CREATE TABLE PatientRecords ( PatientID INTEGER PRIMARY KEY,

Name TEXT, Age INTEGER,

Gender TEXT, DiabetesType TEXT, LastConsultation DATE, HbA1c FLOAT

);

INSERT INTO PatientRecords (PatientID, Name, Age, Gender, DiabetesType, LastConsultation, HbA1c)

# VALUES

(1, 'John Doe', 45, 'Male', 'Type 2', '2024-03-25', 7.2),

(2, 'Jane Smith', 52, 'Female', 'Type 1', '2024-03-20', 6.5),

(3, 'Emily Davis', 38, 'Female', 'Type 2', '2024-03-22', 8.1),

(4, 'Michael Brown', 60, 'Male', 'Type 2', '2024-02-28', 7.5),

(5, 'Jessica Wilson', 44, 'Female', 'Type 1', '2024-03-15', 6.8),

(6, 'William Johnson', 35, 'Male', 'Type 1', '2024-03-30', 7.0),

(7, 'Olivia Martin', 29, 'Female', 'Type 1', '2024-03-05', 7.3),

(8, 'James Taylor', 48, 'Male', 'Type 2', '2024-01-29', 8.4),

(9, 'Laura Anderson', 55, 'Female', 'Gestational', '2024-03-11', 5.9),

(10, 'Robert Thomas', 50, 'Male', 'Type 2', '2024-03-18', 7.6);

CREATE TABLE AppointmentDetails ( AppointmentID INTEGER PRIMARY KEY, PatientID INTEGER,

AppointmentDate DATE, HealthcareProfessional TEXT, VisitPurpose TEXT, ConsultationNotes TEXT,

FOREIGN KEY (PatientID) REFERENCES PatientRecords(PatientID)

);

INSERT INTO AppointmentDetails (AppointmentID, PatientID, AppointmentDate, HealthcareProfessional, VisitPurpose, ConsultationNotes)

# VALUES

(1, 1, '2024-04-05', 'Dr. Sarah Lee', 'Routine Check-up', 'Patient managing well, continue current medication'),

(2, 2, '2024-04-03', 'Dr. Mike Brown', 'Routine Check-up', 'Adjust insulin dosage'),

(3, 3, '2024-04-10', 'Dr. Sarah Lee', 'Follow-up', 'Check blood pressure and adjust medication'),

(4, 4, '2024-03-29', 'Dr. Emily Clark', 'Routine Check-up', 'Recommend dietary changes'),

(5, 5, '2024-04-15', 'Dr. Mike Brown', 'Emergency', 'Hypoglycemic event, provided treatment'),

(6, 6, '2024-04-08', 'Dr. Sarah Lee', 'Follow-up', 'Good progress, keep up with the exercise'),

(7, 7, '2024-04-12', 'Dr. Emily Clark', 'Routine Check-up', 'Adjust medication, monitor closely'),

(8, 8, '2024-03-25', 'Dr. Mike Brown', 'Routine Check-up', 'Stable condition, continue treatment'),

(9, 9, '2024-04-20', 'Dr. Sarah Lee', 'Follow-up', 'Gestational diabetes management'),

(10, 10, '2024-04-18', 'Dr. Emily Clark', 'Routine Check-up', 'Encouraged weight management'),

(11, 1, '2024-05-05', 'Dr. Sarah Lee', 'Follow-up', 'Continue with current plan, next checkup in 3 months'),

(12, 2, '2024-05-03', 'Dr. Mike Brown', 'Emergency', 'Advised hospital visit for further tests'),

(13, 3, '2024-05-10', 'Dr. Sarah Lee', 'Routine Check-up', 'Stable HbA1c, review in 6 months'),

(14, 5, '2024-05-15', 'Dr. Mike Brown', 'Follow-up', 'Positive response to new insulin'),

(15, 7, '2024-05-12', 'Dr. Emily Clark', 'Routine Check-up', 'Continue monitoring glucose levels');

