

Online Appointment Database

Management System

Name: Vanessa Van Dinh

ID: 917519192

Github: vandinhvanessa

Milestone/Version	Date
M2V1	11/02/21
M1V1	10/5/21

Table of Contents

Section I: Product Description and Summary.....	2
Section II: Use Cases.....	3
Section III: Database Requirements.....	11
Section IV: List of Main Entities, Attributes, and Keys.....	23
Section V: Entity Relationship Diagram (ERD).....	33
Section VI: Testing Table.....	53
Section VII: Database Model/ EER Table.....	56
Section XI: Testing Table.....	70

Section I: Product Description

The purpose of this Online Appointment Database Management System (OADMS) is to efficiently help users/patients book and manage their appointment(s) with their healthcare providers at Happy Health Medical Foundation. The database system's primary outline is for the user/patient to conveniently and easily schedule/change an appointment(s) under their insurance or other form of payment, and on the healthcare provider's availability and speciality. The OADMS is similar to other databases implemented in other medical institutions and medical foundations, such as Stanford Medical Center and Palo Alto Medical Foundation; however, many users of their implemented databases may experience frustration over the complexity of their interface and hassle of over-the-phone booking, in addition to figuring out whether their insurance covers costs or if they have to pay out of pocket. With this database implemented by Happy Health Medical Foundation, it is designed to eliminate the frustration and hassle of overbooking and paperwork, to help patients get the medical attention they need, to help healthcare providers maintain their schedules and business hours, and keep track of their upcoming appointments. Overall, the Online Appointment Database Management System provides better means to maintain, retrieve, and update data so that the patient(s) can sooner receive treatment from their healthcare provider(s), pay reduced costs, and in the long-term, increase patient satisfaction.

Section II: Use Cases

1. Use Case: Patient registers for an account

Actors: Patient, Patient Account, General User, Registered User, and Healthcare Provider

Description: Aaron is frustrated because he urgently wants to schedule an appointment with his healthcare provider for his concerning skin symptoms. And since scheduling is only over the phone or in-person and he has to provide his information everytime to the receptionist to locate his patient file, the availability of his dermatologist doctor is unclear as the receptionist always puts him on hold for long periods of time. Aaron switches to Happy Health Medical Foundation. He is a new user that needs to register for an account in order to schedule an appointment online with a dermatologist at Happy Health Medical Foundation. When Aaron is registering his account on the website, he is required to create a username and password and give his: full name, address, phone number, date of birth, medical history such as allergies or conditions, medications, and insurance or any other form of payment. If all given information is valid, then he can book an appointment through the online interface with a dermatologist and book with other providers at his convenience, he no longer has to experience the hassle and frustration of giving his information over the phone to schedule an appointment. He can clearly see the availability of a dermatologist to have his skin symptoms looked at.

The Online Appointment Database Management System is designed to help Aaron maintain and manage his patient information as he is now a registered user. He can easily schedule an appointment online with a dermatologist without the hassle of giving his patient information everytime to a receptionist.

2. Use Case: Patient updates insurance and reviews upcoming appointments

Actors: Registered User, Healthcare Provider, Appointments, Patient Account, Account Type

Description: Billy is annoyed with his healthcare provider since updating his patient information and insurance is a hassle to do over-the-phone and is not convenient to verbally speak about every single detail and the reception may input it incorrectly. Billy switches to Happy Health Medical Foundation and goes through the registration process for his account and inputs all his patient information. After a while, his insurance has expired and he needs to update it on his account and wants to review his upcoming appointment visits. In order to do so, he must first log in into his account with his username and password. If the authentication is invalid, the authentication use case ends and Billy is not allowed to continue unless his username and password is correct. If the authentication use case succeeds, then Billy is redirected to his patient center page that manages his patient information and appointments. Billy updates his insurance on his profile without hassle. Billy goes back to his patient center page to see that his next upcoming appointment is in 3 days with his healthcare provider. He no longer has to experience the hassle of verbally trying to update his insurance over the phone.

The Online Appointment Database Management System is designed to help Billy update his insurance or any other patient information for his visit without hassle and conveniently helps him see his upcoming appointments with his healthcare provider. He can easily input his insurance plan and information.

3. Use Case: Patient schedules an appointment

Actors: Registered User, Appointment, Time, Date, Healthcare Provider, Location, Clinic, Account Type, Patient, Account, Actions

Description: Calvin is a registered patient at Happy Health Medical Foundation after switching over from his former healthcare provider for unclear availability of two of his needed healthcare providers over the phone. He urgently needs to schedule two appointments back-to-back with an oncologist and a cardiologist because he is not able to go back to the clinic another time during the week. In order to do so, he must first log in into his account with his username and password. If the authentication is invalid, the authentication use case ends and Calvin cannot continue unless his username and password is correct. If the authentication use case succeeds, then Calvin is redirected to his patient center page and selects ‘Schedule Appointment’. He selects his preferred dermatologist and cardiologist (and clinic location if it is considered) and views the available time slots and clinic locations that are one after the other (next to each other). If there is no availability for back-to-back appointments at any location, then he must book the two appointments separately. If there is availability for back-to-back appointments at his selected clinic location, he selects the time slot, inputs his symptoms, reason for visit for both appointments, and must confirm the appointments. Once he confirms the appointment, Calvin can now view the appointments in his calendar page. He no longer has to experience the difficulty of trying to schedule two appointments back-to-back over the phone. He can clearly see the availability of both healthcare providers at convenience.

The Online Appointment Database Management System is designed to help Calvin schedule multiple appointments online without the frustration of trying to figure out time, date, and location with a receptionist over the phone and efficiently helps him see the availability for back-to-back time slots for any selected clinic location.

4. Use Case: Patient reschedules an existing appointment

Actors: Registered Patient, Appointment, Time, Date, Healthcare Provider, Location,

Actions

Description: Danny is a registered patient at Happy Health Medical Foundation and just scheduled an appointment with his Internal Medicine physician. Danny has just been notified of a family emergency and is distressed since he urgently needs to reschedule his appointment for another date. In order to do so, he must first log in into his account with his username and password. If the authentication is invalid, the use case ends and Danny cannot continue unless his username and password is correct. If the authentication use case succeeds, then Danny is redirected to his patient center page and selects his calendar page to view his upcoming appointments, he selects ‘Reschedule Appointment’ on the appointment he wants to reschedule, and selects the date he wants to move it to. Danny is then able to view the time slots available on that date with the same healthcare provider he selected originally. He confirms the appointment’s date and time and is able to view his rescheduled appointment in his calendar. If Danny does not want to reschedule with the same healthcare provider, then he must cancel his appointment and schedule it with his preferred healthcare provider. If Danny scheduled back-to-back appointments with multiple healthcare providers, then he must select all of the back-to-back appointments, after selecting “Reschedule Appointment”, to view the date and time slots available for back-to-back. Once he confirms the appointment(s), Danny can now see his updated upcoming appointments on his calendar. If there is no time slot for back-to-back appointments, then Danny must reschedule each appointment separately.

The Online Appointment Database Management System is designed to help Danny reschedule an appointment(s) in a time of urgency. Danny no longer has to experience the

distress of trying to reschedule over the phone in a time of urgency and figuring out the next available opening. He can, without hassle, reselect a date, time slot, and clinic that will be more suitable and convenient for him.

