

MOHAMED MOSTAFA | 22-101203

```
-- 1. Get patient appointment history (Patient Name, Appointment Time, Type of illness)
--     with provider details (Provider Name, Specialty) and payment info (PaymentAmount, PaymentMethod)
--     Order them decsendingly by appointment time
-- Frequency: Daily - For patient history lookups
SELECT
    p.Name AS PatientName,
    hp.Name AS ProviderName,
    hp.Specialty,
    a.Time,
    a.Type_of_illness,
    a.PaymentAmount,
    a.PaymentMethod
FROM Patient p
JOIN Appointment a ON p.PatientID = a.PatientID
JOIN HealthProviderAppointments hpa ON a.AppointmentID = hpa.AppointmentID
JOIN HealthProvider hp ON hpa.ProviderID = hp.ProviderID
WHERE p.PatientID = 'PAT001'
ORDER BY a.Time DESC;
```

110 %

Results Messages

	PatientName	ProviderName	Specialty	Time	Type_of_illness	PaymentAmount	PaymentMethod
1	Alice Thompson	Dr. Sarah Johnson	Cardiology	2024-01-20	Routine Checkup	150	Cash

```
-- 2. Find provider names and specialties with the total number of emergency appointments scheduled with them
--     Order them decsendingly by 'total number of emergency appointments'
SELECT
    hp.Name,
    hp.Specialty,
    COUNT(a.AppointmentID) AS EmergencyCount
FROM HealthProvider hp
JOIN HealthProviderAppointments hpa ON hp.ProviderID = hpa.ProviderID
JOIN Appointment a ON hpa.AppointmentID = a.AppointmentID
WHERE a.EmergencyStatus = 'High'
GROUP BY hp.Name, hp.Specialty
ORDER BY EmergencyCount DESC;
```

110 %

Results Messages

	Name	Specialty	EmergencyCount
1	Dr. Sarah Johnson	Cardiology	1
2	Dr. David Kim	Neurology	1

```
-- 3. Calculate insurance coverage statistics by CompanyName and packageID (CompanyName, PackageID, EnrolledPatients, AverageCoverage)
-- Frequency: Yearly - For insurance analysis
SELECT
    ic.CompanyName,
    p.PackageID,
    COUNT(pt.PatientID) AS EnrolledPatients,
    AVG(pd.Percentage) AS AverageCoverage
FROM InsuranceCompany ic
JOIN InsuranceCompanyPackages icp ON ic.InsuranceID = icp.InsuranceID
JOIN Package p ON icp.PackageID = p.PackageID
JOIN PackageDetails pd ON p.PackageID = pd.PackageID
-- LEFT JOIN to display CompanyNames & PackageIDs with zero EnrolledPatients
LEFT JOIN Patient pt ON ic.InsuranceID = pt.InsuranceID AND p.PackageID = pt.PackageID
GROUP BY ic.CompanyName, p.PackageID;
```

110 %

Results Messages

	CompanyName	PackageID	EnrolledPatients	AverageCoverage
1	Care First	P1	0	0.850000
2	HealthGuard	P1	2	0.850000
3	HealthGuard	P2	1	0.700000
4	MediCare Plus	P3	1	0.850000
5	Shield Insurance	P5	0	0.600000
6	WellCare	P4	1	0.950000

```
-- 4. Calculate average payment amounts by specialty and emergency status.
-- Display a column for the number of appointments for each specialty and emergency status as well.
-- Frequency: Monthly - For financial analysis
```

```
SELECT
    hp.Specialty,
    a.EmergencyStatus,
    AVG(a.PaymentAmount) AS AvgPayment,
    COUNT(a.AppointmentID) AS AppointmentCount
FROM HealthProvider hp
JOIN HealthProviderAppointments hpa ON hp.ProviderID = hpa.ProviderID
JOIN Appointment a ON hpa.AppointmentID = a.AppointmentID
GROUP BY hp.Specialty, a.EmergencyStatus;
```

10 %

Results Messages

	Specialty	EmergencyStatus	AvgPayment	AppointmentCount
1	Cardiology	High	500	1
2	Neurology	High	300	1
3	Cardiology	Low	150	1
4	Pediatrics	Low	100	1
5	Orthopedics	Mid	250	1

```
-- 5. Display the coverage percentage and number of patients for each insurance companies (Company Name, Average Coverage, Enrolled Patients)
-- Order by the average coverage descending
-- Frequency: Quarterly - For insurance analysis
```

```
SELECT
    ic.CompanyName,
    AVG(pd.Percentage) AS AvgCoverage,
    COUNT(DISTINCT p.PatientID) AS EnrolledPatients
FROM InsuranceCompany ic
JOIN InsuranceCompanyPackages icp ON ic.InsuranceID = icp.InsuranceID
JOIN PackageDetails pd ON icp.PackageID = pd.PackageID
-- LEFT JOIN to display rows with zero EnrolledPatients
LEFT JOIN Patient p ON ic.InsuranceID = p.InsuranceID
GROUP BY ic.CompanyName
ORDER BY AvgCoverage DESC;
```