CREATE TABLE HealthcareProfessionals ( ProfessionalID INTEGER PRIMARY KEY, Name TEXT,

Specialty TEXT, ContactInfo TEXT

);

INSERT INTO HealthcareProfessionals (ProfessionalID, Name, Specialty, ContactInfo)

# VALUES

(1, 'Dr. Sarah Lee', 'Endocrinology', 'sarah.lee@example.com'),

(2, 'Dr. Mike Brown', 'General Practice', 'mike.brown@example.com'), (3, 'Dr. Emily Clark', 'Endocrinology', 'emily.clark@example.com'), (4, 'Dr. John Carter', 'Cardiology', 'john.carter@example.com'),

(5, 'Dr. Olivia White', 'Nephrology', 'olivia.white@example.com'), (6, 'Dr. Lucas Graham', 'Endocrinology', 'lucas.graham@example.com'), (7, 'Dr. Sophia Taylor', 'General Practice',

'sophia.taylor@example.com'),

(8, 'Dr. Mason Scott', 'Cardiology', 'mason.scott@example.com'),

(9, 'Dr. Isabella Hall', 'Nephrology', 'isabella.hall@example.com'), (10, 'Dr. Ethan King', 'Endocrinology', 'ethan.king@example.com');

CREATE TABLE MedicationsPrescribed ( PrescriptionID INTEGER PRIMARY KEY, AppointmentID INTEGER, MedicationName TEXT,

Dosage TEXT, Instructions TEXT,

FOREIGN KEY (AppointmentID) REFERENCES

AppointmentDetails(AppointmentID)

);

INSERT INTO MedicationsPrescribed (PrescriptionID, AppointmentID, MedicationName, Dosage, Instructions)

# VALUES

(1, 1, 'Metformin', '500mg', 'Twice a day with meals'),

(2, 2, 'Insulin Glargine', '20 units', 'Once a day at bedtime'), (3, 3, 'Lisinopril', '10mg', 'Once a day in the morning'),

(4, 4, 'Atorvastatin', '40mg', 'Once a day in the evening'), (5, 5, 'Metformin', '850mg', 'Twice a day with meals'),

(6, 6, 'Humalog', 'As per need', 'Before meals'),

(7, 7, 'Metformin', '1000mg', 'Twice a day with meals'), (8, 8, 'Insulin Aspart', '25 units', 'Twice a day'),

(9, 9, 'Simvastatin', '20mg', 'Once a day in the evening'), (10, 10, 'Amlodipine', '5mg', 'Once a day in the morning'), (11, 11, 'Metformin', '750mg', 'Twice a day with meals'),

(12, 12, 'Insulin Detemir', '18 units', 'Once a day at bedtime'),

(13, 13, 'Hydrochlorothiazide', '25mg', 'Once a day in the morning'), (14, 14, 'Glipizide', '10mg', 'Twice a day, 30 minutes before meals'), (15, 15, 'Insulin Lispro', 'As per need', 'Before meals');

CREATE TABLE Transactions ( TransactionID INTEGER PRIMARY KEY, PatientID INTEGER,

TransactionDate DATE, ServiceProvided TEXT, AmountCharged FLOAT,

FOREIGN KEY (PatientID) REFERENCES PatientRecords(PatientID)

);

INSERT INTO Transactions (TransactionID, PatientID, TransactionDate, ServiceProvided, AmountCharged)