5. Use Case: Patient reviews after-care appointment notes from healthcare provider

Actors: Patient, Appointment, Healthcare Provider, Prescription, Medicine, Account

Description: Elon is a registered patient that just finished his appointment visit with his ENT (Ears, nose, and throat) doctor for his concerning symptoms. When Elon arrives home, he realizes he forgot the specifications on what his doctor told him about his prescription and instructions on his medication so he needs to see/review his after-care notes from his ENT doctor. In order to do so, he must first log in into his account with his username and password. If the authentication is invalid, the use case ends and Elon cannot continue unless his username and password is correct. If the authentication use case succeeds, then Elon is redirected to his patient center page and selects ‘Appointment History’. He selects the most recently finished appointment to see his after-care notes from his healthcare provider. Elon is also able to view his other after-care notes from other appointments, view his medication history, and see that his new prescription and medication from his doctor has been added. Elon now knows the specifics on his medications because the online interface easily and conveniently lets him view his after-care notes. He no longer has to doubt himself and can correctly take his medications as it is documented for him to see.

The Online Appointment Database Management System is designed and implemented to help Elon view his instructions and plans for his medication for his concerning symptoms, he can also view previous after-care notes from other past appointments and keep track of his

medications to notify his healthcare provider as some interactions of certain medications can be harmful.

6. Use Case: Registered Patient Pays with other form of payment (not Insurance)

Actors: Registered Patient, Bank Account, Payment Type, Billing, Checking, Saving

Description: Frank is a registered patient at Happy Health Medical Foundation. His insurance has just expired and Frank has not yet renewed it. Frank needs to schedule an appointment with his Oncologist, but because his insurance has not been renewed, Frank wants to input a different form of payment on his account for now. If the authentication use case succeeds, then Frank is redirected to his patient center page. He can access his patient information and select “Payment” to see what his form of payment he is currently using. He sees that his current form of payment, which is his insurance, is expired. Frank selects “Add Payment” and connects his bank account. Frank can choose to input his checking account or savings account. Once he updates his form of payment, Frank books an appointment with his Oncologist.

The Online Appointment Database Management System is designed to help notify Frank that his insurance has expired, and it is Frank’s responsibility to renew it in time or enter a different form of payment for the time being.

7. Use Case: Registered Patient Inputs Insurance and Other Form of Payment

Actors: Registered Patient, Payment Type, Insurance, Insurance Type, Private Insurance, Public Insurance, Appointment, Bank, Checking, Saving, Receptionist

Description: Greg has just registered his account at Happy Health Medical Foundation and wants to book an appointment with his primary healthcare provider. He wants to input his

insurance beforehand to avoid any hassle at the reception. If the authentication use case succeeds, then Greg is redirected to his patient center page where he can access his patient information. Greg selects “Payment” so to see all his current forms of payment, he sees that there is none yet added. He selects “Add Payment” and locates his current insurance plan from the list. Greg adds his insurance information and confirms the update. Due to the insurance, it does not fully cover the cost of the visit so there is a copayment Greg must pay. Greg must then add another form of payment on his account if he doesn’t want to experience the hassle of giving his card every time at once he’s at the clinic. He again selects “Add Payment” and adds his bank account information. Once he finishes, Greg schedules an appointment with his primary healthcare provider.

The Online Appointment Database Management System is designed to help Greg avoid giving his card in person every time he visits his primary healthcare provider, as his card is already on file to cover his copay.

8. Use Case: Sick Patient is Prescribed Medication and is given a Testing Service by Healthcare Provider

Actors: Patient, Registered User, Healthcare Provider, Appointment, Prescription, Medication, Testing Service

Description: Harry is a patient that has symptoms of night sweats, fever, and chills. He is a registered user and schedules an appointment with his Internal Medicine physician at a time and a date. After Harry visits the physician the diagnosis has been inconclusive, so the healthcare provider prescribes Harry a certain medication to help stop his fever for the time being and requests a blood testing to be done on Harry. Harry goes to the testing service at the same clinic and has his blood drawn to be tested. When Harry arrives home, he logs into his account to see

that the results have come back, and it has shown that he has a low white blood cell count indicating an infection. In his aftercare notes, his healthcare provider prescribes him the right medication for his infection and he can pick it up at his preferred pharmacy. Harry no longer has to deal with the frustration of waiting for a result back and he is able to get the right medication for his symptoms and his infection without going back in person to see his healthcare provider.

The Online Appointment Database Management System is designed to help Harry get prescribed the right medication and have his blood tested efficiently so that he could get the results in a timely manner and get the correct medication for it.

9. Use Case: Patient is given a Treatment Plan by Healthcare Provider and undergoes a Procedure

Actors: Patient, Registered User, Healthcare Provider, Treatment Plan, Procedure, Medication, Appointment

Description: Isaac is a registered user at Happy Health Medical Foundation and schedules an appointment with this Internal Medicine physician for his enlarged lymph nodes. Looking at the inconclusive results from his previous tests, his physician suggests a treatment plan to undergo a procedure to partially remove and test his swollen lymph node. Isaac agrees and is scheduled for the procedure at a time, date, and clinic. After the procedure, Isaac is able to see his results on his account. Unfortunately, the swollen lymph node has tested positive for stage 1 of cancerous cells. In his after-care notes, Isaac is urgently recommended to come back to see his physician for his revised treatment plan to help him tackle his cancer and fully remove potentially other cancerous lymph nodes.

The Online Appointment Database Management System is designed to help Isaac view his test results after his procedure and help him every step of the way by keeping him informed.

Section III: Database Requirements (Business Rules)

1. General User

- 1.1. A general user shall be able to create at most one account using a unique email
- 1.2. A general user shall have a user email
- 1.3. A general user shall have one unique user ID
- 1.4. A general user shall have a full name
- 1.5. A general user shall have a phone number
- 1.6. A general user is a registered user

2. Registered User

- 2.1. A registered user is a general user
- 2.2. A registered user shall create sessions on zero or more devices

3. Account

- 3.1. An account shall belong to one and only one user
- 3.2. An account shall have one unique account ID
- 3.3. An account shall have at least one form of payment recorded
- 3.4. An account shall be linked to either a bank account or an insurance plan
- 3.5. An account shall have one and only one profile
- 3.6. An account shall have one encrypted password
- 3.7. An account shall have one unique email
- 3.8. An account shall have one creation date
- 3.9. An account is either an employee account or a patient account

4. Employee account

- 4.1. An employee account shall have a unique employee account id

- 4.2. An employee account shall have a position type
- 4.3. An employee account shall have a description
- 4.4. An employee account shall belong to one and only one employee

5. Patient Account

- 5.1. A patient account shall have a patient account id
- 5.2. A patient account shall have a description
- 5.3. A patient account shall have belong to one and only patient
- 5.4. A patient account shall have zero or more prescription recorded
- 5.5. A patient account shall have zero or more treatment plans recorded
- 5.6. A patient account shall have zero or more visit history recorded
- 5.7. A patient account shall schedule zero or more appointments with a healthcare provider

6. Patient

- 6.1. A patient shall visit zero or more clinics
- 6.2. A patient shall have one unique patient ID
- 6.3. A patient shall visit one and only one healthcare provider at an appointment
- 6.4. A patient shall pay zero or more bills
- 6.5. A patient shall pay with either a bank account or insurance plan
- 6.6. A patient shall have at most one account
- 6.7. A patient shall have take at least one medicine
- 6.8. A patient shall have receive at least one procedure
- 6.9. A patient shall receive at least one testing service
- 6.10. A patient shall have zero or more scheduled appointments

6.11. A patient shall have at least one address

7. Appointment

- 7.1. An appointment shall belong to one patient
- 7.2. An appointment shall be scheduled at one time
- 7.3. An appointment shall have scheduled at one date
- 7.4. An appointment shall be scheduled with one healthcare provider
- 7.5. An appointment shall be assigned with at least one nurse
- 7.6. An appointment shall have one payment type
- 7.7. An appointment shall be at one clinic
- 7.8. An appointment shall issue one bill
- 7.9. An appointment shall have aftercare notes

