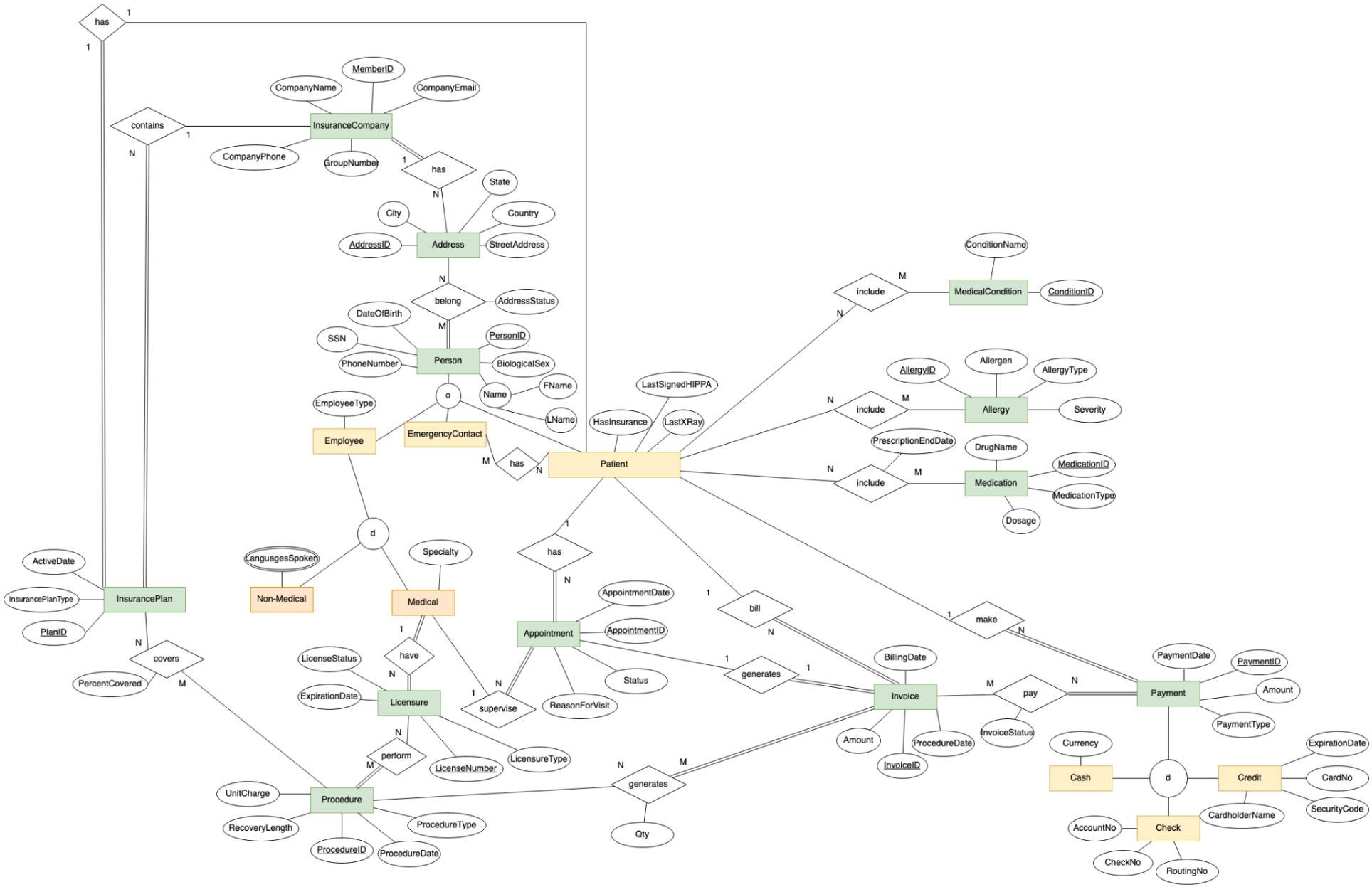
Part 2: Smilow Dentistry Relational Schema

Team 2: Anushka Nath, Sam Ziessler, Karl Chavez, Christina Duong

Intr Database Sys (7194)

February 2023

Part 1: Updated EERD



Part 2: Sentence Notation

Step 1: Regular Entities

Address(AddressID, City, State, Country, StreetAddress)

Person(PersonID, BiologicalSex, FName, LName, DateOfBirth, SSN, PhoneNumber)

InsuranceCompany(MemberID, CompanyEmail, GroupNumber, CompanyPhone)

InsurancePlan(PlanID,ActiveDate, InsurancePlanType)