# VALUES

(1, 1, '2024-04-05', 'Consultation', 100.00),

(2, 2, '2024-04-03', 'Lab Test: Blood Panel', 200.00),

(3, 3, '2024-04-10', 'Medication: Insulin', 50.00),

(4, 4, '2024-03-29', 'Consultation', 100.00),

(5, 5, '2024-04-15', 'Emergency Service', 300.00),

(6, 6, '2024-04-08', 'Consultation', 100.00),

(7, 7, '2024-04-12', 'Lab Test: HbA1c', 150.00),

(8, 8, '2024-03-25', 'Medication: Metformin', 30.00),

(9, 9, '2024-04-20', 'Consultation', 100.00),

(10, 10, '2024-04-18', 'Lab Test: Cholesterol', 180.00),

(11, 1, '2024-05-05', 'Medication: Blood Pressure', 60.00),

(12, 2, '2024-05-03', 'Emergency Service', 300.00),

(13, 3, '2024-05-10', 'Consultation', 100.00),

(14, 5, '2024-05-15', 'Medication: Insulin', 50.00),

(15, 7, '2024-05-12', 'Lab Test: Kidney Function', 170.00); CREATE TABLE Patients (

PatientID INTEGER PRIMARY KEY,

FullName TEXT, DateOfBirth DATE, Address TEXT, PhoneNumber TEXT, Email TEXT,

MedicalHistorySummary TEXT

);

INSERT INTO Patients (PatientID, FullName, DateOfBirth, Address, PhoneNumber, Email, MedicalHistorySummary)

# VALUES

(1, 'John Doe', '1979-02-15', '123 Elm St, Springfield', '555-0101', 'johndoe@email.com', 'Type 2 Diabetes, Hypertension'),

(2, 'Jane Smith', '1971-07-24', '456 Oak St, Rivertown', '555-0202', 'janesmith@email.com', 'Type 1 Diabetes'),

(3, 'Emily Davis', '1985-05-30', '789 Pine St, Lakeview', '555-0303', 'emilydavis@email.com', 'Type 2 Diabetes, High Cholesterol'),

(4, 'Michael Brown', '1964-04-12', '101 Maple Ave, Hillside', '555-

0404', 'michaelbrown@email.com', 'Type 2 Diabetes, Cardiopathy'),

(5, 'Jessica Wilson', '1976-08-09', '202 Birch Rd, Seaside', '555- 0505', 'jessicawilson@email.com', 'Type 1 Diabetes, Thyroid disorder'),

(6, 'William Johnson', '1988-03-15', '303 Cedar Ln, Greenwood', '555-

0606', 'williamjohnson@email.com', 'Type 1 Diabetes'),

(7, 'Olivia Martin', '1994-12-22', '404 Spruce St, Westfield', '555-

0707', 'oliviamartin@email.com', 'Type 1 Diabetes, Asthma'),

(8, 'James Taylor', '1975-06-05', '505 Elm St, Easton', '555-0808',

'jamestaylor@email.com', 'Type 2 Diabetes, Obesity'),

(9, 'Laura Anderson', '1969-11-08', '606 Oak St, Brookside', '555- 0909', 'lauraanderson@email.com', 'Gestational Diabetes'),

(10, 'Robert Thomas', '1974-01-20', '707 Pine St, Cliffside', '555-

1010', 'robertthomas@email.com', 'Type 2 Diabetes, Hypertension');

Select \* from Patients select \* from Transactions

select \* from AppointmentDetails select \* from HealthcareProfessionals select \* from MedicationsPrescribed ANSWERS

Appointment and Patient Data 1.

# SELECT

p.PatientID, p.Name,

MAX(a.AppointmentDate) AS LastAppointmentDate FROM

PatientRecords p LEFT JOIN

AppointmentDetails a ON p.PatientID = a.PatientID GROUP BY

p.PatientID, p.Name;

# 2. SELECT

p.PatientID, p.Name,

SUM(t.AmountCharged) AS TotalAmountCharged FROM

PatientRecords p INNER JOIN

Transactions t ON p.PatientID = t.PatientID GROUP BY

p.PatientID, p.Name;

# 3. SELECT

MedicationName,

COUNT(\*) AS PrescriptionCount FROM

MedicationsPrescribed GROUP BY

MedicationName ORDER BY

PrescriptionCount DESC LIMIT 1;