8. Payment Types

- 8.1. A payment type shall be either a bank account or insurance plan
- 8.2. A payment type shall be saved in one account
- 8.3. A payment type shall have one unique payment type id
- 8.4. A payment type shall have one billing address
- 8.5. A payment type shall have one zip code
- 8.6. A payment type shall have one country

9. Billing

- 9.1. A bill shall be issued to one patient
- 9.2. A bill shall have a time and date
- 9.3. A bill shall have a unique billing id
- 9.4. A bill shall have a list of charges

10. Insurance Plan

- 10.1. An insurance plan shall cover for zero or more patients
- 10.2. An insurance plan shall be saved in at most one account
- 10.3. An insurance plan shall have one unique insurance plan type id
- 10.4. An insurance plan shall have one expiration date
- 10.5. An insurance plan shall have one insurance type
- 10.6. An insurance plan shall have a issue date
- 10.7. An insurance plan shall have a expire date

11. Bank Account

- 11.1. A bank account shall pay zero or more bills
- 11.2. A bank account shall pay for zero or more patients
- 11.3. A bank account shall pay for zero or more prescriptions
- 11.4. A bank account shall pay for zero or more treatments
- 11.5. A bank account shall have one unique account number
- 11.6. A bank account shall have one bank code
- 11.7. A bank account shall have routing number
- 11.8. A bank account shall belong to one payment type

12. Saving

- 12.1. A saving account is a bank account
- 12.2. A saving account shall have one unique account number
- 12.3. A saving account shall have a routing number

13. Checking

- 13.1. A checking account is a bank account

- 13.2. A checking account shall have one unique account number
- 13.3. A checking account shall have a routing number

14. Healthcare Provider

- 14.1. A healthcare provider shall have zero or more patients
- 14.2. A healthcare provider shall meet with one patient at an appointment
- 14.3. A healthcare provider is assigned to zero or more appointments
- 14.4. A healthcare provider shall work for one department
- 14.5. A healthcare provider is a employee
- 14.6. A healthcare provider works with at least one nurse
- 14.7. A healthcare provider is a user
- 14.8. A healthcare provider shall have zero or more appointments
- 14.9. A healthcare provider shall work at least one clinic
- 14.10. A healthcare provider shall prescribe at least one medicine to a patient
- 14.11. A healthcare provider shall give at least one procedure to a patient
- 14.12. A healthcare provider shall work for one company (HHMF)
- 14.13. A healthcare provider shall have one salary
- 14.14. A healthcare provider shall have a unique healthcare provider id
- 14.15. A healthcare provider shall give at least one testing service to a patient
- 14.16. A healthcare provider shall have a specialization
- 14.17. A healthcare provider shall have a description

15. Employee

- 15.1. An employee shall work for one or more company
- 15.2. An employee shall have a unique employee ID

- 15.3. An employee shall have one salary
- 15.4. An employee is a user
- 15.5. An employee shall have a start date
- 15.6. An employee shall have a position
- 15.7. An employee shall have an email
- 15.8. An employee shall have a salary
- 15.9. An employee shall have a date of birth
- 15.10. An employee shall have a phone number
- 15.11. An employee shall have an one address

16. Department

- 16.1. A department shall have one or more employees
- 16.2. A department shall have a unique department id
- 16.3. A department shall have be under one company (HHMF)
- 16.4. A department shall have a unique email
- 16.5. A department shall have a unique phone number
- 16.6. A department shall have a unique face number

17. Nurses

- 17.1. A nurse shall work with at least one healthcare provider
- 17.2. A nurse shall work in one department
- 17.3. A nurse is an employee
- 17.4. A nurse is a user
- 17.5. A nurse shall work at one clinic
- 17.6. A nurse shall have one salary

17.7. A nurse shall have a unique nurse id

17.8. A nurse shall have a description

18. Calendar Date

18.1. A date shall have zero or more appointments

18.2. A date shall have an availability and booked flag

18.3. A date shall have one day

18.4. A date shall have one month

18.5. A date shall have one year

19. Calendar Time

19.1. A time slot shall have zero or more appointments

19.2. A time slot shall have a availability and booked flag

19.3. A time slot shall have an hour and minute

19.4. A time slot shall be either in AM or PM

20. Receptionist

20.1. A receptionist shall work for at least one clinic

20.2. A receptionist shall work for one department

20.3. A receptionist is an employee

20.4. A receptionist shall have a unique receptionist id

20.5. A receptionist shall have a description

21. Clinic

21.1. A clinic shall have at least one location

21.2. A clinic shall have at least one healthcare providers

21.3. A clinic shall have at least one nurses

- 21.4. A clinic shall have at least one patients
- 21.5. A clinic shall have at least one receptionists
- 21.6. A clinic shall be owned by one company (HHMF)
- 21.7. A clinic shall have a unique number
- 21.8. A clinic shall have a unique fax number

22. Location

- 22.1. A location shall have at most one clinic
- 22.2. A location shall be in one city
- 22.3. A location shall be in one zip code
- 22.4. A location shall in one country
- 22.5. A location shall have a unique address

23. Prescription

- 23.1. A prescription shall be for one or more patients
- 23.2. A prescription shall be given by one healthcare provider
- 23.3. A prescription shall be paid by either a bank account or insurance plan
- 23.4. A prescription shall have a unique prescription id
- 23.5. A prescription shall be signed off by a healthcare provider
- 23.6. A prescription shall have a description

24. Treatment Plan

- 24.1. A treatment shall be given by one healthcare provider
- 24.2. A treatment shall be given to zero or more patient
- 24.3. A treatment shall be paid by either a bank account or insurance plan
- 24.4. A treatment plan shall have a treatment plan id

- 24.5. A treatment plan shall be signed off by a healthcare provider
- 24.6. A treatment plan shall have a description

25. Profile

- 25.1. A profile shall have one unique profile ID
- 25.2. A profile shall be owned by one and only one account
- 25.3. A profile shall have one profile picture
- 25.4. A profile shall have zero or more pronouns
- 25.5. A profile shall create many actions

26. Medicine

- 26.1. A medicine shall have a unique medicine id
- 26.2. A medicine shall have a name
- 26.3. A medicine shall have a manufacturer date
- 26.4. A medicine shall have an expiration date
- 26.5. A medicine shall be made by zero or more pharmaceutical
- 26.6. A medicine shall have a price
- 26.7. A medicine shall be taken by zero or more patients
- 26.8. A medicine shall be in zero or more prescriptions
- 26.9. A medicine shall be prescribed by zero or more healthcare providers

27. Device

- 27.1. A device shall hold session with zero or more registered users
- 27.2. A device shall have an ip address
- 27.3. A device shall have zero or more sessions
- 27.4. A device shall have a unique device id

27.5. A device shall log into at most one account

28. Procedure

28.1. A procedure shall be received by zero or more patients

28.2. A procedure shall be in zero or more treatment plans

28.3. A procedure shall be given by zero or more healthcare providers

28.4. A procedure shall have a unique procedure id

28.5. A procedure shall have a time and a date

28.6. A procedure shall be performed in a clinic

28.7. A procedure shall have after-care notes

28.8. A procedure shall have a description

29. Patient Address

29.1. A patient address shall belong to zero or more patients

29.2. A patient address shall have a street address

29.3. A patient address shall have a city

29.4. A patient address shall have a zip code

29.5. A patient address shall have a country

29.6. A patient address shall have a

30. Action

30.1. An action shall have an action id

30.2. An action shall be created by one profile

30.3. An action shall have a type

30.4. An action shall have a description

- 30.5. An action shall be permissible by an employee account type and a patient account type

