(c."Firstname")THEN  RAISE EXCEPTION 'Firstname not be a case insensitive';  END IF;  IF c."Lastname"!=LOWER(c."Lastname")THEN  RAISE EXCEPTION 'Lastname not be a case insensitive';  END IF;  RETURN s;  RETURN c;  END;  $$ LANGUAGE plpgsql;    CREATE OR REPLACE TRIGGER verify\_space\_case  BEFORE INSERT ON "Patients" FOR EACH ROW EXECUTE PROCEDURE space\_case\_func();  **Output:** | |
| **32.Display the first 6 characters of the lab names.**  **Query:**  SELECT Substring("Lab\_names",1,6) AS "First 6 characters of lab names"  FROM "Lab\_Visit"  **Output:** | |
| **33.Write a query to create a table to get patients’ demographic details whose birth year is 1939. Name the table as “Patient\_Detail”**  **Query:**  CREATE TABLE "Patient\_Detail" (  Firstname varchar(255),  Lastname varchar(255),  Birthyear int,  Gender varchar(255),  Race varchar(255)    )  INSERT INTO "Patient\_Detail"  SELECT "Firstname","Lastname",EXTRACT(year from "Visit\_Date") - "Age" As "Birthyear","Gender","Race"  FROM "Patients"  INNER JOIN "Gender" using ("Gender\_ID")  INNER JOIN "Race" using ("Race\_ID")  WHERE EXTRACT(year from "Visit\_Date") - "Age"='1939'  SELECT \*  FROM"Patient\_Detail"  **Output:** | |
| **34.Write a query to get the number of patients above age 50 in each group**  **QUERY :**  SELECT count( "Age") AS "No. of patients Age greater than 50","Group"  FROM "Patients" p  INNER JOIN "Group" g  ON p."Group\_ID" = g."Group\_ID"  WHERE "Age">50  GROUP BY "Group"  **OUTPUT :** | |
| **35.Write a query to find the number of patients visited each month. (Display with month Name)**  **Query**  SELECT DISTINCT TO\_CHAR("Visit\_Date", 'Mon')AS "Month Name",count(\*) AS "Number of Patients visited"  FROM "Patients"  GROUP BY "Month Name" ;  **Output** | |
| **36.Write a query to get a number of visual/motor dementia patients who have any 2 abnormal conditions. (Display with condition name). (dementia/cognitive impairment: any patient who has any two abnormal test results).**  **QUERY :**  CREATE TEMP TABLE Temp\_Lab\_Visit("Lab\_visit\_ID" text ,"Lab\_Visit\_Date" timestamp,"Lab\_names" text);  INSERT INTO Temp\_Lab\_Visit select \* from "Lab\_Visit";  update Temp\_Lab\_Visit  set "Lab\_names" = 'Cultivate lab'  from Temp\_Lab\_Visit lv  inner join "Link\_Reference" lr on lr."Lab\_visit\_ID" = lv."Lab\_visit\_ID"  inner join "Lab\_Test" lt on lt."Lab\_ID" = lr."Lab\_ID"  where lt."Lab\_ID" = 'LB002'  **OUTPUT** :    **37.Write a query to get a list of patient IDs whose fasting glucose is 80, 85, and 89.**  **Query:**  SELECT  "Patient\_ID","Fasting\_Glucose"  FROM  "Lab\_Test"  WHERE  "Fasting\_Glucose" IN (80,85,89)  **Output:** | |
| **38. calculate the difference between Day and night HR. (Display 2 decimal only)**  **Query:**  SELECT  "Patient\_ID",  ROUND(CAST(("24Hr\_Day\_HR"-"24Hr\_Night\_HR") AS NUMERIC),2) AS "Diff\_In\_HR"  FROM  "Blood\_Pressure"  **Output :** | |
| **39.Find out the tables where column Patient\_ID is present. (Display column position number with respective table also)** | |
