**Simple Queries**

**a)** Create a list of patients and the medications they currently take. Sort your list by patient’s last name and medication name in alphabetical order. Include other applicable details such as date prescribed and dosage.

π P.LastName, M.MedicationName, M.Dose, M.PrescribeDate (γ patientid,(σ p . id = m . patientid (ρ m medications × ρ p people) ) )

SELECT P.LastName, M.MedicationName, M.Dose, M.PrescribeDate

FROM Medications as M, People as P

WHERE P.ID = M.PatientID

GROUP BY PatientID

ORDER BY LastName

**b)** Display patient information for patients with Delta Dental insurance policy.

PATIENT\_DELTA←(PATIENT ⋈ID =PatientID) (INSURANCE)

RESULT ← σ(InsuranceCompany = Delta Dental Insurance)(PATIENT\_DELTA)

SELECT PE.ID, PE.FirstName, PE.lastname, PE.PhoneNumber, PE.EmergencyContactNumber, PE.EmergencyContactRelationship

FROM Patients AS P, Insurance AS I, people as PE

WHERE I.InsuranceCompany = "Delta" AND PE.ID = I.PatientID;

**c)** Generate a list of procedures and dates of service performed by doctor Smilow.

MEDICAL\_ID ←EMPLOYEES ⋈ID = EmployeeID (MEDICAL)

SMILLOW\_ID ←σ(LastName = Smillow) (MEDICAL\_ID)

APPOINTMENT\_SMILLOW←APPOINTMENTS⋈ProcedureDoctor = LastName (SMILLOW\_ID)

RESULT ← π(TreatmentType, AppointmentDate)(APPOINTMENT\_SMILLOW))

SELECT A.ProcedureDoctor, TP.TreatmentType, A.AppointmentDate

FROM Treatments as TP, Appointments as A

WHERE A.ProcedureDoctor = "Smillow" AND TP.AppointmentID = A.AppointmentID

**d)** Print out a list of past-due invoices with patient contact information. Past due is defined As over 30 days old with a balance over $10.

DUE\_LIST ← σ(RemainingBalance > 10 AND Unpaid > 30)(INVOICE)

PATIENT\_LIST ← PATIENT⋈(PatientID = PatientID)(DUE\_LIST)

RESULT ← π(PatientID, InvoiceID, InvoiceAmount)(PATIENT\_LIST)

SELECT PT.ID, P.FirstName, P.LastName, DI.DateOfBirth

FROM Patients as PT, People as P, Invoice as I, Appointments as A, DemographicInfo as DI

WHERE PT.ID = A.PatientID and A.AppointmentDate = I.InvoiceDate AND I.DaysPastDue > 30 AND I.InvoicedAmount > 10;

**e)** Find the patients who brought the most revenue in the past year. You can define how many

records you want to display in the result of this query.

RESULT ← PatientID F(SUM PatientPaymentAmount) (Payment)

SELECT SUM(PP.PatientPaymentAmount) as Total, PE.LastName, PE.FirstName

FROM PatientPayments AS PP, Patients AS P, Appointments AS A, Invoice AS I, People as PE

WHERE P.ID = A.PatientID AND PP.InvoiceID = I.InvoiceID

ORDER BY Total DESC;

**f)** Create a list of doctors who performed less than 5 procedures this year.

MEDICAL\_ID ←EMPLOYEES ⋈EmployeeID = EmployeeID (MEDICAL)

APPOINTMENT\_MEDICALS←APPOINTMENTS ⟗ ProcedureDoctor = LastName (MEDICAL\_ID)

RESULT ← σ(AppointmentID\_Count < 5)(APPOINTMENT\_MEDICALS)

SELECT P.LastName, P.FirstName, C.ProcedureCompletionAmount, C.ProcedureCompleted

FROM CompletedProcedures AS C, Employees as E, People as P

WHERE E.ID = P.ID and C.ProcedureCompletionAmount < 5;

**g)** Find the highest paying procedures, procedure price, and the total number of those procedures performed. Sort your list with the highest paying procedures showing at the top of your list.

TREATMENT\_NUM ←Invoice ⟗ProcedurePerformed = TreatmentType (Treatment/Procedure)

MAX ←TreatmentType FMAX ProcedureAmount , COUNT TreatmentType (TREATMENT\_NUM)

SELECT T.TreatmentType, T.treatmentAmount, COUNT(T.TreatmentType) as Total

FROM Treatments as T

Group By T.treatmentamount

**h)** Create a list of all payment types accepted, the number of times each was used, and the total amount charged to that type of payment.