31. Test Services

- 31.1. A testing service shall have a unique testing service id
- 31.2. A testing service shall have a name
- 31.3. A testing service shall have a time and a date
- 31.4. A testing service shall have report date
- 31.5. A testing service shall have a report description
- 31.6. A testing service is given by zero or more healthcare providers
- 31.7. A testing service can be taken by zero or more patients
- 31.8. A testing service shall have a serial number for the patient

32. Insurance Company

- 32.1. An insurance company shall have a unique insurance company id
- 32.2. An insurance company shall have a name
- 32.3. An insurance type shall have a policy
- 32.4. An insurance type shall have a deductible
- 32.5. An insurance type shall have a premium

33. Pharmaceutical Company

- 33.1. A pharmaceutical company shall have a name
- 33.2. A pharmaceutical company shall have a head quarter address
- 33.3. A pharmaceutical company shall have an email
- 33.4. A pharmaceutical company shall have a phone number
- 33.5. A pharmaceutical company shall have net worth

- 33.6. A pharmaceutical company shall have a list of sold drugs
- 33.7. A pharmaceutical company shall manufacture zero or more drugs
- 33.8. A pharmaceutical company shall contract with zero or more pharmacies
- 33.9. A pharmaceutical company shall have a unique pharmaceutical id

34. Pharmacy

- 34.1. A pharmacy shall have a name
- 34.2. A pharmacy shall have a pharmacy id
- 34.3. A pharmacy shall have a address
- 34.4. A pharmacy shall have a phone number
- 34.5. A pharmacy shall contract with zero or more pharmaceutical companies
- 34.6. A pharmacy shall sell zero or more drugs to patients
- 34.7. A pharmacy shall have an email

35. Employee Address

- 35.1. An employee address shall have a unique employee address id
- 35.2. An employee address shall have an address
- 35.3. An employee address shall have a city
- 35.4. An employee address shall have a state
- 35.5. An employee address shall have a zip code
- 35.6. An employee address shall have a country
- 35.7. An employee address shall belong to zero or more employees

Section IV: List of Entities and Attributes

1. User (Strong)

- a. user_id: key, numeric
- b. user_email: key, alphanumeric
- c. user_name: multivalue, alphanumeric
- d. user_last_name: multivalue, alphanumeric
- e. user_phone_number: key, numeric
- f. user_address: multivalue, alphanumeric
- g. user_dob: multivalue, timestamp

2. Account (Weak)

- a. account_id: key, numeric
- b. account_email: key, alphanumeric
- c. account_password: alphanumeric
- d. account_creation: composite, multivalue, date

3. Profile (Weak)

- a. profile_id: key, numeric
- b. profile_picture: multivalue, image
- c. profile_pronouns: alphanumeric

4. Patient (Weak)

- a. patient_id: key, numeric
- b. medication_history: composite, multivalue, alphanumeric
- c. treatment_history: composite, multivalue, alphanumeric
- d. medical_history: composite, multivalue, alphanumeric

- e. patient_dob: multivalue, timestamp
- f. patient_phone_number: numeric
- g. patient_email: key, alphanumeric
- h. patient_name: multivalue, alphanumeric

5. Appointment (Weak)

- a. hcp_id: key, numeric
- b. patient_id: key, numeric
- c. patient_name: multivalue, alphanumeric
- d. user_id: key, numeric
- e. symptoms: alphanumeric
- f. payment_type_id: key, numeric
- g. appt_id: key, numeric
- h. appt_date: composite, date
- i. appt_time: composite, time
- j. appt_confirmation: boolean flag
- k. appt_clinic: composite, alphanumeric
- l. after_care_notes: alphanumeric

6. Payment Type (Strong)

- a. payment_type_id: key, numeric
- b. billing_address: composite, alphanumeric
- c. billing_zipcode: numeric
- d. billing_city: alphanumeric
- e. billing_state: alphanumeric

f. billing_country: alphanumeric

7. Bank Account (Weak)

- a. account_number: key, numeric
- b. bank_code: alphanumeric
- c. routing_number: numeric
- d. bank_type: alphanumeric

8. Savings (Weak)

- a. saving_account_number: key, numeric
- b. saving_routing_number: numeric
- c. saving_type: alphanumeric

9. Checking (Weak)

- a. checking_account_number: key, numeric
- b. checking_routing_number: numeric
- c. checking_type: alphanumeric

10. Clinic (Weak)

- a. clinic_name: alphanumeric
- b. clinic_location: composite, alphanumeric
- c. clinic_fax_number: numeric
- d. clinic_email: key, alphanumeric
- e. clinic_phone_number: numeric

11. Location (Strong)

- a. location_address: composite, alphanumeric
- b. location_city: alphanumeric

- c. location_zipcode: numeric
- d. location_country: alphanumeric

12. Insurance Plan(Weak)

- a. insurance_id: key, numeric
- b. insurance_plan_type: alphanumeric
- c. insurance_expire_date: composite, date
- d. insurance_issue_date: composite, date

13. Department (Strong)

- a. department_name: multivalue, alphanumeric
- b. department_id: key, numeric
- c. department_email: key, alphanumeric
- d. department_fax_number: numeric
- e. department_phone_number: numeric
- f. department_floor: numeric
- g. department_clinic: alphanumeric

14. Employee (Weak)

- a. employee_id: key, numeric
- b. employee_start_date: composite, date
- c. employee_salary: numeric
- d. employee_position: composite, alphanumeric
- e. employee_dob: multivalue, timestamp
- f. employee_number: numeric
- g. employee_email: key, alphanumeric

- h. employee_address: composite, alphanumeric
- i. user_id: key, numeric

15. Healthcare Provider (Weak)

- a. hcp_specialization_pos: composite, alphanumeric
- b. hcp_department: alphanumeric
- c. hcp_id: key, numeric
- d. hcp_description: alphanumeric

16. Nurse (Weak)

- a. nurse_id: key, numeric
- b. Nurse_description: alphanumeric
- c. nurse_position: alphanumeric
- d. nurse_department: alphanumeric

17. Receptionist (Weak)

- a. reception_id: key, numeric
- b. reception_description: alphanumeric
- c. reception_department: key, numeric

18. Prescription (Weak)

- a. prescription_id: key, numeric
- b. prescription_hcp: composite, alphanumeric
- c. prescription_sign_off: boolean flag
- d. prescription_name: multivalue, alphanumeric
- e. prescription_description: alphanumeric
- f. prescription_receiver: multivalue, alphanumeric

19. Medicine (Strong)

- a. medicine_id: key, numeric
- b. medicine_name: alphanumeric
- c. medicine_quantity: numeric
- d. medicine_date: composite, date
- e. expire_date: composite, date
- f. medicine_company: alphanumeric
- g. medicine_price: alphanumeric

20. Treatment Plan (Weak)

- a. treatment_id: key, numeric
- b. treatment_hcp: multivalue, alphanumeric
- c. treatment_sign_off: boolean flag
- d. treatment_patient: multivalue, alphanumeric
- e. Treatment_description: alphanumeric

21. Procedure Operation(Strong)

- a. procedure_id: key, numeric
- b. procedure_patient: key, numeric
- c. procedure_hcp: key, numeric
- d. procedure_time: composite, alphanumeric
- e. procedure_date: composite, date
- f. procedure_description: alphanumeric
- g. procedure_after_care: alphanumeric
- h. procedure_clinic: alphanumeric