110 %

Results Messages

	CompanyName	AvgCoverage	EnrolledPatients
1	WellCare	0.950000	1
2	Care First	0.850000	0
3	MediCare Plus	0.850000	1
4	HealthGuard	0.800000	2
5	Shield Insurance	0.600000	0

```
-- 6. Find patients with expired cards (PatientName, CardType, ExpirationDate, BankName)
-- Frequency: Daily - For payment validation
```

```
SELECT
    p.Name AS PatientName,
    c.CardType,
    c.ExpirationDate,
    c.BankName
FROM Patient p
JOIN Card c ON p.PatientID = c.PatientID
WHERE c.ExpirationDate < GETDATE();
```

110 %

Results Messages

PatientName	CardType	ExpirationDate	BankName
-------------	----------	----------------	----------

-- 7. Find providers (ProviderID, Provider Name) who generate more notifications than average.

```
SELECT ProviderID, Name
FROM HealthProvider
WHERE ProviderID IN (
    SELECT ProviderID
    FROM ReportGeneration
    GROUP BY ProviderID
    HAVING COUNT(NotificationID) > (
        SELECT AVG(temp.notification_count) FROM (
            SELECT ProviderID, COUNT(NotificationID) AS notification_count
            FROM ReportGeneration
            GROUP BY ProviderID
        ) AS temp
    )
)
```

110 %

Results Messages

	ProviderID	Name
1	HP001	Dr. Sarah Johnson

-- 8. List patients (Patient ID, Patient Name) receiving notifications more frequently than the average.

```
SELECT PatientID, Name
FROM Patient
WHERE PatientID IN (
    SELECT PatientID
    FROM Notification
    GROUP BY PatientID
    HAVING COUNT(NotificationID) > (
        SELECT AVG(temp.notification_count)
        FROM (
            SELECT PatientID, COUNT(NotificationID) AS notification_count
            FROM Notification
            GROUP BY PatientID
        ) AS temp
    )
);
```

.10 %

Results Messages

PatientID	Name
-----------	------

-- 9. Retrieve appointments scheduled after the last appointment for a specific patient.
-- Frequency: Used in patient appointment tracking.

```
SELECT AppointmentID, Time
FROM Appointment
WHERE Time > (
    SELECT MAX(Time)
    FROM Appointment
    WHERE PatientID = 'P123'
);
```

110 %

Results Messages

AppointmentID	Time
---------------	------

```
-- 10. List all patients (Patient ID, Patient Name) without an assigned insurance
SELECT PatientID, Name
FROM Patient
WHERE InsuranceID IS NULL;
```

110 %

Results Messages

	PatientID	Name
1	PAT003	Carol White

```
-- 11. Find patients (Patient ID, Patient Name) with no associated caregiver
-- Method 1:
SELECT PatientID, Name
FROM Patient
WHERE PatientID NOT IN (SELECT PatientID FROM Caregiver)
-- Method 2:
SELECT PatientID, Name
FROM Patient
WHERE PatientID IN (
SELECT PatientID FROM Patient
EXCEPT
SELECT PatientID FROM Caregiver
)
```

110 %

Results Messages

	PatientID	Name
--	-----------	------

```
-- 12. Count the number of appointments by each type of payment method
SELECT PaymentMethod, COUNT(AppointmentID) AS PaymentCount
FROM Appointment
GROUP BY PaymentMethod;
```