| **QUERY :**  SELECT  table\_name AS "Tables\_Having\_PatientID",  ordinal\_position AS "Position\_Of\_PatientID"  FROM  INFORMATION\_SCHEMA.COLUMNS  WHERE  column\_name = 'Patient\_ID'  **OUTPUT :** | |
| **40.Display the first name and Last name of patients whose Race is Latino.**  **Query :**  SELECT  "Firstname",  "Lastname",  "Race"  FROM  "Patients"  INNER JOIN "Race" on "Patients"."Race\_ID" = "Race"."Race\_ID"  WHERE  "Race" = 'Latino'  **Output:** | |
| **41. write a query to get the number of patients whose urine creatinine is in a normal range (Gender wise).**  Select "Gender", count("Gender")  from "Patients"  inner join "Gender" using("Gender\_ID")  inner join "Link\_Reference" using("Link\_Reference\_ID")  INNER JOIN "Urine\_Test" using ("Urine\_ID")  where "Gender" = 'Male' and "Creatinine" >= 65.4 and "Creatinine" <=119.3  or "Gender" = 'Female' and "Creatinine" >= 52.2 and "Creatinine" <=91.9  GROUP by "Gender" ; | |
| **42. Write a query to update id LB002 with the lab name Cultivate Lab.**  **Query:**  CREATE TEMP TABLE Temp\_Lab\_Visit("Lab\_visit\_ID" text ,"Lab\_Visit\_Date" timestamp,"Lab\_names" text);  INSERT INTO Temp\_Lab\_Visit select \* from "Lab\_Visit";  update Temp\_Lab\_Visit  set "Lab\_names" = 'Cultivate lab'  from Temp\_Lab\_Visit lv  inner join "Link\_Reference" lr on lr."Lab\_visit\_ID" = lv."Lab\_visit\_ID"  inner join "Lab\_Test" lt on lt."Lab\_ID" = lr."Lab\_ID"  where lt."Lab\_ID" = 'LB002'  **Output:** | |
| **43. Create an index on any table and use explain analyze to show differences if any.**  create index idx\_gender\_id on "Gender"("Gender\_ID");  explain select \* from public."Gender"    explain analyze select \* from public."Gender"    Using the explain command shows the generated query plan but does not run the query. Using the explain analyze command shows the query planning time and execution time. Explain analyze command can be run before creating an index and run again after creating an index. The query optimization is done using index. In the 2nd screen shot of the output image we can see that though the planning time is 0.074ms the execution time is 0.024ms. Since the planning time is more it’s always better to run the explain analyze before creating the index. The rise in planning time is negligible when compared to the reduction in execution time. | |
| **44. Write a query to split the lab visit date into two different columns lab\_visit\_date and lab\_visit\_time.**  **QUERY :**  select cast(to\_char("Lab\_Visit\_Date", 'YYYY-MM-DD') as date) as lab\_visit\_date,  cast(to\_char("Lab\_Visit\_Date", 'HH24:MI:SS') as time) as lab\_visit\_time from "Lab\_Visit";  **Output:** | |
| **45. Please go through the below screenshot and create the exact output.**      SELECT CAST(LTRIM("Patient\_ID",'S00') AS INT) as pat\_id,  cast(CASE WHEN CAST(LTRIM("Patient\_ID",'S00') AS integer) % 2 = 0 THEN 'true' else 'false' end as boolean)as even,  cast(CASE WHEN CAST(LTRIM("Patient\_ID",'S00') AS integer) % 2 = 1 THEN 'true' else 'false' end as boolean)as odd  FROM "Patients"; | |
|  | |
