**Section A: Libraries, Imports & Basic Manipulations**

/\* Create a library named hospital \*/

libname hospital 'C:\\sasdata\\hospital';

/\* Import a CSV version of patients\_2024 into the hospital library \*/

proc import datafile='C:\\sasdata\\patients\_2024.csv'

out=hospital.patients\_2024

dbms=csv

replace;

getnames=yes;

run;

/\* Use PROC CONTENTS to explore the structure of treatments\_2024 \*/

proc contents data=treatments\_2024;

run;

**Section B: Advanced PROC SQL**

/\* Join patients\_2024 and treatments\_2024 to get name, diagnosis, treatment name, and cost \*/

proc sql;

select a.Name, a.Diagnosis, b.TreatmentName, b.Cost

from patients\_2024 as a

inner join treatments\_2024 as b

on a.PatientID = b.PatientID;

quit;

/\* Add physicians to this join and display patient name, doctor name, and treatment \*/

proc sql;

select a.Name as PatientName, c.Name as DoctorName, b.TreatmentName

from patients\_2024 as a

inner join treatments\_2024 as b

on a.PatientID = b.PatientID

inner join physicians as c

on b.PhysicianID = c.PhysicianID;

quit;

/\* Find the total treatment cost per patient \*/

proc sql;

select PatientID, sum(Cost) as TotalTreatmentCost

from treatments\_2024

group by PatientID;

quit;

/\* Identify patients who received more than one treatment \*/

proc sql;

select PatientID, count(\*) as NumberOfTreatments

from treatments\_2024

group by PatientID

having count(\*) > 1;

quit;

/\* Use a subquery to find patients whose total treatment cost is above average \*/

proc sql;

select PatientID, sum(Cost) as TotalCost

from treatments\_2024

group by PatientID

having calculated TotalCost > (select avg(sum(Cost)) from treatments\_2024 group by PatientID);

quit;

/\* Create a new SQL table with only ICU patients \*/

proc sql;

create table ICU\_Patients as

select \*

from patients\_2024

where Ward = 'ICU';

quit;

/\* Use aliasing to rename columns for readability \*/

proc sql;

select a.Name as Patient,

a.Diagnosis as Illness,

b.TreatmentName as Medication,

b.Cost as Expense

from patients\_2024 as a

inner join treatments\_2024 as b

on a.PatientID = b.PatientID;

quit;

**Section C: PROC REPORT & TABULATE**

/\* Create a PROC REPORT showing patient name, diagnosis, cost, and physician specialty \*/

proc sql;

create table report\_data as

select a.Name as PatientName,

a.Diagnosis,

b.Cost,

c.Specialty

from patients\_2024 as a

inner join treatments\_2024 as b on a.PatientID = b.PatientID

inner join physicians as c on b.PhysicianID = c.PhysicianID;

quit;

proc report data=report\_data nowd;

column PatientName Diagnosis Cost Specialty;

run;

/\* Use a DEFINE block to add custom column headers \*/

proc report data=report\_data nowd;

column PatientName Diagnosis Cost Specialty;

define PatientName / 'Patient Name';

define Diagnosis / 'Medical Condition';

define Cost / 'Treatment Cost';

define Specialty / 'Doctor Specialty';

run;

/\* Highlight patients who spent more than ₹2000 using COMPUTE block \*/

proc report data=report\_data nowd;

column PatientName Diagnosis Cost Specialty;

define Cost / display;

compute Cost;

if Cost > 2000 then

call define(\_col\_, "style", "style={background=lightred font\_weight=bold}");

endcomp;

run;

/\* Generate a PROC REPORT grouped by diagnosis, showing count of patients and average cost \*/

proc sql;

create table diag\_summary as

select Diagnosis, count(distinct a.PatientID) as NumPatients, avg(b.Cost) as AvgCost

from patients\_2024 as a

inner join treatments\_2024 as b on a.PatientID = b.PatientID

group by Diagnosis;

quit;

proc report data=diag\_summary nowd;

column Diagnosis NumPatients AvgCost;

define Diagnosis / group 'Diagnosis';

define NumPatients / 'No. of Patients';

define AvgCost / 'Average Cost';

run;

