Views for Hospital Management System

Simple Views (PostgreSQL)

1. PatientDoctorDetails

```
CREATE OR REPLACE VIEW PatientDoctorDetails AS
SELECT
  p.patient_id,
  p.name AS patient name,
  p.age,
  p.gender,
  p.disease,
  pd.visit_date,
  pd.diagnosis,
  d.doctor id,
 d.name AS doctor_name,
  d.specialization
FROM
  PATIENT p
JOIN
  PATIENT_DOCTOR pd ON p.patient_id = pd.patient_id
JOIN
  DOCTOR d ON pd.doctor_id = d.doctor_id;
```

2. OutpatientDetails

```
CREATE OR REPLACE VIEW OutpatientDetails AS SELECT
p.patient_id,
p.name AS patient_name,
p.age,
p.gender,
p.disease,
o.visit date,
```

```
o.next_appointment_date,
o.treatment_details
FROM
    PATIENT p
JOIN
    OUTPATIENT o ON p.patient_id = o.patient_id;
```

3. PrescriptionDetails

```
CREATE OR REPLACE VIEW PrescriptionDetails AS SELECT
```

```
pr.prescription id,
  p.patient id,
  p.name AS patient_name,
  d.doctor id,
  d.name AS doctor_name,
  d.specialization,
  pr.medicine name,
  pr.medicine_quantity,
  pr.dosage,
  pr.prescribed_date
FROM
  PRESCRIPTION pr
JOIN
  PATIENT p ON pr.patient_id = p.patient_id
JOIN
  DOCTOR d ON pr.doctor id = d.doctor id;
```

4. MedicalTestDetails

```
CREATE OR REPLACE VIEW MedicalTestDetails AS SELECT
```

```
mt.test_id,
p.patient_id,
p.name AS patient_name,
d.doctor_id,
d.name AS doctor_name,
mt.test_date,
mt.test_name,
```

```
mt.test_results,
mt.cost

FROM
MEDICAL_TEST mt

JOIN
PATIENT p ON mt.patient_id = p.patient_id

JOIN
DOCTOR d ON mt.doctor_id = d.doctor_id;
```

5. NurseAssignments

```
CREATE OR REPLACE VIEW NurseAssignments AS
SELECT
  n.nurse_id,
  n.name AS nurse name,
  n.qualification,
  n.shift,
  w.ward id,
  w.ward_name,
  w.ward_type,
  wn.assignment_date
FROM
  NURSE n
JOIN
  WARD_NURSE wn ON n.nurse_id = wn.nurse_id
JOIN
  WARD w ON wn.ward id = w.ward id;
```

5 Materialized Views (PostgreSQL)

1. InpatientWardDetails

```
CREATE MATERIALIZED VIEW InpatientWardDetails AS SELECT p.patient_id,
```

```
p.name AS patient_name,
  p.age,
  p.gender,
  p.disease,
 i.admission date,
 i.discharge_date,
 w.ward id,
 w.ward name,
 w.ward type,
 w.charge per day,
 CASE
    WHEN i.discharge date IS NULL THEN CURRENT DATE - i.admission date
    ELSE i.discharge_date - i.admission_date
  END AS days stayed,
 CASE
    WHEN i.discharge_date IS NULL THEN (CURRENT_DATE - i.admission_date)
* w.charge_per_day
    ELSE (i.discharge date - i.admission date) * w.charge per day
  END AS ward charges
FROM
  PATIENT p
JOIN
  INPATIENT i ON p.patient id = i.patient id
JOIN
  WARD w ON i.ward id = w.ward id;
2. WardOccupancy
CREATE MATERIALIZED VIEW WardOccupancy AS
SELECT
  w.ward id,
 w.ward name,
 w.ward type,
 w.capacity,
 COUNT(i.patient id) AS occupied beds,
 w.capacity - COUNT(i.patient_id) AS available_beds,
  ROUND((COUNT(i.patient id)::numeric * 100 / w.capacity), 2) AS
```

occupancy percentage

FROM

WARD w

```
LEFT JOIN
   INPATIENT i ON w.ward_id = i.ward_id AND i.discharge_date IS NULL
GROUP BY
   w.ward_id, w.ward_name, w.ward_type, w.capacity;
```

3. BillingSummary

```
CREATE MATERIALIZED VIEW BillingSummary AS
SELECT
  b.bill id,
  p.patient_id,
  p.name AS patient name,
  d.doctor id,
  d.name AS doctor name,
  b.bill date,
  b.doctor_fee,
  b.room charge,
  b.test charge,
  b.total amount,
  CASE
    WHEN EXISTS (
      SELECT 1 FROM INPATIENT i WHERE i.patient id = p.patient id
    ) THEN 'Inpatient'
    ELSE 'Outpatient'
  END AS patient_type
FROM
  BILL b
JOIN
  PATIENT p ON b.patient_id = p.patient_id
JOIN
  DOCTOR d ON b.doctor id = d.doctor id;
```

4. DoctorPatientCount

```
CREATE MATERIALIZED VIEW DoctorPatientCount AS SELECT d.doctor_id, d.name AS doctor_name, d.specialization,
```

```
COUNT(DISTINCT pd.patient_id) AS total_patients,
    COUNT(DISTINCT i.patient_id) AS inpatient_count,
    COUNT(DISTINCT o.patient_id) AS outpatient_count
FROM
    DOCTOR d

LEFT JOIN
    PATIENT_DOCTOR pd ON d.doctor_id = pd.doctor_id

LEFT JOIN
    INPATIENT i ON pd.patient_id = i.patient_id

LEFT JOIN
    OUTPATIENT o ON pd.patient_id = o.patient_id

GROUP BY
    d.doctor_id, d.name, d.specialization;
```

5. MonthlyRevenueSummary

month;

```
CREATE MATERIALIZED VIEW MonthlyRevenueSummary AS
SELECT

DATE_TRUNC('month', bill_date) AS month,
SUM(doctor_fee) AS total_doctor_fee,
SUM(room_charge) AS total_room_charge,
SUM(test_charge) AS total_test_charge,
SUM(total_amount) AS total_revenue
FROM
BILL
GROUP BY
DATE_TRUNC('month', bill_date)
ORDER BY
```