| **46. Calculate the Number of Diabetic Male and Female patients who are Anemic.**  **Query:**  SELECT "Gender", COUNT(\*) AS Anemic  FROM "Patients"  INNER JOIN "Gender" USING("Gender\_ID")  INNER JOIN "Lab\_Test" USING ("Patient\_ID")  where "Gender" = 'Male' AND "Hgb" < 13.2  OR "Gender" = 'Female' AND "Hgb" < 11.6 OR "Hb\_A1C" >=6.5  OR "Fasting\_Glucose">=120  GROUP by "Gender";  **Output:** | |
| **47. Write a query to display the Patient\_ID, last name, and the position of the substring 'an' in the last name column for those patients who have a substring 'an'.**  **Query:**  SELECT "Patient\_ID","Lastname", POSITION('an' in "Lastname")  FROM "Patients" WHERE SUBSTRING("Lastname",POSITION('an' in "Lastname"),2)='an';  **Output:** | |
| **48. List of patients from rows 30-40 without using the where condition.**  **Query:**  SELECT \* FROM "Patients" OFFSET 29 ROWS FETCH FIRST 10 ROWS ONLY;  **Output:** | |
| **49. Find all patients whose last name contains the text either Mc or Bu using Regular expression matches.**    **Query:**  SELECT "Firstname","Lastname"  FROM "Patients"  WHERE "Lastname" ~'Mc|Bu';  **Output:** | |
|  | |
| **50. Create materialized view with no data, to display no of male and female patients.**  **Query:**  CREATE MATERIALIZED VIEW mv\_Patients  AS SELECT g."Gender",COUNT(\*) FROM "Patients" p,"Gender" g  WHERE p."Gender\_ID"=g."Gender\_ID"  GROUP BY g."Gender";  SELECT \* FROM mv\_Patients;  **Output:** | |
| **51. Get a list of unique lab names whose names is starting with G and end with b**    SELECT DISTINCT LV."Lab\_names" FROM "Lab\_Visit" LV WHERE  LV."Lab\_names" LIKE ('G%b')    **OUTPUT** | |
| **52. Write the query to create an Index on table Verbal\_Cognitive by selecting a**  **column and also write the query drop the same index.**  **Query**    CREATE INDEX VC\_ID  ON "Verbal\_Cognitive"("VC\_ID");  DROP INDEX VC\_ID  **Output**    **53. Get the number of patients born in a leap year.**  select count("Patient\_ID") patients\_born\_leapyear from "Patients"  WHERE (EXTRACT(year from "Visit\_Date") - "Age") % 4 = 0; | |
| **54. Write a query to get a list of patient IDs from the DM group and above age 60**  **in sequence.**  **Query**  SELECT PA."Patient\_ID" FROM "Patients" PA,"Group" GR where  GR."Group"='DM' AND PA."Age">=60.  **Output** | |
| **55. Find the patient who has the most damage in the eyes with the use of a max function.**  **Query:**  SELECT "Patient\_ID","Firstname","Lastname"  FROM "Patients"  JOIN "Opthalmology" USING("Opthal\_ID")  WHERE"Diabetic\_Retinopathy"+"Macular\_Edema"=(  SELECT MAX("Diabetic\_Retinopathy" + "Macular\_Edema")  FROM "Opthalmology"  )  **Output** | |
| **56. Create a procedure for checking if Race exists using an if else statement.**  **Query**  CREATE OR REPLACE PROCEDURE check\_race\_exists(race\_name text)AS $$ DECLARE    BEGIN  SELECT "Race" INTO race\_name FROM "Race"  WHERE "Race" IN ('White','Latino','African\_american')  LIMIT 1;  IF  race\_name IS NOT NULL THEN  RAISE NOTICE 'Race exists';  ELSE  RAISE NOTICE 'Race not exists';  END IF;  END $$ LANGUAGE plpgsql;  call check\_race\_exists('White');  **Output** | |