# 4. SELECT

p.PatientID, p.Name,

COUNT(a.AppointmentID) AS AppointmentCount,

ROW\_NUMBER() OVER (ORDER BY COUNT(a.AppointmentID) DESC) AS Rank

# FROM

PatientRecords p LEFT JOIN

AppointmentDetails a ON p.PatientID = a.PatientID GROUP BY

p.PatientID, p.Name ORDER BY

AppointmentCount DESC;

# 5. SELECT

p.PatientID, p.Name

# FROM

PatientRecords p LEFT JOIN

AppointmentDetails a ON p.PatientID = a.PatientID WHERE

a. AppointmentID IS NULL; To clarify whether paitents are SELECT p.PatientID, p.Name

FROM PatientRecords p

WHERE p.PatientID NOT IN (SELECT DISTINCT PatientID FROM

AppointmentDetails);

# 6. SELECT

p.PatientID, p.Name,

a. AppointmentDate AS CurrentAppointment,

LEAD(a.AppointmentDate) OVER (PARTITION BY p.PatientID ORDER BY a.AppointmentDate) AS NextAppointment

# FROM

PatientRecords p LEFT JOIN

AppointmentDetails a ON p.PatientID = a.PatientID ORDER BY

p.PatientID, a.AppointmentDate;

# 7. SELECT

h.ProfessionalID, h.Name

# FROM

HealthcareProfessionals h RIGHT JOIN

AppointmentDetails a ON h.Name = a.HealthcareProfessional WHERE

a. AppointmentID IS NULL;

used left join SELECT

h.ProfessionalID, h.Name

# FROM

HealthcareProfessionals h

# LEFT JOIN

AppointmentDetails a ON h.Name = a.HealthcareProfessional WHERE

a. AppointmentID IS NULL;

8.

WITH AppointmentDetailsWithLead AS ( SELECT

p.PatientID, p.Name,

a. AppointmentDate AS CurrentAppointment,

LEAD(a.AppointmentDate) OVER (PARTITION BY p.PatientID ORDER BY a.AppointmentDate) AS NextAppointment

# FROM

PatientRecords p LEFT JOIN

AppointmentDetails a ON p.PatientID = a.PatientID

# ) SELECT

PatientID, Name,

CurrentAppointment, NextAppointment

# FROM

AppointmentDetailsWithLead WHERE

NextAppointment IS NOT NULL

AND NextAppointment - CurrentAppointment <= 30;

# 9. SELECT

h.Name AS HealthcareProfessional, AVG(t.AmountCharged) AS AverageCharge

# FROM

HealthcareProfessionals h JOIN

AppointmentDetails a ON h.Name = a.HealthcareProfessional JOIN

Transactions t ON a.PatientID = t.PatientID AND a.AppointmentDate = t.TransactionDate

# GROUP BY

h.Name;

Medication and Revenue Analysis 1.

# SELECT

ranked\_appointments.HealthcareProfessional, ranked\_appointments.PatientID, ranked\_appointments.AppointmentDate

# FROM (

SELECT

a. HealthcareProfessional,

a. PatientID,

a. AppointmentDate,

DENSE\_RANK() OVER (PARTITION BY a.HealthcareProfessional ORDER BY a.AppointmentDate DESC) AS rank

# FROM

AppointmentDetails a

) AS ranked\_appointments WHERE

ranked\_appointments.rank = 1;

# 2. SELECT

p.PatientID, p.Name, CASE

WHEN m.MedicationName LIKE '%Insulin%' THEN 'Yes' ELSE 'No'

END AS PrescribedInsulin FROM

PatientRecords p JOIN

AppointmentDetails a ON p.PatientID = a.PatientID JOIN

MedicationsPrescribed m ON a.AppointmentID = m.AppointmentID;

# 3. SELECT

t.PatientID,

COUNT(t.TransactionID) AS NumberOfAppointments, SUM(t.AmountCharged) AS TotalAmountCharged

# FROM

Transactions t GROUP BY

t.PatientID;

# 4. SELECT

a. HealthcareProfessional,

COUNT(DISTINCT a.PatientID) AS UniquePatientsSeen,

RANK() OVER (ORDER BY COUNT(DISTINCT a.PatientID) DESC) AS

ProfessionalRank FROM

AppointmentDetails a GROUP BY

a. HealthcareProfessional;

Advanced Analysis with Subqueries and CTEs 1.