22. Calendar Time (Strong)

- a. time_am_pm: alphanumeric
- b. time_available: boolean flag
- c. time_hour: numeric
- d. time_booked: boolean flag

23. Calendar Date (Strong)

- a. date_day: alphanumeric
- b. date_month: alphanumeric
- c. date_year: numeric
- d. date_available: boolean flag
- e. date_booked: boolean flag

24. Billing (Weak)

- a. billing_id: key, numeric
- b. patient_id: key, numeric
- c. description: composite, alphanumeric
- d. time: alphanumeric
- e. date: alphanumeric

25. Devices (Strong)

- a. device_id: key, numeric
- b. device_ip: alphanumeric
- c. device_type: alphanumeric

26. Sessions (Weak)

- a. session_id: key, numeric

- b. user_id: key, numeric
- c. session_expire: composite, date

27. Employee Account (Weak)

- a. employee_account_id: key, numeric
- b. employee_account_type: alphanumeric
- c. employee_description: alphanumeric

28. Patient Account (Weak)

- a. patient_account_id: key, numeric
- b. patient_account_description: alphanumeric
- c. patient_id: key, numeric
- d. appointment_history: key, composite, alphanumeric

29. Patient Address (Strong)

- a. patient_street_address: alphanumeric
- b. patient_city: alphanumeric
- c. patient_zipcode: numeric
- d. patient_country: alphanumeric
- e. patient_id: key, numeric

30. Actions (Strong)

- a. action_id: key, numeric
- b. action_type: key, alphanumeric
- c. action_descriprion: alphanumeric

31. Insurance Company (Strong)

- a. insurance_company_id : key, numeric

- b. insurance_company_name: alphanumeric
- c. insurance_company_policy: alphanumeric
- d. insurance_company_deductible: alphanumeric
- e. insurance_company_premium: alphanumeric

32. Pharmaceutical Company (Strong)

- a. pharmaceutical_id: key, numeric
- b. pharmaceutical_name: alphanumeric
- c. pharmaceutical_netsworth: numeric
- d. pharmaceutical_address: composite, alphanumeric
- e. pharmaceutical_phone_number: numeric
- f. pharmaceutical_email: alphanumeric

33. Pharmacy (Strong)

- a. pharmacy_id: key, numeric
- b. Pharmacy_name: alphanumeric
- c. pharmacy_address: composite, alphanumeric
- d. pharmacy_phone_number: numeric
- e. pharmacy_email: alphanumeric

34. Testing Service (Strong)

- a. testing_service_id: key, numeric
- b. testing_service_name: key, alphanumeric
- c. testing_time: composite, alphanumeric
- d. testing_date: composite, date
- e. testing_report_description: alphanumeric

35. Permissions (Weak)

- a. permission_id: key, numeric
- b. account_type_id: key, numeric
- c. permission_flag: flag, boolean

36. Contract (Weak)

- a. contract_start_date: composite, date
- b. contract_end_date: composite, date
- c. contract_description: alphanumeric

37. Selling/Sale (Weak)

- a. sale_quantity: numeric
- b. sale_product: alphanumeric
- c. sale_product_id: key, numeric

38. Manufacture (Weak)

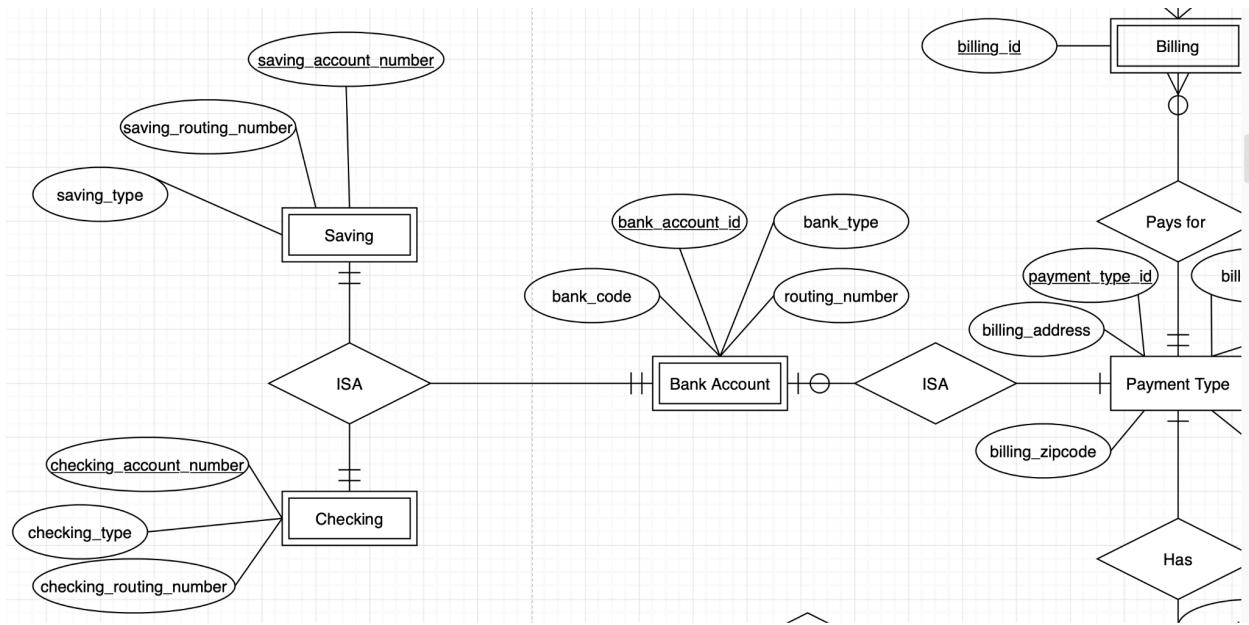
- a. manufacture_date: composite, date
- b. manufacture_id: key, numeric
- c. manufacture_location: composite, alphanumeric

