```sql
/* ==========================================================
   HEALTHCARE DATA ANALYTICS PROJECT – COMPLETE SQL SCRIPT
   Database  : hospital_db
   Table     : patient_records_new
   Note      : Data Imported via SSMS Import Wizard
   ========================================================== */

------------------------------------------------------------
-- 1) CREATE DATABASE
------------------------------------------------------------
CREATE DATABASE hospital_db;
GO

USE hospital_db;
GO

------------------------------------------------------------
-- 2) CREATE TABLE (Matches CSV Structure)
------------------------------------------------------------
CREATE TABLE patient_records_new (
    name VARCHAR(100),
    age INT,
    gender VARCHAR(10),
    blood_type VARCHAR(5),
    medical_condition VARCHAR(100),
    date_of_admission DATE,
    doctor VARCHAR(100),
    hospital VARCHAR(150),
    insurance_provider VARCHAR(50),
    billing_amount VARCHAR(20),   -- contains $ and commas
    room_number INT,
    admission_type VARCHAR(20),
    discharge_date DATE,
    medication VARCHAR(50),
    test_results VARCHAR(20),
    length_of_stay INT,
    age_group VARCHAR(20),
    high_cost_flag VARCHAR(10),
    chronic_flag VARCHAR(10),
    emergency_flag INT
);
GO

------------------------------------------------------------
-- 3) DATA IMPORT STEP (DONE USING SSMS IMPORT WIZARD)
--     No BULK INSERT USED AS PER PROJECT SETUP
------------------------------------------------------------


------------------------------------------------------------
-- 4) VERIFY DATA LOAD
------------------------------------------------------------
SELECT TOP 10 * FROM patient_records_new;
SELECT COUNT(*) AS total_records FROM patient_records_new;
GO


------------------------------------------------------------
-- 5) CLEAN BILLING AMOUNT (TEXT → NUMERIC)
------------------------------------------------------------
ALTER TABLE patient_records_new
ADD billing_amount_num FLOAT;
```

```
GO

UPDATE patient_records_new
SET billing_amount_num =
  CAST(REPLACE(REPLACE(billing_amount, '$', ''), ',', '') AS FLOAT);
GO

SELECT billing_amount, billing_amount_num
FROM patient_records_new;
GO

-------------------------------------------------------------
-- 6) DATA VALIDATION CHECKS
-------------------------------------------------------------

-- Age validation
SELECT * FROM patient_records_new
WHERE age < 0 OR age > 120;

-- Gender validation
SELECT DISTINCT gender FROM patient_records_new;

-- Discharge before admission (critical error)
SELECT * FROM patient_records_new
WHERE discharge_date < date_of_admission;

-- Invalid billing
SELECT * FROM patient_records_new
WHERE billing_amount_num <= 0;

-- Emergency flag mismatch
```

```sql
SELECT admission_type, emergency_flag
FROM patient_records_new
WHERE admission_type = 'Emergency'
  AND emergency_flag <> 1;

-- Test result distribution
SELECT test_results, COUNT(*)
FROM patient_records_new
GROUP BY test_results;
GO

------------------------------------------------------------
-- 7) DATA CLEANING
------------------------------------------------------------

-- Standardize gender
UPDATE patient_records_new
SET gender = UPPER(gender);
GO

-- Remove duplicate patients
WITH cte AS (
    SELECT *,
           ROW_NUMBER() OVER (
               PARTITION BY name, date_of_admission, hospital
               ORDER BY name
           ) AS rn
    FROM patient_records_new
)
DELETE FROM cte WHERE rn > 1;
GO
```

```
-- Fix negative billing
UPDATE patient_records_new
SET billing_amount_num = ABS(billing_amount_num)
WHERE billing_amount_num < 0;
GO

----------------------------------------------------------
-- 8) DATA GOVERNANCE CONSTRAINTS
----------------------------------------------------------
ALTER TABLE patient_records_new
ADD CONSTRAINT chk_age_new
CHECK (age BETWEEN 0 AND 120);

ALTER TABLE patient_records_new
ADD CONSTRAINT chk_gender_new
CHECK (gender IN ('Male','Female'));

ALTER TABLE patient_records_new
ADD CONSTRAINT chk_admission_type_new
CHECK (admission_type IN ('Emergency','Elective','Urgent'));

ALTER TABLE patient_records_new
ADD CONSTRAINT chk_test_results_new
CHECK (test_results IN ('Normal','Abnormal','Inconclusive'));

ALTER TABLE patient_records_new
ADD CONSTRAINT chk_emergency_flag_new
CHECK (emergency_flag IN (0,1));
GO
```

```
------------------------------------------------------------
-- 9) CORE HEALTHCARE KPI QUERIES
------------------------------------------------------------

-- Total Patients
SELECT COUNT(*) AS total_patients
FROM patient_records_new;

-- Total Revenue
SELECT SUM(billing_amount_num) AS total_revenue
FROM patient_records_new;

-- Average Billing
SELECT AVG(billing_amount_num) AS avg_billing
FROM patient_records_new;

-- Revenue by Hospital
SELECT hospital, SUM(billing_amount_num) AS hospital_revenue
FROM patient_records_new
GROUP BY hospital;

-- Revenue by Insurance Provider
SELECT insurance_provider, SUM(billing_amount_num) AS insurance_revenue
FROM patient_records_new
GROUP BY insurance_provider;

-- Patients by Medical Condition
SELECT medical_condition, COUNT(*) AS total_patients
FROM patient_records_new
GROUP BY medical_condition;
```

```
-- Emergency Percentage
SELECT
(COUNT(CASE WHEN admission_type = 'Emergency' THEN 1 END) * 100.0)
 / COUNT(*) AS emergency_percentage
FROM patient_records_new;

-- Chronic Percentage
SELECT
(COUNT(CASE WHEN chronic_flag = 'Yes' THEN 1 END) * 100.0)
 / COUNT(*) AS chronic_percentage
FROM patient_records_new;

-- Abnormal Test Percentage
SELECT
(COUNT(CASE WHEN test_results = 'Abnormal' THEN 1 END) * 100.0)
 / COUNT(*) AS abnormal_test_percentage
FROM patient_records_new;

-- Average Length of Stay
SELECT AVG(length_of_stay) AS avg_length_of_stay
FROM patient_records_new;

-- Average Stay by Hospital
SELECT hospital, AVG(length_of_stay) AS avg_stay
FROM patient_records_new
GROUP BY hospital;

-- Doctor Workload
SELECT doctor, COUNT(*) AS total_patients
FROM patient_records_new
GROUP BY doctor
```

```
ORDER BY total_patients DESC;

-- Room Utilization
SELECT room_number, COUNT(*) AS total_admissions
FROM patient_records_new
GROUP BY room_number
ORDER BY total_admissions DESC;

-- Monthly Admission Trend
SELECT
FORMAT(date_of_admission, 'yyyy-MM') AS admission_month,
COUNT(*) AS total_admissions
FROM patient_records_new
GROUP BY FORMAT(date_of_admission, 'yyyy-MM')
ORDER BY admission_month;

-- Monthly Revenue Trend
SELECT
FORMAT(date_of_admission, 'yyyy-MM') AS admission_month,
SUM(billing_amount_num) AS monthly_revenue
FROM patient_records_new
GROUP BY FORMAT(date_of_admission, 'yyyy-MM')
ORDER BY admission_month;
GO
```