/\* Use PROC TABULATE to cross-tab diagnosis and gender \*/

proc tabulate data=patients\_2024;

class Diagnosis Gender;

table Diagnosis, Gender\*N;

run;

/\* Export the PROC REPORT output as an Excel file \*/

ods excel file="C:\\sasdata\\hospital\\patient\_report.xlsx";

proc report data=report\_data nowd;

column PatientName Diagnosis Cost Specialty;

run;

ods excel close;

**Section D: BY-group & FIRST./LAST.**

/\* Sort treatments by PatientID and use BY group processing \*/

proc sort data=treatments\_2024 out=sorted\_treatments;

by PatientID;

run;

/\* Use FIRST. and LAST. to flag patients with multiple treatments \*/

data flagged\_treatments;

set sorted\_treatments;

by PatientID;

if first.PatientID and last.PatientID then Multi\_Treatment = 'No';

else Multi\_Treatment = 'Yes';

run;

proc print data=flagged\_treatments;

title "Patients flagged for multiple treatments";

run;

/\* Count the number of distinct patients receiving each treatment using BY-groups \*/

proc sort data=treatments\_2024 out=sorted\_by\_treatment;

by TreatmentName PatientID;

run;

data unique\_treatments;

set sorted\_by\_treatment;

by TreatmentName PatientID;

if first.PatientID then output;

run;

proc sql;

select TreatmentName, count(\*) as UniquePatients

from unique\_treatments

group by TreatmentName;

quit;

/\* Display the first treatment each patient received \*/

data first\_treatment;

set sorted\_treatments;

by PatientID;

if first.PatientID;

run;

proc print data=first\_treatment;

title "First treatment per patient";

run;

**Section E: Conditional Logic**

/\* Create a new variable CostLevel: Low (<1000), Medium (1000–2000), High (>2000) \*/

data treatments\_costlevel;

set treatments\_2024;

length CostLevel $10;

if Cost < 1000 then CostLevel = 'Low';

else if 1000 <= Cost <= 2000 then CostLevel = 'Medium';

else if Cost > 2000 then CostLevel = 'High';

run;

proc print data=treatments\_costlevel;

title "Treatments with Cost Level Classification";

run;

/\* Use SELECT-WHEN to assign room types based on diagnosis \*/

data patients\_roomtype;

set patients\_2024;

length RoomType $15;

select (Diagnosis);

when ('Flu') RoomType = 'Isolation';

when ('Covid') RoomType = 'Critical Care';

when ('Fracture') RoomType = 'Recovery Ward';

when ('Diabetes', 'Hypertension') RoomType = 'Observation';

otherwise RoomType = 'General';

end;

run;

proc print data=patients\_roomtype;

title "Patients with Room Type Assigned Based on Diagnosis";

run;

/\* Add a flag HighRisk if age > 50 or diagnosis = "Covid" \*/

data patients\_risk;

set patients\_2024;

length HighRisk $3;

if Age > 50 or Diagnosis = "Covid" then HighRisk = 'Yes';

else HighRisk = 'No';

run;

proc print data=patients\_risk;

title "High Risk Patients Flagged";

run;

/\* Classify patients into age groups using multiple IF-THEN/ELSE \*/

data patients\_agegroup;

set patients\_2024;

length AgeGroup $12;

if Age < 30 then AgeGroup = 'Young';

else if 30 <= Age < 50 then AgeGroup = 'Adult';

else AgeGroup = 'Senior';

run;

proc freq data=patients\_agegroup;

tables AgeGroup;

run;