39. Employee Address (Strong)

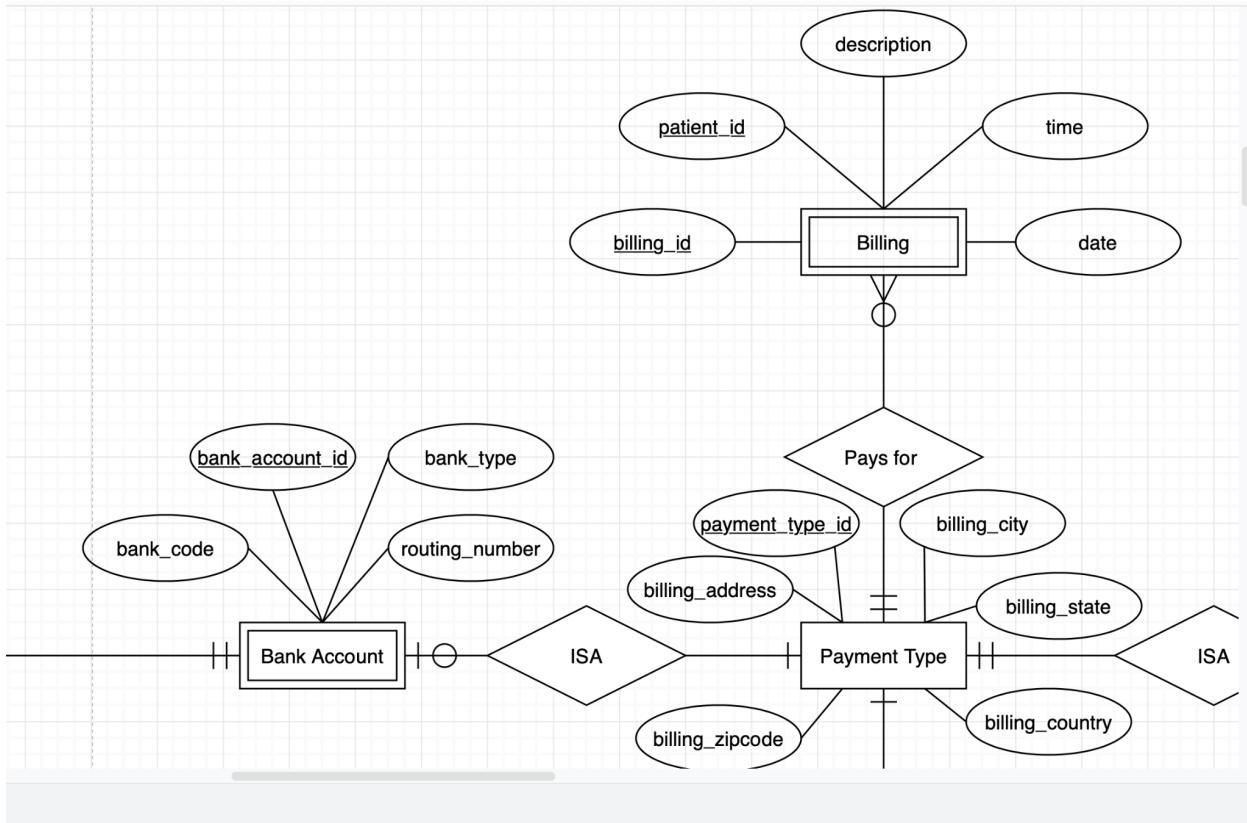
- a. employee_add_id: key, numeric
- b. employee_address: alphanumeric
- c. employee_city: alphanumeric
- d. employee_state: alphanumeric
- e. employee_zipcode: alphanumeric
- f. employee_country: alphanumeric

Section V: Entity Relationship Diagram

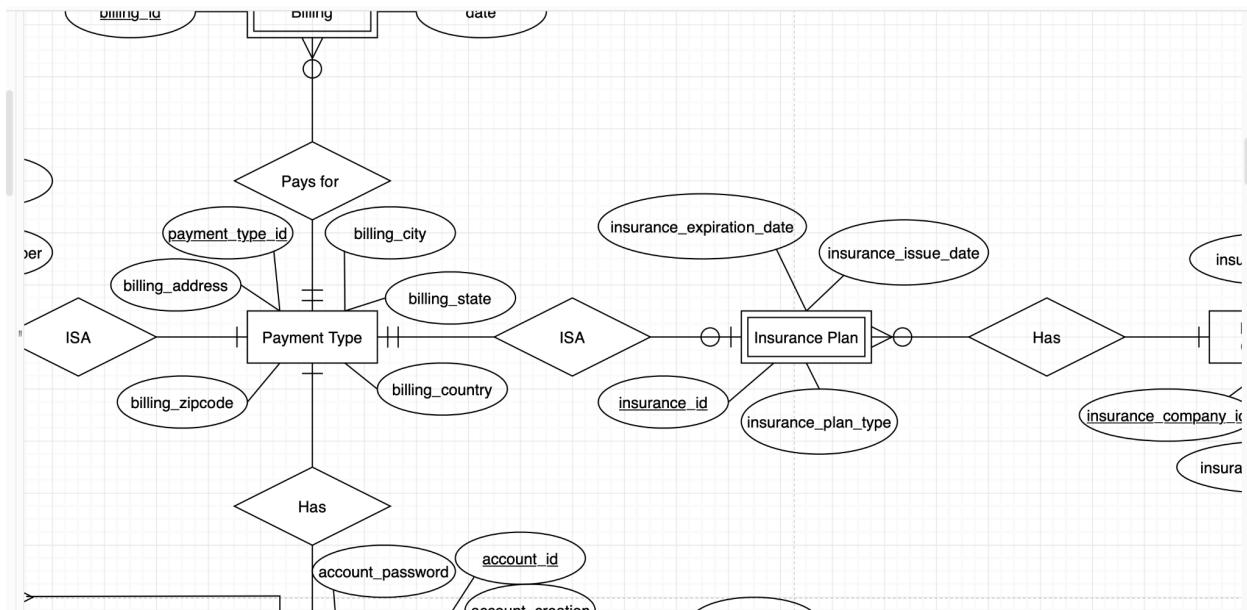
- ERD starts from left to right. The screenshot below shows Savings, Checking, Bank Account to Payment Type.



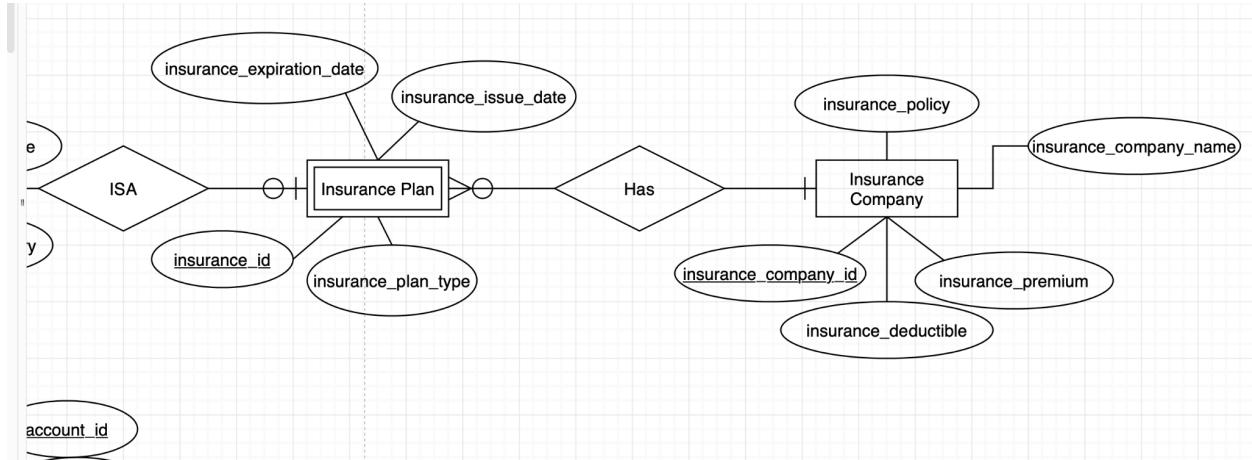
- The screenshot below shows the Bank Account to Payment Type to Billing