Procedure(ProcedureID, UnitCharge, RecoveryLength, ProcedureDate, ProcedureType)

Invoice(InvoiceID, Amount, BillingDate, ProcedureDate)

Payment(PaymentID, PaymentDate, Amount, PaymentType)

Appointment(AppointmentID, AppointmentDate, Status, ReasonForVisit)

Allergy(AllergyID, Allergen, AllergyType, Severity)

Medication(MedicationID, DrugName, MedicationType, Dosage)

MedicalCondition(ConditionID, ConditionName)

Licensure(LicenseNumber, LicensureType, ExpirationDate, LicenseStatus)

Step 2:

N/A

Step 3: 1:N

Address (AddressID, StreetAddress, Zipcode, State, City, Country, MemberID (FK))

Payment (PaymentID, PaymentDate, Amount, PaymentType, PatientID (FK))

Invoice (InvoiceID, BillingDate, Amount, ProcedureDate, MedicalID (FK))

Appointment (AppointmentID, AppointmentDate, Status, ReasonForVisit, PatientID (FK))

Licensure (LicenseNumber, LicenseType, ExpirationDate, LicenseStatus, MedicalID (FK))

Step 4: 1:1

Invoice (InvoiceID, BillingDate, Amount, ProcedureDate, AppointmentID (FK), MedicalID (FK))

Appointment (AppointmentID, AppointmentDate, Status, ReasonForVisit, PatientID (FK), InvoiceID (FK))

InsurancePlan (PlanID, InsurancePlanType, ActiveDate, MemberID (FK), PatientID (FK))

Step 5: Binary M:N

InsurancePlan_Procedure (PlanID (FK), ProcedureID (FK), PercentCovered)

Procedure_Invoice (ProcedureID (FK), InvoiceID (FK), Qty)

Invoice_Payment (InvoiceID (FK), PaymentID (FK), InvoiceStatus)

Patient_Medication (PatientID (FK), MedicationID (FK), PrescripEnd, PrescripStart)

Patient_Allergy (PatientID (FK), AllergyID (FK))

Patient_MedicalCondition (PatientID (FK), ConditionID (FK), DiagnosisDate)

EmergencyContact_Patient (EmergencyPersonID (FK), Relationship, PatientID (FK))

Person_Address (AddressID (FK), PersonID (FK), AddressStatus)

Step 6: Multivalued

NonMedical_Languages (EmployeeID (FK), LanguagesSpoken)

Final Step: Combine Previous Steps + Add Specialization Features

Patient(PatientID (FK), LastSignedHIPPA, LastXRay)

Employee (EmployeeID (FK), MedicalFlag, Specialty)

- EmployeeID is a FK to Person.PersonID.

Payment (PaymentID, PaymentDate, Amount, PaymentType, CashFlag, Currency, PatientID (FK))

- PatientID is a FK to Patient.PatientID.

Licensure (LicenseNumber, LicenseType, ExpirationDate, LicenseStatus, MedicalID (FK))

- MedicalID is a FK to Employee.EmployeeID.

Invoice (InvoiceID, BillingDate, Amount, ProcedureDate, AppointmentID (FK), MedicalID (FK))

- MedicalID is a FK to Person.MedicalID.
- AppointmentID is a FK to Appointment.AppointmentID.

Appointment (AppointmentID, AppointmentDate, Status, ReasonForVisit, PatientID (FK), MedicalPersonID (FK), InvoiceID (FK))

- PatientID is a FK to Patient.PatientID.
- MedicalPersonID is a FK to Employee.EmployeeID.
- InvoiceID is a FK to Invoice.InvoiceID.

Check (PaymentID (FK), AccountNo, CheckNo, RoutingNo)

- PaymentID is a FK to Payment.PaymentID.

Credit (PaymentID (FK), CardholderName, SecurityCode, CardNo, ExpirationDate)

- PaymentID is a FK to Payment.PaymentID.

NonMedical_Languages (EmployeeID (FK), LanguagesSpoken)

- EmployeeID is a FK to Employee.EmployeeID.

InsurancePlan_Procedure (PlanID (FK), ProcedureID (FK), PercentCovered)

- PlanID is a FK to InsurancePlan.PlanID.
- ProcedureID is a FK to Procedure.ProcedureID.

Licensure_Procedure (LicenseNumber (FK), ProcedureID (FK))

- LicenseNumber is a FK to License.LicenseNumber.
- ProcedureID is a FK to Procedure.ProcedureID.

Procedure_Invoice (ProcedureID (FK), InvoiceID (FK), Qty)

- ProcedureID is a FK to Procedure.ProcedureID.