| **57. Write a query to get a list of female patients who are at risk of heart diseases**  **with the help of Fasting HDL.**  **Query**  SELECT "Firstname","Lastname","Gender","Fasting\_HDL"  FROM "Patients"  INNER JOIN "Link\_Reference" USING ("Link\_Reference\_ID")  INNER JOIN "Lipid\_Lab\_Test" USING ("Lipid\_ID")  INNER JOIN "Gender" USING("Gender\_ID")  where "Gender" = 'Female' AND "Fasting\_HDL" < 50  **Output** | |
| **58. Create a role via query.**  **Query:**  CREATE ROLE ReadwriteRole1;  SELECT rolname FROM pg\_roles;  **Output:**      **readwriterole1 created successfully on last row**  **59. Query to list all the users in the DB.**  **Query**  SELECT usename as "DB UserName" FROM pg\_catalog.pg\_user;  **Output** | |
| **60. Write a query using the trigger after insert on the lab test table. If the patient has abnormal HbA1C and fasting glucose values.**  **Query:**  CREATE OR REPLACE FUNCTION patient\_diabetic\_risk() RETURNS TRIGGER AS  $$  BEGIN  if ab.Fasting\_Glucose>=100 OR ab.Hb\_A1C>=6.5  THEN  SELECT ab.Patient\_ID;  END IF;  END;  $$ LANGUAGE plpgsql;  CREATE OR REPLACE TRIGGER abnormal\_diabetic\_risk  AFTER INSERT OR UPDATE OR DELETE ON "Lab\_Test" FOR EACH STATEMENT EXECUTE PROCEDURE  patient\_diabetic\_risk();  **Output:** | |
| **61. write a query to get the number of patients for each age bin without using the CASE statement.(Bin size - 5)**  **Query**  SELECT COUNT("Patient\_ID") FILTER (WHERE ("Age" < 50)) AS "Under 50",  COUNT("Patient\_ID") FILTER (WHERE ("Age" >= 50 AND "Age" <=55)) AS "50-55",  COUNT("Patient\_ID") FILTER (WHERE ("Age" > 55 AND "Age" <= 60)) AS "55-60",  COUNT("Patient\_ID") FILTER (WHERE ("Age" >60 AND "Age" <= 65)) AS "60-65",  COUNT("Patient\_ID") FILTER (WHERE ("Age" >65 AND "Age" <= 70)) AS "65-70"  FROM "Patients";  **Output** | |
| **62. Write a query to get the number of patients who have normal platelets for each group.**  **Query**  SELECT "Group",count( "Platelets") AS "Platelets"  FROM "Lab\_Test"  INNER JOIN "Link\_Reference" using("Lab\_ID")  INNER JOIN "Patients" using("Link\_Reference\_ID")  INNER JOIN "Group" using("Group\_ID")  WHERE "Platelets">=150 and "Platelets"<=450  GROUP BY "Group";  **Output** | |
| **63. Create a trigger on a view of the Blood Pressure table.**  **Query :**  CREATE VIEW "vw\_BloodPressure" AS  SELECT \* FROM "Blood\_Pressure";  CREATE OR REPLACE FUNCTION trigger\_func() RETURNS TRIGGER AS $$  BEGIN  IF TG\_OP = 'INSERT' then  raise notice 'INSERT trigger, NEW = [%]', NEW;  ELSIF TG\_OP = 'UPDATE' then  raise notice 'UPDATE trigger, OLD = [%], NEW = [%]', OLD, NEW;  ELSE  raise notice 'DELETE trigger, OLD = [%]', OLD;  END IF;  RETURN NULL;  END;  $$ LANGUAGE plpgsql;  CREATE TRIGGER vw\_trigger  INSTEAD OF INSERT OR UPDATE OR DELETE ON "vw\_BloodPressure"  FOR EACH ROW EXECUTE PROCEDURE trigger\_func();  INSERT INTO "vw\_BloodPressure"("BP\_ID","Patient\_ID","24Hr\_Day\_SBP","24Hr\_Day\_DBP","24Hr\_Day\_HR","24Hr\_Night\_SBP","24Hr\_Night\_DBP","24Hr\_Night\_HR")  VALUES ('BP079','S1111',133,74.9,75.12,111,64.42,66.38)  **Output :** | |