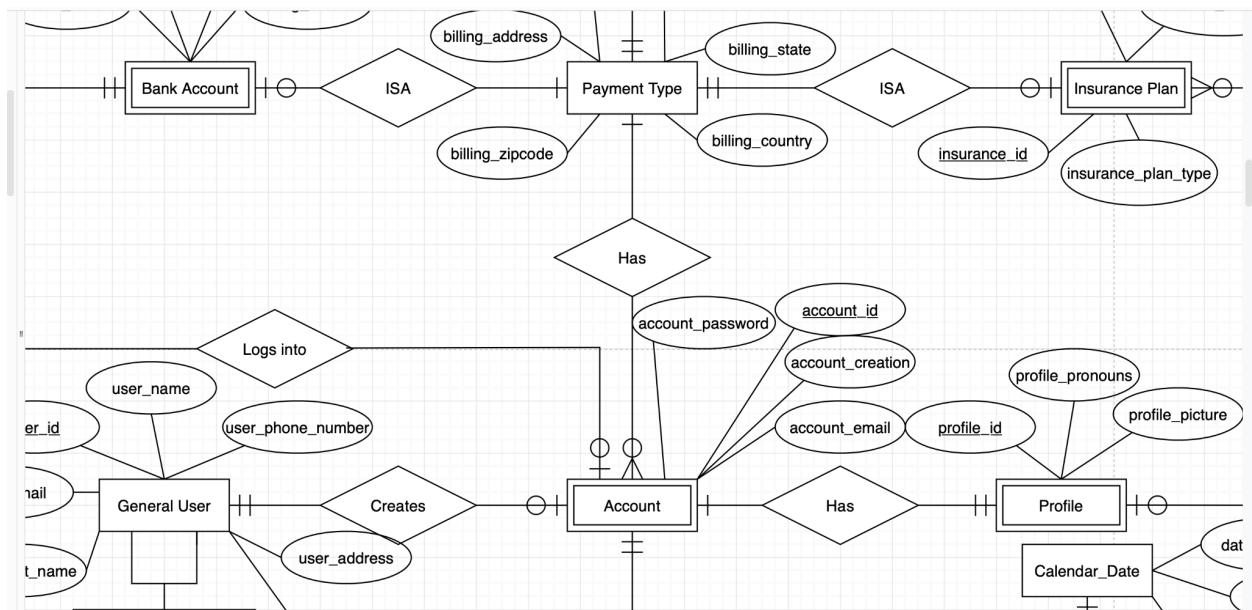
- The screenshot below shows the Payment Type and Insurance Plan relationship



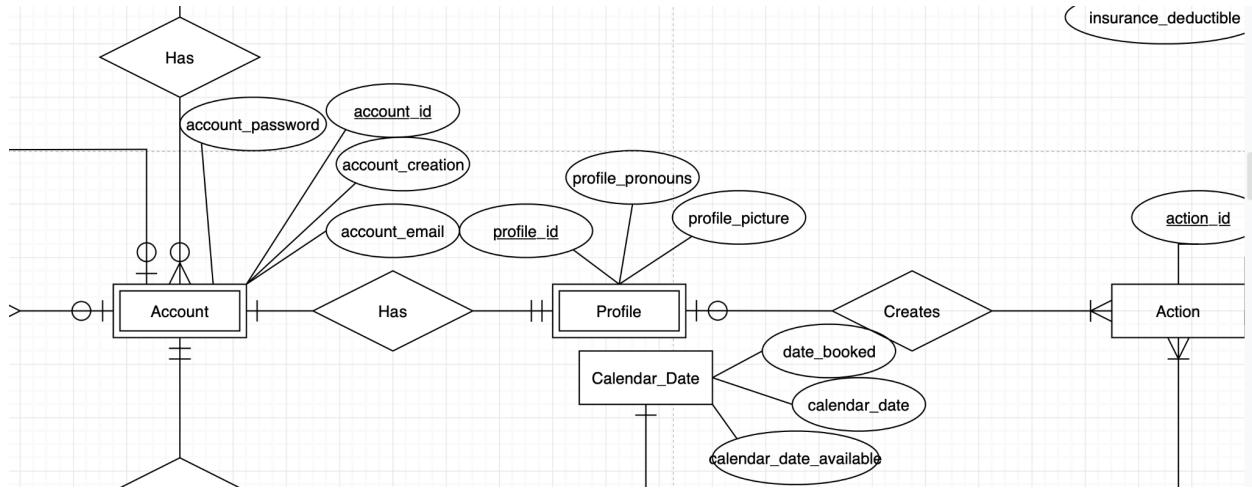
- The screenshot below shows the Insurance Plan to Insurance Company



- The screenshot below shows from Payment Type, it goes to Account. The next screenshot will go right of the Account entity.

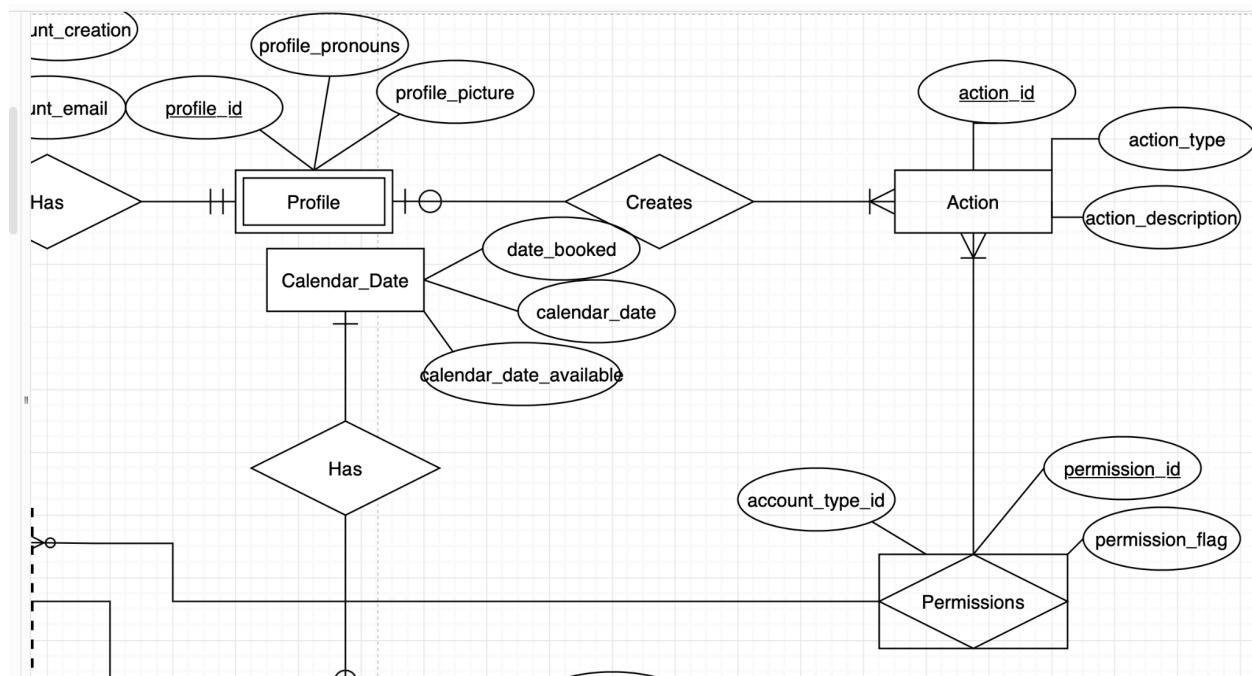


- This screenshot below shows the right side of the ERD, from Account to Profile and to Action

