-- STEP 1: VERIFY DATABASE AND TABLE -- Ensure the correct database is selected

USE hospital;

SET SQL\_SAFE\_UPDATES = 0;

-- STEP 2: HANDLE DATA CLEANSING -- 2.1 Handle missing ages by filling with the average age

UPDATE healthcare SET Age = (SELECT AVG(Age) FROM (SELECT \* FROM healthcare) AS temp)

WHERE Age IS NULL;

-- 2.2 Replace Null Values in Gender with 'Unknown'

UPDATE healthcare

SET Gender = 'Unknown'

WHERE Gender IS NULL OR Gender = '';

-- 2.3 Correct invalid age entries (negative or unreasonably high ages)

UPDATE healthcare

SET Age = NULL

WHERE Age < 0;

-- 2.4 Standardize blood type entries

UPDATE healthcare

SET `Blood Type` = CASE

WHEN `Blood Type` IN ('a+', 'A +', 'A pos', 'a positive') THEN 'A+'

WHEN `Blood Type` IN ('b+', 'B pos', 'B positive') THEN 'B+'

WHEN `Blood Type` IN ('o+', 'O pos', 'O positive') THEN 'O+' WHEN `Blood Type` IN ('ab+', 'AB pos', 'AB positive') THEN 'AB+' ELSE 'Unknown'

END;

-- STEP 3: IDENTIFY RECORDS WITH MISSING ESSENTIAL FIELDS -- 3.1 Filter out records with missing critical fields

SELECT \*

FROM healthcare

WHERE Name IS NULL

OR Age IS NULL

OR Gender IS NULL

OR `Blood Type` IS NULL

OR `Medical Condition` IS NULL

OR `Date of Admission` IS NULL

OR Doctor IS NULL

OR Hospital IS NULL

OR `Insurance Provider` IS NULL

OR `Billing Amount` IS NULL

OR `Room Number` IS NULL

OR `Admission Type` IS NULL

OR `Discharge Date` IS NULL

OR Medication IS NULL

OR `Test Results` IS NULL;

-- STEP 4: DATA ENRICHMENT

-- 4.1 Extract Month and Year from Admission Dates for Further Analysis

ALTER TABLE healthcare ADD COLUMN Admission\_Month VARCHAR(20), ADD COLUMN Admission\_Year INT;

-- 4.2 Categorize patients into age groups

UPDATE healthcare

SET Age\_Group = CASE

WHEN Age < 18 THEN 'Child'

WHEN Age BETWEEN 18 AND 35 THEN 'Young Adult'

WHEN Age BETWEEN 36 AND 60 THEN 'Adult'

ELSE 'Senior'

END;

-- 4.3 Standardize date formats for admission dates

UPDATE healthcare SET Date\_of\_Admission = STR\_TO\_DATE(Date\_of\_Admission, '%d-%m-%Y')

WHERE Date\_of\_Admission IS NOT NULL;

-- 4.4 Trim whitespace from all text fields

UPDATE healthcare

SET Name = TRIM(Name),

Gender = TRIM(Gender),

`Blood Type` = TRIM(`Blood Type`), `Medical Condition` = TRIM(`Medical Condition`),

Doctor = TRIM(Doctor),

Hospital = TRIM(Hospital), `Insurance Provider` = TRIM(`Insurance Provider`),

Medication = TRIM(Medication),

`Test Results` = TRIM(`Test Results`);

-- 4.5 Ensure consistent capitalization in text fields

UPDATE healthcare

SET Gender = UPPER(Gender),

`Blood Type` = UPPER(`Blood Type`),

`Test Results` = UPPER(`Test Results`);

-- STEP 5: BASIC DATA ANALYSIS

-- 5.1 View all data

SELECT \* FROM healthcare;

-- 5.2 Count the number of records

SELECT COUNT(\*) AS Total\_Patients FROM healthcare;

-- 5.3 Display distinct medical conditions SELECT DISTINCT `Medical Condition` FROM healthcare;

-- 5.4 Count patients by gender SELECT Gender, COUNT(\*) AS Count FROM healthcare GROUP BY Gender;

-- 5.5 Find the average age of patients

SELECT AVG(Age) AS Average\_Age FROM healthcare;

-- STEP 6: INTERMEDIATE ANALYSIS

-- 6.1 List patients admitted after 2020

SELECT \* FROM healthcare WHERE Date\_of\_Admission > '2020-01-01';

-- 6.2 Find the total billing amount grouped by hospital SELECT Hospital, SUM(`Billing Amount`) AS Total\_Billing

FROM healthcare

GROUP BY Hospital;

-- 6.3 Count the number of patients by admission type SELECT `Admission Type`, COUNT(\*) AS Count

FROM healthcare

GROUP BY `Admission Type`;

-- 6.4 Find the doctor with the most patients SELECT Doctor, COUNT(\*) AS Patient\_Count

FROM healthcare

GROUP BY Doctor ORDER BY Patient\_Count DESC

LIMIT 1;

-- 6.5 Find the room number with the highest total billing SELECT `Room Number`, SUM(`Billing Amount`) AS Total\_Billing