110 %

Results Messages

	PaymentMethod	PaymentCount
1	Cash	2
2	Credit	2
3	Debit	1

YOUSSEF WALID | 22-101048

```
--1.How many health records each regulator accessed (get their name and position) and how many from those reports are unique pateints
-- Frequency: Weekly - For compliance monitoring
SELECT
    gr.Name AS RegulatorName,
    gr.Position,
    COUNT(rar.RecordID) AS AccessedRecords,
    COUNT(DISTINCT hr.PatientID) AS UniquePatients
FROM GovernmentRegulator gr
JOIN Regulator_Access_HealthRecord rar ON gr.RegulatorID = rar.RegulatorID
JOIN HealthRecord hr ON rar.RecordID = hr.RecordID
GROUP BY gr.Name, gr.Position;
```

```
-- 2.Find the volume of notification sent for each type and how many unique patient they reached and how many of those was caregivers
-- Frequency: Monthly - For communication optimization
SELECT
    n.NotificationType,
    COUNT(*) AS TotalNotifications,
    COUNT(DISTINCT n.PatientID) AS UniquePatients,
    COUNT(DISTINCT cn.Name) AS CaregiversNotified
FROM Notification n
LEFT JOIN CaregiversNotifications cn ON n.NotificationID = cn.NotificationID
GROUP BY n.NotificationType;
```

```
23
24 -- 3. Find high-frequency patients by their names in the last 6 months and how many times they visited and how much they spend in total $
25 -- Frequency: Monthly - For patient monitoring
26 SELECT
27     p.Name,
28     COUNT(*) AS VisitCount,
29     SUM(a.PaymentAmount) AS TotalPayments
30 FROM Patient p
31 JOIN Appointment a ON p.PatientID = a.PatientID
32 WHERE a.Time >= DATEADD(MONTH, -6, GETDATE())
33 GROUP BY p.Name
34 HAVING COUNT(*) > 3
35 ORDER BY VisitCount DESC;
36
```

```
-- 4. Polularity of payment methods with each group and the average spending of appontment by each age group and payment method
-- Frequency: Quarterly - For financial planning
SELECT
    FLOOR(p.Age/10)*10 AS AgeGroup,
    a.PaymentMethod,
    COUNT(*) AS PaymentCount,
    AVG(a.PaymentAmount) AS AvgPayment
FROM Patient p
JOIN Appointment a ON p.PatientID = a.PatientID
GROUP BY FLOOR(p.Age/10)*10, a.PaymentMethod
ORDER BY AgeGroup;
```

```
-- 5. Find insurance companies name and illness type and the freqency of them and the average claim amount for each illness and company
-- Frequency: Monthly - For insurance analysis
SELECT
    ic.CompanyName,
    pd.IllnessType,
    COUNT(*) AS ClaimCount,
    AVG(a.PaymentAmount) AS AvgClaimAmount
FROM InsuranceCompany ic
JOIN Patient p ON ic.InsuranceID = p.InsuranceID
JOIN PackageDetails pd ON p.PackageID = pd.PackageID
JOIN Appointment a ON p.PatientID = a.PatientID
WHERE p.InsuranceStatus = 1
GROUP BY ic.CompanyName, pd.IllnessType
ORDER BY ClaimCount DESC;
```

```

-- 6. Find each medical speciality and the total emergencies and unique patients and the average cost each speciality served
-- Frequency: Monthly - For resource planning
SELECT
    hp.Specialty,
    COUNT(*) AS TotalEmergencies,
    COUNT(DISTINCT p.PatientID) AS UniquePatients,
    AVG(a.PaymentAmount) AS AvgEmergencyCost
FROM HealthProvider hp
JOIN HealthProviderAppointments hpa ON hp.ProviderID = hpa.ProviderID
JOIN Appointment a ON hpa.AppointmentID = a.AppointmentID
JOIN Patient p ON a.PatientID = p.PatientID
WHERE a.EmergencyStatus = 'High'
GROUP BY hp.Specialty
ORDER BY TotalEmergencies DESC;

```

```

-- 7. Find doctor names with their speciality and how many patients they have treated with how diverse the age is and the average patient age
--for each doctor
-- Frequency: Quarterly - For demographic analysis
SELECT
    hp.Name AS DoctorName,
    hp.Specialty,
    COUNT(DISTINCT p.PatientID) AS TotalPatients,
    MAX(p.Age) - MIN(p.Age) AS AgeRange,
    AVG(p.Age) AS AvgPatientAge
FROM HealthProvider hp
JOIN HealthProviderAppointments hpa ON hp.ProviderID = hpa.ProviderID
JOIN Appointment a ON hpa.AppointmentID = a.AppointmentID
JOIN Patient p ON a.PatientID = p.PatientID
GROUP BY hp.Name, hp.Specialty
HAVING COUNT(DISTINCT p.PatientID) > 2
ORDER BY AgeRange DESC;

```

```

-- 8. List reports generated after the most recent report by a specific regulator.
-- Frequency: Used in report generation tracking.
SELECT ReportID, GenerateDate
FROM Report
WHERE GenerateDate > (
    SELECT MAX(GenerateDate)
    FROM GovernmentRegulatorReports gr
    JOIN Report r ON gr.ReportID = r.ReportID
    WHERE gr.RegulatorID = 'R101'
);

```

```

-- 9. Identify patients with more emergency appointments than the average.
-- Frequency: Useful for emergency care analytics.
SELECT PatientID, Name
FROM Patient
WHERE PatientID IN (
    SELECT PatientID
    FROM Appointment
    WHERE EmergencyStatus = 'High'
    GROUP BY PatientID
    HAVING COUNT(AppointmentID) > (
        SELECT AVG(emergency_appointments)
        FROM (
            SELECT PatientID, COUNT(AppointmentID) AS emergency_appointments
            FROM Appointment
            WHERE EmergencyStatus = 'High'
            GROUP BY PatientID
        ) AS temp
    )
);

```

```

-- 10. Find patients with higher payment totals than average.
-- Frequency: Common in patient financial analysis.
SELECT PatientID, Name
FROM Patient
WHERE PatientID IN (
    SELECT PatientID
    FROM Appointment
    GROUP BY PatientID
    HAVING SUM(PaymentAmount) > (
        SELECT AVG(total_payment)
        FROM (
            SELECT PatientID, SUM(PaymentAmount) AS total_payment
            FROM Appointment
            GROUP BY PatientID
        ) AS temp
    )
);

```

ADHAM SOBHY | 23-101003

MOHAMED IBRAHIM | 22-101058