| **64.Write a query to find the number of Patients whose Gait RPE start is greater than the end and vice versa. (Display exact output as shown below)**    **Query :**  SELECT  "Type",  CASE WHEN "Type" = 'RPE\_End>Start' THEN CAST(SUM("Count") AS BIGINT)  WHEN "Type" = 'RPE\_Start>End' THEN CAST(SUM("Count") AS BIGINT) END "Count"  FROM  (  SELECT  CASE WHEN "Gait\_RPE\_End " > "Gait\_RPE\_Start " THEN 'RPE\_End>Start'  WHEN "Gait\_RPE\_Start " > "Gait\_RPE\_End " THEN 'RPE\_Start>End' END AS "Type",  CASE WHEN "Gait\_RPE\_End " > "Gait\_RPE\_Start " THEN COUNT("Patient\_ID")  WHEN "Gait\_RPE\_Start " > "Gait\_RPE\_End " THEN COUNT("Patient\_ID") END AS "Count"  FROM "Walking\_Test"  GROUP BY "Gait\_RPE\_Start ","Gait\_RPE\_End "  ) AS FINAL  WHERE "Type" IS NOT NULL  GROUP BY "Type"  ORDER BY "Type"  **Output :** | |
| **65. Create a view without using any schema or table and check the created view using a select statement.**  **Query :**  CREATE VIEW vw\_series AS  (  SELECT \* FROM generate\_series(0,20,2)  );  SELECT \* FROM vw\_series;    **Output :** | |
| **66. Display patients names who have the same last name.**  **Query :**  SELECT "Firstname","Lastname"  FROM ( SELECT \*, count(1) over (partition BY "Lastname") as "Same\_LastName"  FROM "Patients") AS pt  WHERE "Same\_LastName" > 1 ;  **Output :** | |
| **67. Write a query to get the Sum of Diabetes Duration for Group id 'GRP\_02'.**  **Query :**  SELECT  SUM("Diabetes\_Duration") AS "DiabetesDuration"  FROM  "Patients"  WHERE  "Group\_ID" = 'GRP\_02'  **Output** : | |
| **68. Write a query to get a patient name who has a chance to have kidney disease with Albumin.**  **Query :**  SELECT  "Firstname",  "Lastname"  FROM  "Patients" AS pt  INNER JOIN "Link\_Reference" AS lr ON pt."Link\_Reference\_ID" = lr."Link\_Reference\_ID"  INNER JOIN "Urine\_Test" AS ut ON lr."Urine\_ID" = ut."Urine\_ID"  WHERE  "Albumin" >= 30  **Output** : | |
| **69. Get the patient's name who has a max speed.**  select "Firstname", "Lastname" ,"Gait\_DT\_Speed" from "Walking\_Test"  inner join "Patients" using("Patient\_ID")  order by "Gait\_DT\_Speed" desc  limit 1; | |
| **70. Write a query to find out the percentage of Lab visits by Lab names.**  WITH tlv AS (SELECT "Lab\_names", COUNT(DISTINCT "Lab\_visit\_ID") AS pat\_visited  FROM "Lab\_Visit"  GROUP BY "Lab\_names")  SELECT  "Lab\_names",round(pat\_visited \* 100.0 /(SELECT COUNT(\*) FROM "Lab\_Visit"),2) AS percentage  FROM tlv; | |
| **71. Write a query to get Patient IDs for verbally cognitively impaired who satisfy any 2 conditions. (HINT: dementia/cognitive impaired: any patient who has any two abnormal test results).**  select "Patient\_ID" from public."Verbal\_Cognitive"  where  "DS"<13 and "HVLT" <14  or  "DS"<13 and "VF"<42  or  "DS"<13 and "WTAR"<=20  or  "HVLT"<14 and "VF"<42  or  "HVLT"<14 and "WTAR"<=20  or  "VF"<42 and "WTAR"<=20; | |