- InvoiceID is a FK to Invoice.InvoiceID.

Invoice_Payment (InvoiceID (FK), PaymentID (FK), InvoiceStatus)

- InvoiceID is a FK to Invoice.InvoiceID.
- PaymentID is a FK is Payment.PaymentID.

Patient_Medication (PatientID (FK), MedicationID (FK), PrescripEnd, PrescripStart)

- PatientID is a FK to Patient.PatientID.
- MedicationID is a FK to Medication.MedicationID.

Patient_Allergy (PatientID (FK), AllergyID (FK))

- PatientID is a FK to Patient.PatientID.
- AllergyID is a FK to Allergy.AllergyID.

Patient_MedicalCondition (PatientID (FK), ConditionID (FK), DiagnosisDate)

- PatientID is a FK to Patient.PatientID.
- ConditionID is a FK to Condition.ConditionID.

Person_Address (AddressID (FK), PersonID (FK), AddressStatus)

- AddressID is a FK to Address.AddressID.
- PersonID is a FK to Person.PersonID.

EmergencyContact_Patient (EmergencyPersonID (FK), Relationship, PatientID (FK))

- PatientID is a FK to Patient.PatientID.
- EmergencyPersonID is a FK to Person.PersonID.

Address (AddressID, StreetAddress, Zipcode, State, City, Country, MemberID (FK))

- MemberID is a FK to InsuranceCompany.MemberID.

InsurancePlan (PlanID, InsurancePlanType, ActiveDate, MemberID (FK), PatientID (FK))

- MemberID is a FK to InsuranceCompany.MemberID.
- PatientID is a FK to Patient.PatientID.

Person (PersonID, FName, LName, SSN, DateOfBirth, BiologicalSex, PhoneNumber)

InsuranceCompany (MemberID, CompanyName, CompanyEmail, GroupNumber, CompanyPhone)

Procedure (ProcedureID, UnitCharge, RecoveryLength, ProcedureType, ProcedureDate)

Allergy (AllergyID, Allergen, AllergyType, Severity)

Medication (MedicationID, DrugName MedicationType, Dosage)

MedicalCondition (ConditionID, ConditionName)

Part 3: Relational Queries

a. Create a list of patients and the medications they currently take

   i. Patient_Person ← $\Pi_{\text{(PatientIID, FName, LName)}}$(Person $\bowtie_{\text{PersonID = PatientID}}$ Patient)

   ii. Current_Medication ← $\sigma_{\text{PrescripStart<TODAY (PrescipEnd > TODAY OR PrescripEnd = NULL)}}$(Patient_Medication)

   iii. Medication_Info ← $\Pi_{\text{(PatientID, DrugName)}}$(Current_Medication * Medication)

   iv. **Patient_Medication ← $\Pi_{\text{(FName, LName, DrugName)}}$ (Patient_Person * Medication_Info)**

b. Display Patient information for patients who currently have Delta Dental insurance policy

   i. Patient_Person ←(Person $\bowtie_{\text{PersonID = PatientID}}$ Patient)

   ii. Plan_Company ← $\Pi_{\text{(PatientID, CompanyName)}}$ (InsurancePlan * InsuranceCompany)

   iii. **DeltaDental_Patients ← $\sigma_{\text{CompanyName="Delta Dental"}}$ (Patient_Person * Plan_Company)**

c. Generate a list of procedures and service dates performed by Dr. Smillow

   i. Medical_Employee ← $\sigma_{\text{MedicalFlag = "True"}}$(Employee)

   ii. Medical_Person ← Medical_Employee $\bowtie_{\text{PersonID = EmployeeID}}$ Person

   iii. Dr_Smillow ← $\sigma_{\text{Lname = "Smillow"}}$ Medical_Person

   iv. Smillow_LicenseNo ← Dr_Smillow * Licensure_Procedure

   v. **Smillow_Procedures ← Smillow_LicenseNo * Procedure**

   vi. Name_Date ← $\Pi_{\text{ProcedureID, ProcedureType, ProcedureDate}}$(Procedure)

   vii. **Result ← Smillow_Procedures * Name_Date**

d. Print out a list of due invoices with patient contact info.