FROM healthcare

GROUP BY `Room Number` ORDER BY Total\_Billing DESC

LIMIT 1;

-- STEP 7: ADVANCED ANALYSIS -- 7.1 List patients with abnormal test results

SELECT \*

FROM healthcare

WHERE `Test Results` = 'Abnormal';

-- 7.2 Calculate the average billing amount for different medical conditions SELECT `Medical Condition`, AVG(`Billing Amount`) AS Average\_Billing

FROM healthcare

GROUP BY `Medical Condition`;

-- 7.3 Count the number of patients per blood type

SELECT `Blood Type`, COUNT(\*) AS Count

FROM healthcare

GROUP BY `Blood Type`;

-- 7.4 Identify the top 3 medications prescribed SELECT Medication, COUNT(\*) AS Prescription\_Count

FROM healthcare

GROUP BY Medication ORDER BY Prescription\_Count DESC

LIMIT 3;

-- STEP 8: COMPLEX ANALYSIS

-- 8.1 Find the hospital with the highest average billing amount SELECT Hospital, AVG(`Billing Amount`) AS Average\_Billing

FROM healthcare

GROUP BY Hospital ORDER BY Average\_Billing DESC

LIMIT 1;

-- 8.2 Calculate total billing for patients covered by each insurance provider SELECT `Insurance Provider`, SUM(`Billing Amount`) AS Total\_Billing

FROM healthcare

GROUP BY `Insurance Provider`;

-- 8.3 List doctors treating more than 5 patients SELECT Doctor, COUNT(\*) AS Patient\_Count

FROM healthcare

GROUP BY Doctor

HAVING Patient\_Count > 5;

-- 8.4 Identify the most common medical condition treated SELECT `Medical Condition`, COUNT(\*) AS Frequency

FROM healthcare

GROUP BY `Medical Condition` ORDER BY Frequency DESC

LIMIT 1;

-- 8.5 List all patients who were billed more than the average billing amount

SELECT \*

FROM healthcare

WHERE `Billing Amount` > (SELECT AVG(`Billing Amount`) FROM healthcare);

-- 8.6 Find the top 5 hospitals by the number of patients treated

SELECT Hospital, COUNT(\*) AS Patient\_Count

FROM healthcare

GROUP BY Hospital ORDER BY Patient\_Count DESC

LIMIT 5;

-- STEP 9: PERCENTAGE AND UNIQUE ANALYSIS

-- 9.1 Calculate the percentage of patients per gender SELECT Gender, (COUNT(\*) \* 100.0 / (SELECT COUNT(\*) FROM healthcare)) AS Percentage

FROM healthcare

GROUP BY Gender;

-- 9.2 List all unique combinations of medication and test results SELECT DISTINCT Medication, `Test Results`

FROM healthcare;

-- STEP 10: ADVANCED PATTERN ANALYSIS -- 10.1 Find patients who have the same medical condition and medication

SELECT \*

FROM healthcare h1

WHERE EXISTS (

SELECT 1

FROM healthcare h2 WHERE h1.`Medical Condition` = h2.`Medical Condition` AND h1.Medication = h2.Medication AND h1.ID <> h2.ID

);

-- 10.2 Determine the earliest and latest admission dates

SELECT MIN(`Date of Admission`) AS Earliest\_Admission, MAX(`Date of Admission`) AS

Latest\_Admission

FROM healthcare;

-- 10.3 Find all patients who were readmitted (assume patients with the same name) SELECT Name, COUNT(\*) AS Readmissions

FROM healthcare

GROUP BY Name

HAVING Readmissions > 1;

-- 10.4 Identify doctors who handled patients with abnormal test results

SELECT DISTINCT Doctor

FROM healthcare

WHERE `Test Results` = 'Abnormal';

-- STEP 11: ADDITIONAL INSIGHTS

-- 11.1 Calculate the average stay length by admission type

SELECT `Admission Type`, AVG(DATEDIFF(`Discharge Date`, `Date of Admission`)) AS

Avg\_Stay\_Length

FROM healthcare

GROUP BY `Admission Type`;

-- 11.2 Find patients whose billing amount is within the top 10%

SELECT \*

FROM healthcare

WHERE `Billing Amount` >= ( SELECT MIN(`Billing Amount`)

FROM (

SELECT `Billing Amount`, PERCENT\_RANK() OVER (ORDER BY `Billing Amount`) AS Percentile

FROM healthcare

) AS Ranked WHERE Percentile >= 0.9

LIMIT 3

);

-- 11.3 Determine the correlation between age and billing amount (basic observation) SELECT Age, `Billing Amount`

FROM healthcare

ORDER BY Age;

-- 11.4 Find the top 3 conditions with the highest average billing amount SELECT `Medical Condition`, AVG(`Billing Amount`) AS Avg\_Billing

FROM healthcare

GROUP BY `Medical Condition` ORDER BY Avg\_Billing DESC

LIMIT 3;

-- 11.5 Identify the total number of patients for each blood type and their average age SELECT `Blood Type`, COUNT(\*) AS Total\_Patients, AVG(Age) AS Avg\_Age

FROM healthcare

GROUP BY `Blood Type`;