**Section F: Advanced Merging**

/\* Perform a many-to-many merge between patients\_2024 and treatments\_2024 \*/

proc sql;

create table patient\_treatments as

select a.PatientID, a.Name, a.Diagnosis, a.Ward, b.TreatmentCode,

b.TreatmentName, b.Cost, b.PhysicianID

from patients\_2024 as a

inner join treatments\_2024 as b

on a.PatientID = b.PatientID;

quit;

proc print data=patient\_treatments;

title "Many-to-Many Merged Patient & Treatment Data";

run;

/\* Merge the result with physicians to get physician details \*/

proc sql;

create table full\_merge as

select pt.\*, p.Name as PhysicianName, p.Specialty

from patient\_treatments as pt

left join physicians as p

on pt.PhysicianID = p.PhysicianID;

quit;

proc print data=full\_merge;

title "Merged Data with Physician Details";

run;

/\* Create a table showing total earnings per physician from all treatments \*/

proc sql;

create table physician\_earnings as

select PhysicianID, PhysicianName, Specialty,

sum(Cost) as TotalEarnings

from full\_merge

group by PhysicianID, PhysicianName, Specialty;

quit;

proc print data=physician\_earnings;

title "Total Earnings by Physician";

run;

**Section G: Conditional Appending**

/\* Create another dataset: patients\_2023 with older records \*/

data patients\_2023;

input PatientID $ Name $ Gender $ Age VisitDate : yymmdd10. Diagnosis $ Ward $;

format VisitDate yymmdd10.;

datalines;

P090 Rina F 55 2023-11-20 Diabetes General

P091 Arjun M 62 2023-10-15 Hypertension Cardio

P092 Tina F 45 2023-12-05 Flu General

;

run;

/\* Create a temporary dataset with patients aged > 50 from 2023 \*/

data patients\_2023\_over50;

set patients\_2023;

if Age > 50;

run;

/\* Append only patients >50 from 2023 to patients\_2024 \*/

proc append base=patients\_2024 data=patients\_2023\_over50 force;

run;

/\* Verify the appended dataset \*/

proc print data=patients\_2024;

title "Patients 2024 After Appending Selected Records from 2023";

run;

**Section H: Bonus Challenges**

/\* Create a macro to report treatment cost per patient \*/

%macro report\_cost;

proc sql;

select PatientID, sum(Cost) as TotalCost

from treatments\_2024

group by PatientID;

quit;

%mend;

%report\_cost

/\* Generate a horizontal bar chart of total cost by diagnosis \*/

proc sql;

create table cost\_by\_diag as

select a.Diagnosis, sum(b.Cost) as TotalCost

from patients\_2024 as a

inner join treatments\_2024 as b

on a.PatientID = b.PatientID

group by a.Diagnosis;

quit;

proc sgplot data=cost\_by\_diag;

hbar Diagnosis / response=TotalCost datalabel;

title "Total Treatment Cost by Diagnosis";

run;

/\* Create a custom $roomfmt format for Ward values \*/

proc format;

value $roomfmt

'General' = 'GEN - General Ward'

'ICU' = 'INT - Intensive Care Unit'

'Cardio' = 'CRD - Cardiology'

'Ortho' = 'ORT - Orthopedic Unit';

run;

proc print data=patients\_2024;

format Ward $roomfmt.;

title "Patients with Custom Ward Formatting";

run;

/\* Use PROC TRANSPOSE to pivot treatment data per patient \*/

proc transpose data=treatments\_2024 out=transposed\_treatments(drop=\_name\_);

by PatientID;

id TreatmentCode;

var Cost;

run;

proc print data=transposed\_treatments;

title "Treatment Costs Pivoted by Treatment Code";

run;

/\* Add a cumulative cost column using retain and sum in a DATA step \*/

proc sort data=treatments\_2024 out=sorted\_cost;

by PatientID;

run;

data treatments\_cumulative;

set sorted\_cost;

by PatientID;

retain CumulativeCost;

if first.PatientID then CumulativeCost = 0;

CumulativeCost + Cost;

run;

proc print data=treatments\_cumulative;

title "Cumulative Cost by Patient";

run;