   i. Past_Due ← $\sigma_{\text{(Amount > 10) AND (TODAY >= BillingDate + 30)}}$ Invoice

   ii. PastDue_Invoices ← Invoice * Past_Due

   iii. Patient_Info ←$\Pi_{\text{PhoneNumber}}$(Patient $\bowtie_{\text{PatientID = PersonID}}$ Person)

   iv. **Result ← PastDue_Invoices $\bowtie_{\text{PersonID = PatientID}}$ Patient_Info**

e. Find the patients who brought the most revenue in the past year

   i. Patient_Person ← Patient $\bowtie_{\text{PatientID = PersonID}}$Person

   ii. Invoices ← Patient_Person * Invoice

   iii. Current_Invoices ← $\sigma_{\text{BillingDate >= '2022-01-01' AND BillingDate < '2023-01-01'}}$(Invoices)

   iv. **Current_Invoice_Sums ← $_{\text{(Fname, Lname)}}\mathcal{F}_{\text{SUM Amount}}$(Current_Invoices)**

   v. *SQL features can now be used to sort the top X number of patients in Current_Invoice_Sums.*

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

   i. Medical_Employee ← $\sigma_{\text{MedicalFlag = "True"}}$(Employee)

   ii. Medical_Person ← Medical_Employee $\bowtie_{\text{EmployeeID = PersonID}}$Person

iii. LicenseNos ← Medical_Person * Licensure_Procedure

iv. Procedures ← LicenseNos * Procedure

v. Current_Procedures ← $\sigma_{\text{ProcedureDate > '2023-01-01'}}$(Procedures)

vi. Current_Procedures_Count ← $_{\text{(Fname, Lname)}}\mathcal{F}_{\text{COUNT ProcedureID}}$(Current_Procedures)

vii. **Result ← $\pi_{\text{Fname, Lname}}(\sigma_{\text{Count\_procedureid < 5}}$(Current_Procedures_Count))**

g. Find the highest paying procedures, procedure price, and the total number of those procedures performed.

i. Procedure_Info ← $\pi_{\text{ProcedureID, ProcedureType, UnitCharge}}$ (Procedure)

ii. Procedure_Count ← $_{\text{ProcedureType}}\mathcal{F}_{\text{COUNT ProcedureID}}$ (Procedure_Info)

iii. Procedure_Price ← $\pi_{\text{ProcedureType, UnitCharge, COUNT\_ProcedureID}}$(Procedure_Count * Procedure_Info)

iv. *SQL features can now be used to sort the top X highest paying procedures in Procedure_Price.*

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

i. Payment_Type_Amount ← $\pi_{\text{PaymentType, Amount}}$(Payment)

ii. Sum_Count ← $_{\text{PaymentType}}\mathcal{F}_{\text{COUNT PaymentType, SUM Amount}}$ (Payment_Type_Amount)

iii. **Result ← $\pi_{\text{PaymentType, COUNT Amount, SUM Amount}}$(Sum_Count)**

i. List ids and names of insurance plans ever used by patients and how many patients have that plan.

i. Plan_Description ← $\rho_{\text{PlanID, PlanName, ActiveDate, MemberID, PatientID}}$(InsurancePlan)

ii. Id_Name ← $\pi_{\text{PlanID, PlanName}}$(Plan_Description)

iii. Id_PlanName_Count ← $_{\text{PlanID, PlanName}}\mathcal{F}_{\text{COUNT PlanName}}$ (Id_Name)

iv. **Result ← $\pi_{\text{PlanID, PlanName, COUNT PlanName}}$(Id_PlanName_Count)**

Part 4: Additional Interesting Queries

a. Outerjoins: List all the employee names, and if they have a license, display their license number and expiration date. List all employee names regardless if they have a license.

   i. Employee_Person ← $\Pi_{(EmployeeID, FName, LName)}$(Person $\bowtie_{PersonID = EmployeeID}$ Employee)

   ii. Employee_License ← (Employee_Person $⟕_{(EmployeeID=MedicalID)}$ Licensure)

   iii. EmployeeInfo_License ← $\Pi_{(FName, LName, LicenseNumber, ExpirationDate)}$(Employee_License)

b. Aggregate Functions: Find the total number of allergies grouped by patient

   i. Patient_Person ← (Person $\bowtie_{PersonID = PatientID}$ Patient)

   ii. Result ← $_{PatientID}\mathcal{F}_{SUM(COUNT AllergyID)}$ Patient_Person $\bowtie_{PersonID = PatientID}$ Patient_Allergy

c. Extra Entities from PART 1: List the name and the count of addresses for each person from insurance company. Each person and company must have at least one address; therefore, NULL values are not allowed

   i. Company ← $\pi_{MemberID}$(Address)

   ii. Count_Company ← $_{MemberID}\mathcal{F}_{COUNT MemberID}$(Company)

   iii. Name_Count_Company ← $\pi_{CompanyName, COUNT MemberID}$ (Count_Company * InsuranceCompany)

   iv. Count_Person ← $_{PersonID}\mathcal{F}_{COUNT AddressID}$(Person_Address)

   v. Name_Count_Person ← $\pi_{Fname, Lname, COUNT PersonID}$ (Count_Person * Person)

   vi. Result ← Name_Count_Company ∪ Name_Count_Person

Part 5: Specification Sheets

Employee: The employee table serves to store the information of employees such as their employee ID, medical flag, and their specialty. Employee is a subclass of Person entity.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
EmployeeID	NUMERIC	FK	N	Y	FK to PersonID in Person relation
MedicalFlag	BOOLEAN		N	N	
Specialty	STRING (15)		Y	N	