WITH AvgAppointmentCount AS ( SELECT

AVG(AppointmentCount) AS AvgCount FROM (

# SELECT

PatientID,

COUNT(AppointmentID) AS AppointmentCount FROM

AppointmentDetails GROUP BY

PatientID

) AS AppointmentCounts

# ) SELECT

p.PatientID,

p.Name,

COUNT(a.AppointmentID) AS AppointmentCount, CASE

WHEN COUNT(a.AppointmentID) > (SELECT AvgCount FROM AvgAppointmentCount) THEN 'Above Average'

WHEN COUNT(a.AppointmentID) < (SELECT AvgCount FROM AvgAppointmentCount) THEN 'Below Average'

ELSE 'At Par'

END AS ComparisonToAverage FROM

PatientRecords p LEFT JOIN

AppointmentDetails a ON p.PatientID = a.PatientID GROUP BY

p.PatientID, p.Name;

# 2. SELECT

p.PatientID, p.Name,

COALESCE(SUM(t.AmountCharged), 0) AS TotalChargedAmount FROM

PatientRecords p LEFT JOIN

Transactions t ON p.PatientID = t.PatientID GROUP BY

p.PatientID, p.Name;

3.

WITH MedicationCounts AS ( SELECT

pr.DiabetesType, mp.MedicationName,

COUNT(\*) AS MedicationCount FROM

PatientRecords pr JOIN

AppointmentDetails ad ON pr.PatientID = ad.PatientID JOIN

MedicationsPrescribed mp ON ad.AppointmentID = mp.AppointmentID GROUP BY

pr.DiabetesType, mp.MedicationName

),

RankedMedications AS ( SELECT

DiabetesType, MedicationName, MedicationCount,

ROW\_NUMBER() OVER (PARTITION BY DiabetesType ORDER BY

MedicationCount DESC) AS rn FROM

MedicationCounts

# ) SELECT

DiabetesType, MedicationName,

MedicationCount FROM

RankedMedications WHERE

rn = 1;

4.

WITH MonthlyAppointments AS ( SELECT

TO\_CHAR(AppointmentDate, 'YYYY-MM') AS Month, COUNT(AppointmentID) AS AppointmentCount

# FROM

AppointmentDetails GROUP BY

TO\_CHAR(AppointmentDate, 'YYYY-MM')

# ) SELECT

Month, AppointmentCount,

LAG(AppointmentCount) OVER (ORDER BY Month) AS PreviousMonthCount, (AppointmentCount - LAG(AppointmentCount) OVER (ORDER BY Month)) AS

Growth FROM

MonthlyAppointments ORDER BY

Month;

5.

WITH ProfessionalStats AS ( SELECT

a. HealthcareProfessional, COUNT(a.AppointmentID) AS AppointmentCount, SUM(t.AmountCharged) AS TotalRevenue

# FROM

AppointmentDetails a JOIN

Transactions t ON a.PatientID = t.PatientID AND a.AppointmentDate

= t.TransactionDate GROUP BY

a. HealthcareProfessional

# ) SELECT

HealthcareProfessional, AppointmentCount, TotalRevenue,

# CASE

WHEN TotalRevenue > (SELECT AVG(TotalRevenue) FROM ProfessionalStats) THEN 'Above Average Revenue'

WHEN TotalRevenue < (SELECT AVG(TotalRevenue) FROM ProfessionalStats) THEN 'Below Average Revenue'