LIST ←TypeFCOUNT PaymentDate, SUM PatientPaymentAmount(PatientPayments)

π (PaymentType, Num\_Usage, Total\_Amount) (LIST)

SELECT PaymentType, COUNT(PaymentType), SUM(PatientPaymentAmount)

FROM PatientPayments

**i)** Find the name of the most popular insurance plan currently used by the patients.

INSURANCE\_NUM ←Insurance ⟗AmountCovered = PatientPaymentAmount (PatientPayments)

SUM ←InsuranceCompany FCOUNT PolicyID, COUNT InsuranceCompany, COUNT PatientID(INSURANCE)

SELECT I.InsuranceCompany, COUNT(I.insuranceCompany) As Total

FROM Insurance as I

**Extra Queries**

1. **outer joins**

English:

Insure <- SELECT PatientID, PolicyID, InsuranceCompany, StandardPerUnitCharges FROM Insurance

Patient <- SELECT PatientID, FirstName, LastName FROM Patients

InsureCover <- SELECT PolicyID, AcceptedInsurance, AmountCovered FROM CoveredInsurance

coveredPatient <- SELECT I.PatientID, I.PolicyID, I.InsuranceCompany, I.StandardPerUnitCharges, P.FirstName, P.LastName FROM Insure I LEFT JOIN Patient P ON P.PatientID = I.PatientID

insuranceCoveredPatient <- SELECT \*

FROM coveredPatient C

LEFT JOIN InsureCover IC ON IC.PolicyID = C.PolicyID

Relational Algebra:

Insure <- π (PatientID, PolicyID, InsuranceCompany, StandardPerUnitCharges) (Insurance)

Patient <- π (PatientID, FirstName, LastName) (Patients)

InsureCover <- π (PolicyID, AcceptedInsurance, AmountCovered) (CoveredInsurance)

coveredPatient <- Insure ⟕(PatientID = PatientID) (Patient)

insuranceCoveredPatient <- coveredPatient ⟕(PolicyID = PolicyID) (InsureCover)

SELECT I.PatientID, I.PolicyID, I.InsuranceCompany, I.StandardPerUnitCharges, PE.FirstName, PE.LastName, C.AcceptedInsurance, C.AmountCovered

FROM Insurance as I, Patients as P, CoversInsurance as C, People as PE

Where P.ID = I.PatientID and C.PolicyID = I.PolicyID;

1. **aggregate function**

English:

Patient <- SELECT PatientID, FirstName, LastName FROM Patients

Appoint <- SELECT a.PatientID, a.AppointmentID FROM Appointments as a

JOIN Patients on Patients.ID = a.PatientID

Procedures <- SELECT PatientID, PaymentHistory FROM TreatmentProcedure

allPatientProcedures <- Patient X Procedures

ret <- SELECT \* GROUP BY FirstName, LastName COUNT PaymentHistory FROM allPatientProcedures

Relational Algebra:

Patient <- π (PatientID, FirstName, LastName) (Patients)

Appoint <- π( a.PatientID, a.AppointmentID )( Appointments)

X Patients Patients.ID = a.PatientID

Procedures <- π (PatientID, PaymentHistory) (TreatmentProcedure)

allPatientProcedures <- Patient X Procedures

proceduresPerPatient <- FirstName, LastName F COUNT PaymentHistory (allPatientProcedures)

SQL:

Patient <- SELECT ID, FirstName, LastName FROM Patients

Appoint <- SELECT a.PatientID, a.AppointmentID FROM Appointments as a

JOIN Patients on Patients.ID = a.PatientID

Procedures <- SELECT ID, PaymentHistory FROM TreatmentProcedure

allPatientProcedures <- SELECT a.ID, a.FirstName, a.LastName, pr.PaymentHistory FROM Appoint as a JOIN Procedures as pr ON pr.AppointmentID = a.AppointmentID

ret <- SELECT COUNT(PaymentHistory) FROM allPatientProcedures

GROUP BY FirstName, LastName

1. **“extra” entities from PART 1**

English:

frequencyOfTools <- SELECT COUNT(XRayMachine), COUNT(Scaler), COUNT(Frocept), COUNT(File), COUNT(DentalProbe), COUNT(PowerTool), COUNT(DentalMirror) FROM MedicalEquipment GROUP BY EquipmentID

Relational Algebra:

frequencyOfTools <- EquipmentID F  COUNT XRayMachine, COUNT Scaler, COUNT Frocept, COUNT File, COUNT DentalProbe, COUNT PowerTool, COUNT DentalMirror (MedicalEquipment)

SQL:

frequencyOfTools <- SELECT (XRayMachine),

(Scaler),

(Forcept),

(ToolFile),

(DentalProbe),

(PowerTool),

(DentalMirror)

FROM MedicalEquipment GROUP BY EquipmentID