Payment: The payment table serves to store information about payments made by the patient or invoice such as the amount, payment type, the currency the patient is using, and so forth. Payment is linked to patient and invoice.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
PaymentID	NUMERIC	PK	N	Y	
PaymentDate	DATETIME		N	N	
Amount	NUMERIC		N	N	
PaymentType	STRING (6)		N	N	“cash”, “credit”, “check”
CashFlag	BOOLEAN		N	N	
Currency	STRING (3)		N	N	This should be an existing currency.
PatientID	NUMERIC	FK	N	N	FK to PatientID in Patient relation

Licensure: The licensure table serves to store information on licensure that a medical employee has, and what procedures it can perform such as license status, license type, and so forth. Licensure is linked to medical and procedure.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
LicenseNumber	NUMERIC	PK	N	Y	
LicenseType	STRING (20)		N	N	
ExpirationDate	DATE		Y	N	NULL if no expiration date
LicenseStatus	STRING (15)		N	N	“active”, “inactive”, “expired”, “revoked”, “suspended”, “probationary”
MedicalID	NUMERIC	FK	N	N	FK to EmployeeID in Employee relation

Invoice: The invoice table serves to store information about the amount due for goods or services provided such as the billing date, amount, and so forth. Invoice is linked to payment, procedure, appointment, and medical.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
InvoiceID	NUMERIC	PK	N	Y	
BillingDate	DATE		N	N	
Amount	NUMERIC		N	N	
ProcedureDate	DATE		N	N	
AppointmentID	NUMERIC	FK	N	Y	FK to AppointmentID in Appointment relation

Appointment: The appointment table serves to store information about a person scheduling a meeting with a medical employee such as the appointment date, reason for the visit, and so forth. The appointment is linked to the patient, invoice, and medical.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
AppointmentID	NUMERIC	PK	N	Y	
AppointmentDate	DATETIME		N	N	
Status	STRING (9)		N	N	“active”, “cancelled”, “completed”
ReasonForVisit	STRING (20)		N	N	
PatientID	NUMERIC	FK	N	N	FK to PatientID in Patient relation
MedicalPersonID	NUMERIC	FK	N	N	FK to EmployeeID in Employee relation
InvoiceID	NUMERIC	FK	Y	Y	FK to InvoiceID in Invoice relation

Patient: The patient table serves to store information about the patient such as patient ID, insurance status, medical history, and other relevant details. By having access to this data, it makes it easier to schedule appointments, process invoices, and provide better care to patients.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
PatientID	NUMERIC	FK	N	Y	FK to PersonID in Person Relation
PlanID	NUMERIC	FK	Y	Y	FK to InsurancePlan Relation

Check: The check table makes it easier to keep track of check payments made by patients such as the routing number, the account number, and so forth. The company can easily search and retrieve payment history.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
PaymentID	NUMERIC	FK	N	N	FK to Payment Relation
AccountNo	NUMERIC		N	N	12-17 digits
CheckNo	NUMERIC		N	N	3-4 digits
RoutingNo	NUMERIC		N	N	9 digits

Credit: The credit table makes it easier to keep track of credit payments made by patients such security code, card number, and so forth. The company can easily search and retrieve payment history.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
PaymentID	NUMERIC	FK	N	N	FK to Payment Relation
CardholderName	STRING (50)		N	N	
SecurityCode	NUMERIC		N	N	3 digits
CardNo	NUMERIC		N	N	16 digits
ExpirationDate	DATE		N	N	

NonMedical_Languages: The non-medical languages table serves to manage multilingual content and provide language-specific information such as the languages spoken by patients. This allows employees to communicate with patients in the correct language, improving the quality of care provided.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
EmployeeID	NUMERIC	FK	N	N	FK to Employee Relation
LanguagesSpoken	STRING(20)		N	N	

InsurancePlan_Procedure: This stores the percentage of a procedure that is covered by an insurance plan. It references the Procedure and Insurance Plan.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
PlanID	NUMERIC	FK	N	N	FK to Plan Relation