| **72. Display a list of patients who are memory cognitively impaired with the GDS test and whose diabetes duration is between 5 to 30.**  select "Firstname","Lastname", "GDS", "Diabetes\_Duration" from public."Memory\_Cognitive"  inner join public."Link\_Reference" using("MC\_ID")  inner join public."Patients" using ("Link\_Reference\_ID")  where "GDS" >= 15 and "Diabetes\_Duration" between 5 and 30; | |
| **73. Write a query to the get number of Patient\_IDs who visited between March 2005 and March 2006**    **Query**  select count(\*) as "Patient\_Count\_between\_March2005\_06" from "Patients"  where "Visit\_Date" BETWEEN '2005-03-01 00:00:00'  AND '2006-03-31 23:59:59';  **Output:** | |
| **74. Get the number of patients who visited each lab using the windows function.**  SELECT "Lab\_names", COUNT(DISTINCT "Lab\_visit\_ID") AS pat\_visited  FROM "Lab\_Visit"  GROUP BY "Lab\_names"; | |
| **75. Find the number of control and DM patients who visited each lab.**  **Query:**  SELECT COUNT(\*)AS Patients,l."Lab\_names",g."Group" from "Patients" p  JOIN "Group"g ON p."Group\_ID"=g."Group\_ID"  JOIN "Link\_Reference"lr ON p."Link\_Reference\_ID"=lr."Link\_Reference\_ID"  JOIN "Lab\_Visit"l ON lr."Lab\_visit\_ID"=l."Lab\_visit\_ID"  GROUP BY l."Lab\_names",g."Group"  ORDER BY g."Group";  **Output:** | |
| **76. Please go through the below screenshot and create the exact output.**      **Query:**  SELECT CONCAT("Firstname",' ',"Lastname")AS FULLNAME,  LENGTH(CONCAT("Firstname",' ',"Lastname")) FROM "Patients";  **Output:** | |
|  | |
|  | |
| **77. Write a query to get comma-separated values of patient details .(Use a maximum of 6 columns from different tables)**  **Query:**  SELECT p."Patient\_ID",  STRING\_AGG(CONCAT(p."Firstname",',',p."Lastname",',' ,gd."Gender",',',g."Group",',',  r."Race",',',lt."Lab\_ID"),''ORDER BY p."Firstname" ASC)AS Patient\_Details  FROM public."Patients" p  INNER JOIN "Group" g ON g."Group\_ID"=p."Group\_ID"  INNER JOIN "Gender" gd ON gd."Gender\_ID"=p."Gender\_ID"  INNER JOIN "Race"r ON r."Race\_ID"=p."Race\_ID"  INNER JOIN "Lab\_Test" lt ON lt."Patient\_ID"=p."Patient\_ID"  GROUP BY p."Patient\_ID";  **Output:** | |
| **78. Write a query to determine get the Patient IDs ,in DM and Control groups ,that are in prediabetic stage and label them accordingly.**  **Query:**  SELECT p."Patient\_ID", g."Group",l."Hb\_A1C",l."Fasting\_Glucose" from "Group" g  JOIN "Patients"p ON p."Group\_ID"=g."Group\_ID"  JOIN "Lab\_Test"l ON l."Patient\_ID"=p."Patient\_ID"  WHERE l."Hb\_A1C">=5.7 AND l."Hb\_A1C"<=6.4  OR l."Fasting\_Glucose"=100  GROUP BY g."Group",p."Patient\_ID",l."Hb\_A1C",l."Fasting\_Glucose"  ORDER BY "Patient\_ID";  **Output:** | |
| **79.Calculate the Patient's Daytime MAP and Nighttime MAP.**  **Query:**  SELECT bp."Patient\_ID",ROUND(CAST((((bp."24Hr\_Day\_SBP")-(bp."24Hr\_Day\_DBP"))/3)+  (bp."24Hr\_Day\_DBP")AS NUMERIC),2) AS Daytime\_MAP,  ROUND(CAST(((("24Hr\_Night\_SBP")-("24Hr\_Night\_DBP"))/3)+  ("24Hr\_Night\_DBP")AS NUMERIC),2) AS Nighttime\_MAP FROM "Blood\_Pressure" bp;  **Output:**    **80.Write a query using recursive view.**  **Query:**  CREATE RECURSIVE VIEW twenties(n) AS  (  SELECT 1 AS NUM  UNION ALL  SELECT n+1 FROM twenties  );  SELECT \* FROM twenties LIMIT 20;  **Output:** | |

****