ELSE 'Average Revenue' END AS RevenueComparison, CASE

WHEN AppointmentCount > (SELECT AVG(AppointmentCount) FROM ProfessionalStats) THEN 'Above Average Appointments'

WHEN AppointmentCount < (SELECT AVG(AppointmentCount) FROM ProfessionalStats) THEN 'Below Average Appointments'

ELSE 'Average Appointments' END AS AppointmentComparison

# FROM

ProfessionalStats ORDER BY

TotalRevenue DESC, AppointmentCount DESC;

# 6. SELECT

TO\_CHAR(a.AppointmentDate, 'YYYY-MM') AS Month, mp.MedicationName,

COUNT(mp.MedicationName) AS PrescriptionCount FROM

MedicationsPrescribed mp JOIN

AppointmentDetails a ON mp.AppointmentID = a.AppointmentID GROUP BY

TO\_CHAR(a.AppointmentDate, 'YYYY-MM'), mp.MedicationName ORDER BY

Month, MedicationName;

WITH MonthlyMedicationRanks AS ( SELECT

TO\_CHAR(a.AppointmentDate, 'YYYY-MM') AS Month, mp.MedicationName,

COUNT(mp.MedicationName) AS PrescriptionCount,

RANK() OVER (PARTITION BY TO\_CHAR(a.AppointmentDate, 'YYYY-MM')

ORDER BY COUNT(mp.MedicationName) DESC) AS MedicationRank FROM

MedicationsPrescribed mp JOIN

AppointmentDetails a ON mp.AppointmentID = a.AppointmentID GROUP BY

TO\_CHAR(a.AppointmentDate, 'YYYY-MM'), mp.MedicationName

# ) SELECT

Month, MedicationName, PrescriptionCount, MedicationRank

# FROM

MonthlyMedicationRanks ORDER BY

MedicationName, Month;

7.

WITH RankedServices AS ( SELECT

t.PatientID, t.ServiceProvided, t.AmountCharged,

DENSE\_RANK() OVER (PARTITION BY t.PatientID ORDER BY

t.AmountCharged DESC) AS ServiceRank FROM

Transactions t

# ) SELECT

PatientID, ServiceProvided, AmountCharged, ServiceRank

# FROM

RankedServices WHERE

ServiceRank <= 3 ORDER BY

PatientID, ServiceRank;

8.

WITH PatientPrescriptionCounts AS ( SELECT

p.PatientID,

p.Name AS PatientName, COUNT(mp.PrescriptionID) AS PrescriptionCount

# FROM

PatientRecords p JOIN

AppointmentDetails a ON p.PatientID = a.PatientID JOIN

MedicationsPrescribed mp ON a.AppointmentID = mp.AppointmentID GROUP BY

p.PatientID, p.Name

),

MostFrequentPatient AS ( SELECT

PatientID, PatientName, PrescriptionCount

# FROM

PatientPrescriptionCounts ORDER BY

PrescriptionCount DESC LIMIT 1

# ) SELECT

mfp.PatientID, mfp.PatientName, mp.MedicationName

# FROM

MostFrequentPatient mfp JOIN

AppointmentDetails a ON mfp.PatientID = a.PatientID JOIN

MedicationsPrescribed mp ON a.AppointmentID = mp.AppointmentID;

9.

WITH MonthlyRevenue AS ( SELECT

TO\_CHAR(t.TransactionDate, 'YYYY-MM') AS Month, SUM(t.AmountCharged) AS MonthlyRevenue

# FROM

Transactions t GROUP BY

TO\_CHAR(t.TransactionDate, 'YYYY-MM')

)

# SELECT

Month, MonthlyRevenue,

LAG(MonthlyRevenue) OVER (ORDER BY Month) AS PreviousMonthRevenue, (MonthlyRevenue - LAG(MonthlyRevenue) OVER (ORDER BY Month)) AS

RevenueChange FROM

MonthlyRevenue ORDER BY

Month;