ProcedureID	NUMERIC	FK	N	N	FK to Procedure Relation
PercentCovered	NUMERIC		N	N	100 ≥ PercentCovered

Licensure_Procedure: This stores the licensure that was applied to the procedure conducted on a patient(s). It references the License Relation and Procedure Relation.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
LicenseNumber	NUMERIC	FK	N	N	FK to License Relation
ProcedureID	NUMERIC	FK	N	N	FK to Procedure Relation

Procedure_Invoice: This stores the invoice that corresponds to a specific procedure conducted. It references the Procedure Relation and Invoice Relation.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
ProcedureID	NUMERIC	FK	N	N	FK to Procedure Relation
InvoiceID	NUMERIC	FK	N	N	FK to Invoice Relation
Qty	NUMERIC		N	N	Qty ≥ 0

Invoice_Payment: This links invoices to their corresponding payments and includes the status of the invoice. It references the Invoice relation and Payment relation.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
InvoiceID	NUMERIC	FK	N	N	FK to Invoice relation
PaymentID	NUMERIC	FK	N	N	FK to Payment relation
InvoiceStatus	STRING (8)		N	N	“pending” “paid”

Patient_Medication: The medication table serves to store information about a patient’s medication history, including details about specific drugs being taken like name and dosage. Often, dentists look back to medication to avoid overdosage or ensure proper treatment options.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
PatientID	NUMERIC	FK	N	N	FK to Patient relation
MedicationID	NUMERIC	FK	N	N	FK to Medication relation

PrescripStart	DATE		N	N	PrescriptStart can't be after PrescriptEnd
PrescriptEnd	DATE		N	N	PrescriptEnd can't be before PrescriptStart

Patient_Allergy: The allergy table serves to store information about a patient's existing allergies, including details about specific allergies being taken like the allergen and severity. Often, dentists look back to allergies to avoid prescribing a medication that overlaps with a patient's allergy, among other reasons.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
PatientID	NUMERIC	FK	N	N	FK to Patient relation
AllergyID	NUMERIC	FK	N	N	FK to Allergy relation

Patient_MedicalCondition: The medical condition table serves to store information about a patient's pre-existing medical conditions, which may serve as crucial information for visitations. Examples of medical conditions can look like periodontal disease or tooth decay.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
PatientID	NUMERIC	FK	N	N	FK to Patient relation
ConditionID	NUMERIC	FK	N	N	FK to Condition relation
DiagnosisDate	DATE		N	N	

Person_Address: The address table serves to store information about any person's address, including details like country and street address. This information is vital in terms of knowing where to send specific bills to, specific to the person charged. This table is linked to the InsuranceCompany relation.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
AddressID	NUMERIC	FK	N	N	FK to Address relation
PersonID	NUMERIC	FK	N	N	FK to Person relation
AddressStatus	STRING (8)		N	N	"valid" "invalid"

EmergencyContact_Patient: This table relates patients to their emergency contacts and includes the relationship between the patient and the emergency contact.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
EmergencyPersonID	NUMERIC	FK	N	N	FK to Person relation
PatientID	NUMERIC	FK	N	N	FK to Patient relation
HasCompletedForm	BOOLEAN		N	N	

Relationship	STRING (15)		N	N	“mother” “father”, etc..

Address: The address table serves to store information about an address, including details like country and street address. This information is vital in terms of knowing where to send specific bills to. This table is linked to the InsuranceCompany relation.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
AddressID	NUMERIC	PK	N	Y	
MemberID	NUMERIC	FK	Y	N	FK to InsuranceCompany relation
StreetAddress	STRING		N	N	Must be an existing Address
Zipcode	NUMERIC		N	N	5 digits
State	STRING (2)		Y	N	Must be an existing State, Not all countries have state
City	STRING (60)		N	N	Must be an existing city
Country	STRING (56)		N	N	Must be an existing country

InsurancePlan: The insurance plan table serves to store information about insurance, including details about plans like patients with corresponding insurance plans, as well as which type of plan they have. This table is linked to the InsuranceCompany relation and Patient relation.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
PlanID	NUMERIC	PK	N	Y	
MemberID	NUMERIC	FK	N	N	FK to InsuranceCompany relation
PatientID	STRING	FK	N	Y	FK to Patient relation
InsurancePlanType	STRING (50)		Y	N	
ActiveDate	DATE		Y	N	

Person: The person table serves to store general information about any person involved with Smilow Dentistry, including details like first name and biological sex. This information is absolutely crucial in determining demographic information regarding involved parties, whether in a professional or customer viewpoint.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes

PersonID	NUMERIC	PK	N	Y	
Fname	STRING (35)		N	N	
Lname	STRING (35)		N	N	
SSN	NUMERIC		N	N	9 digits
DateOfBirth	DATE		N	N	
BiologicalSex	STRING(8)		N	N	“male”, “female”, “intersex”
PhoneNumber	NUMERIC		N	N	10-15 digits

Insurance Company: The insurance company table serves to store information about partnering insurance companies, including details like the company name and phone number.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
MemberID	NUMERIC	PK	N	Y	
CompanyName	STRING (35)		N	N	
CompanyEmail	STRING (254)		N	N	
GroupNumber	NUMERIC		N	N	
CompanyPhone	NUMERIC		N	N	10-15 digits

Procedure: The procedure table serves to store information about a patient’s procedure, including details about the procedure being taken like the recovery length and date. Often, dentists look back to procedure details in order to prepare corresponding surgical materials and/or charts, among other reasons.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
ProcedureID	NUMERIC	PK	N	Y	
UnitCharge	NUMERIC		N	N	
RecoveryLength	NUMERIC		N	N	
ProcedureType	STRING(20)		N	N	
ProcedureDate	DATE		N	N	

Allergy: The allergy table serves to store information about all allergies, including details about specific allergies being taken like the allergen and severity.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
AllergyID	NUMERIC	PK	N	Y	
Allergen	STRING(20)		N	N	
AllergyType	STRING(10)		N	N	“food”, “medication”, “other”
Severity	STRING(8)		N	N	“mild”, “moderate”, “severe”

Medication: The medication table serves to store information about medication, including details about specific drugs being taken like name and dosage.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
MedicationID	NUMERIC	PK	N	Y	
DrugName	STRING(20)		N	N	
MedicationType	STRING(20)		N	N	“oral”, “topical”, “injections”
Dosage	NUMERIC		N	N	Needs amount and frequency

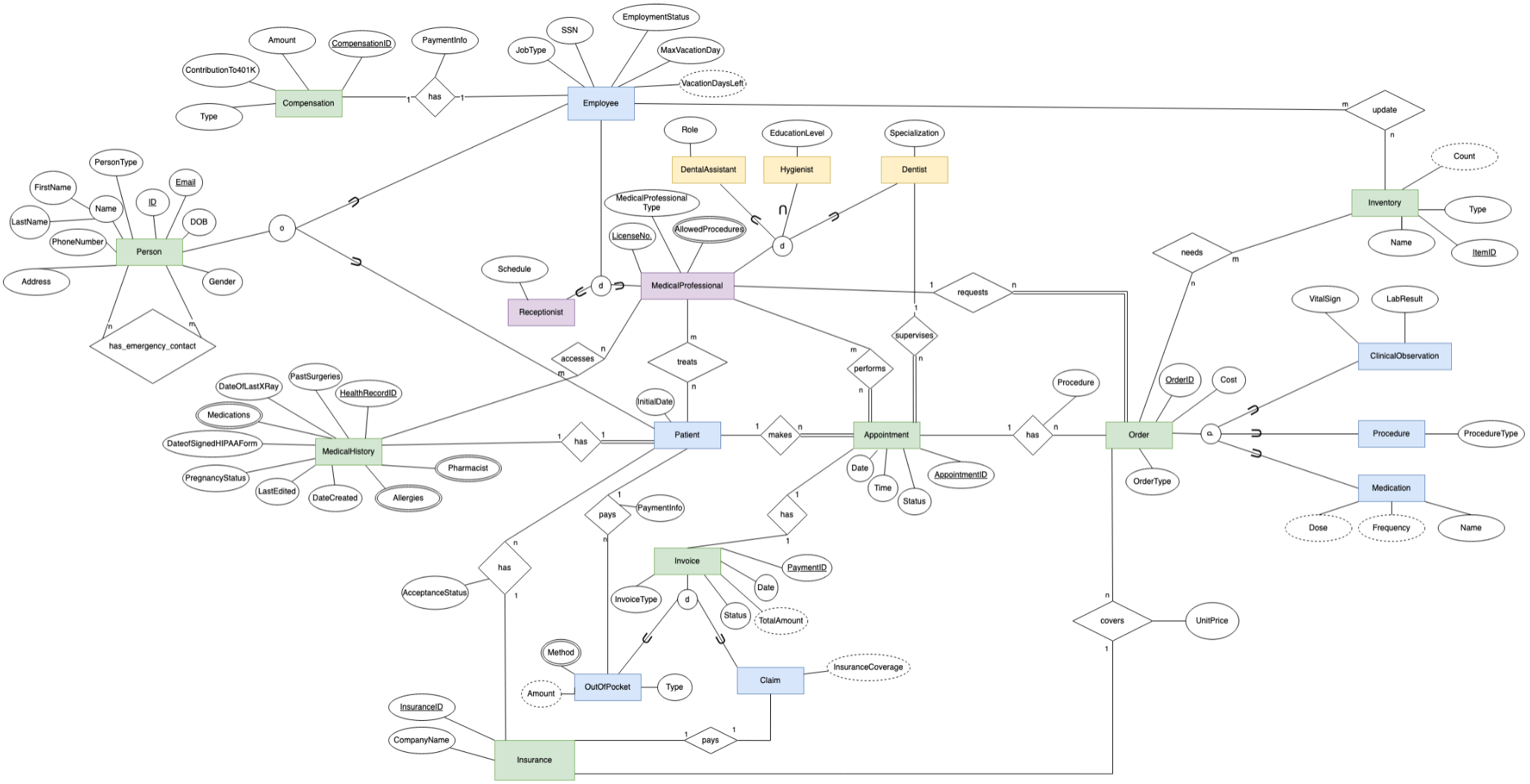
MedicalCondition: The medical condition table serves to store information about pre-existing medical conditions, which may serve as crucial information for visitations. Examples of medical conditions can look like periodontal disease or tooth decay.

Attribute Name	Data Type	Key	NULL (Y/N)	Unique (Y/N)	Other Constraints/ Notes
ConditionID	NUMERIC	PK	N	Y	
ConditionName	STRING(20)		N	N	

PART 6: Original ERD, Feedback, and Member Contributions

All members contributed to this project. Team work has been good and we regularly meet to work on the project.

Original ERD



Feedback

1. Good job on organizing and formatting your report. Going forward, ensure that you follow requirements and include all information/data in your report (not the links). Data and research links are present but data does not show evidence of significant research done and data seems to be just guessed/made up and does not follow how it would actually look for this type of business. For example, no data to properly describe procedures. Dental Procedure is the commodity your business sells (your product). This is where all the money comes from. Would be nice to properly describe what you sell. Why do we have an entity Order? Misuse of specializations. How can Procedure s(your product) Medication and Observation be specializations? Compensation entity not related to Medical Records system and would be part of HR software. Redo the ERD. It needs to correctly represent business requirements. Currently it is far from that. Check for correct use of symbols, cardinality, participation must follow the requirements and common logic. Refer to class discussions and examples. Refer to Feedback 1 announcement. Majority of listed items apply to your team. Refer to class examples including Customer-Order-Product examples. At this point, you are at less than 50% of where your ERD needs to be to be used for Part 2 steps. ERD does not pass the majority of cross-check questions.
2. Consider using Person generalization and Address entities. Make it visually clear which entity set is a generalization and which are specializations. Specializations do not have PKs. Do not overproduce specializations. Remove unnecessary specializations or add attributes to them as discussed in class. We use Specializations only if they have specific attributes or participate in a relationship that are not applicable to other instances in that entity set. Employees should be Medical (have licensure) and Non Medical. Remember to include necessary 'type' or 'flag' attributes. Remember that an Emergency contact is a mandatory data component for a medical information system and your DB must properly store that info. Allergy, Medication, Licensure are most likely going to be entities not attributes. Check for others. Attributes should be attached to the entity they describe. Why do we have AllowedProcedure in Medical professionals? A relationship would show that certain types of licences can perform certain procedures. Check that you have proper handling of insurance and other forms of payments as required in project description. Different Insurances have different rates for procedures. Consider using specializations for payment methods. They should be disjointed as each payment logically belongs to one category. You should be able to store and apply more than one payment to the same invoice. Get clear understanding about what is an Insurance Company Vs. Insurance Policy and how your DB should store this info. !!! Keep an eye on attributes that describe M:N relationships such as quantities, dates, amounts that depend on both sides. However, Procedure is not an attribute of a relationship between an Appointment and Order.
3. FYI: Sample procedure data that came up after 2 clicks in Google search: https://doh.wa.gov/sites/default/files/legacy/Documents/Pubs/410-079-DentalFeeSchedule.pdf?uid=6311634d694ba
https://www.pdffiller.com/jsfiller-desk12/?requestHash=4d3081e03f1b849d61039af18ea4eaf61c3b37eb3cd79967b3ba4bcd9d32baf1&projectId=1208791496&loader=tips&MEDIUM_PDFJS=true&PAGE_REARRANGE_V2_MVP=true&isPageRearrangeV2MVP=true#1e1d9e4c26be4d07b173d7cfcb63fb80