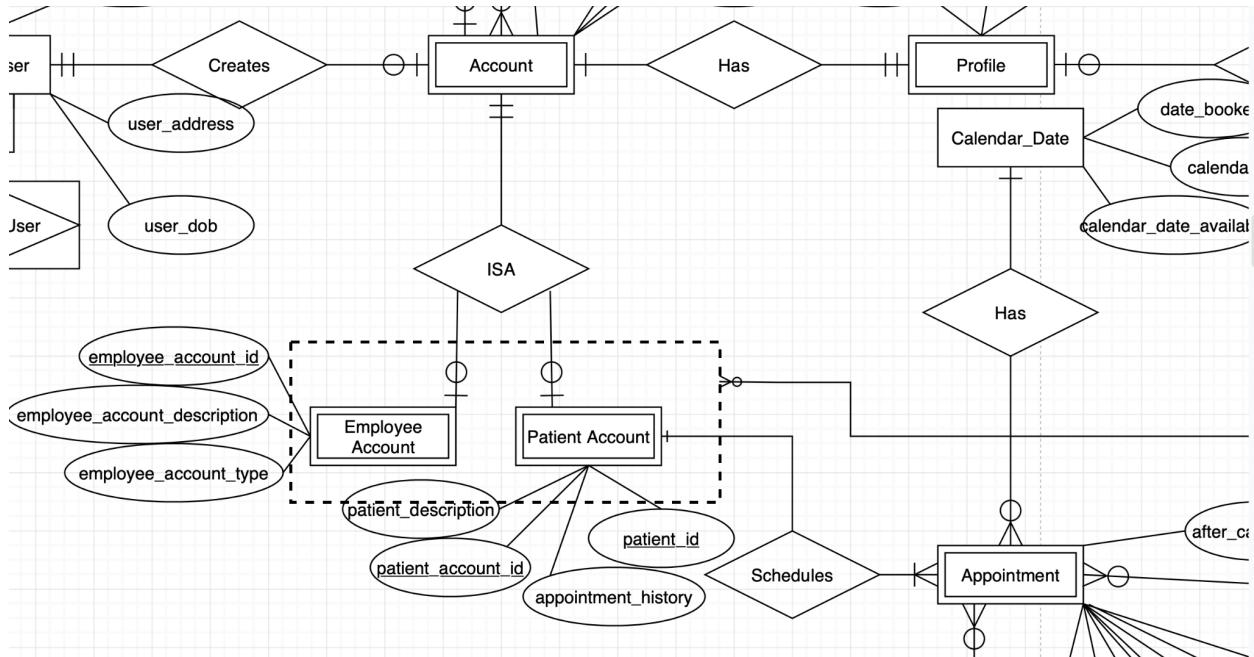


- The screenshot continues right of Profile, it shows Profile to Action to Permissions.

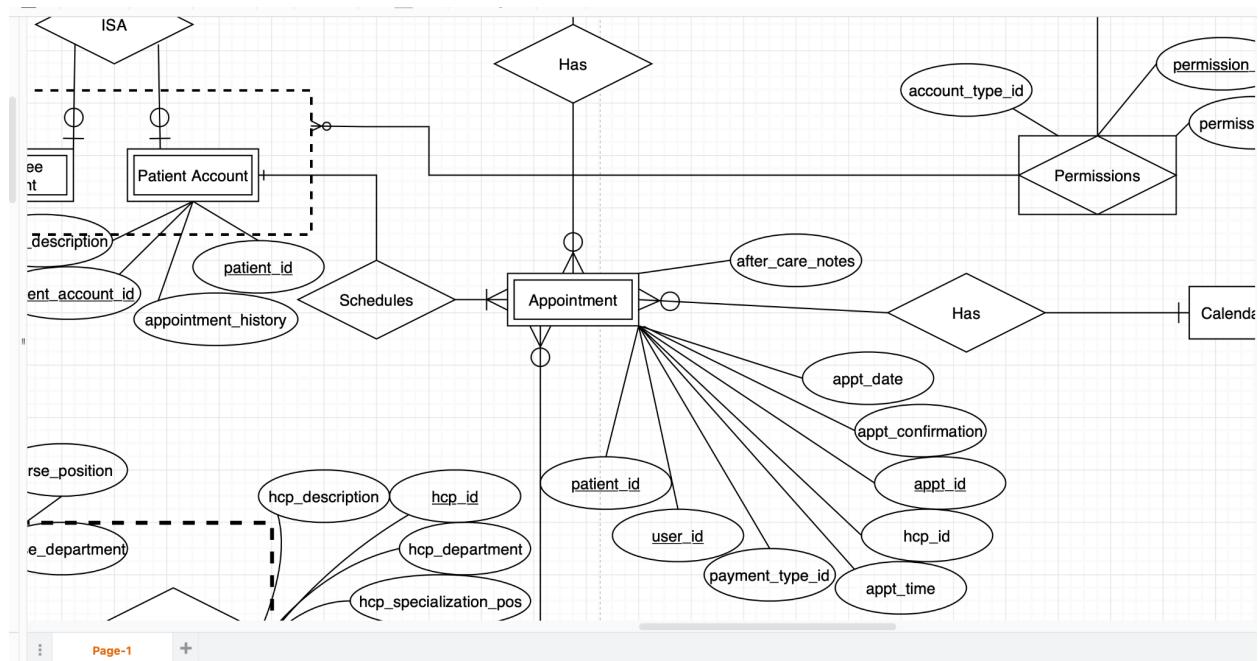
Permission is also connected to the Employee Account and Patient Account aggregation.



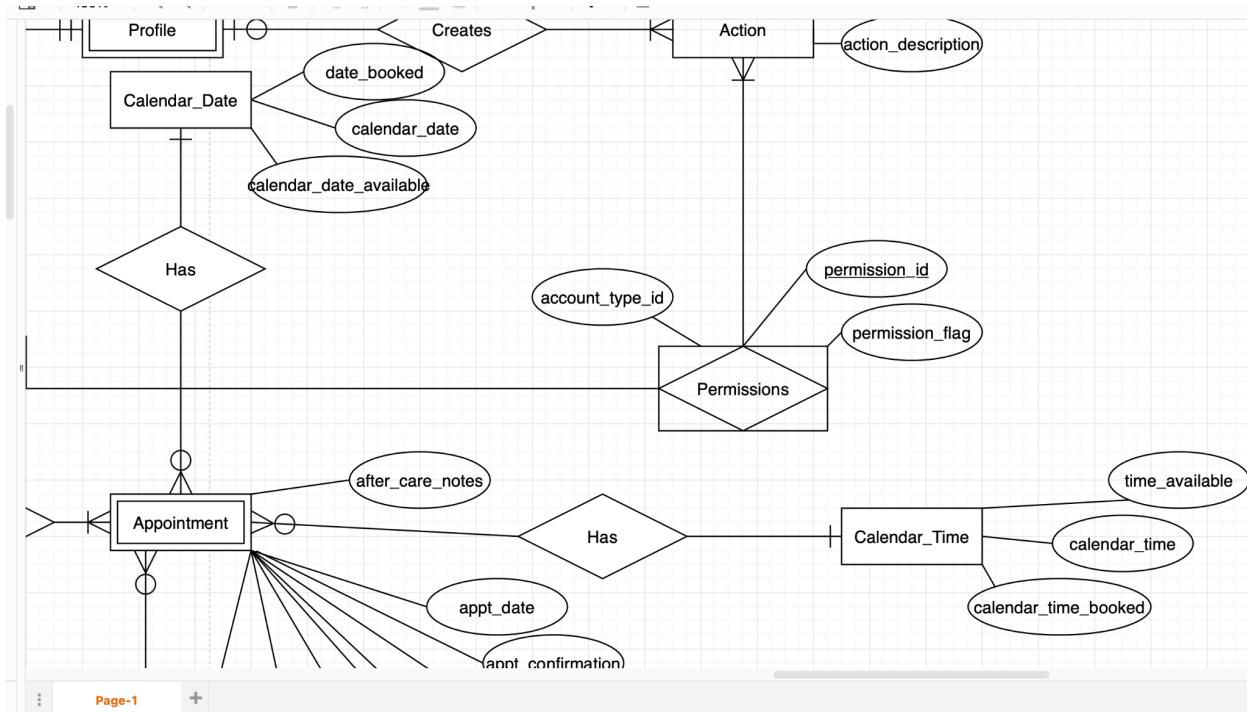
- The screenshot goes left of Permissions to show its relation to the Employee Account and Patient Account aggregation.



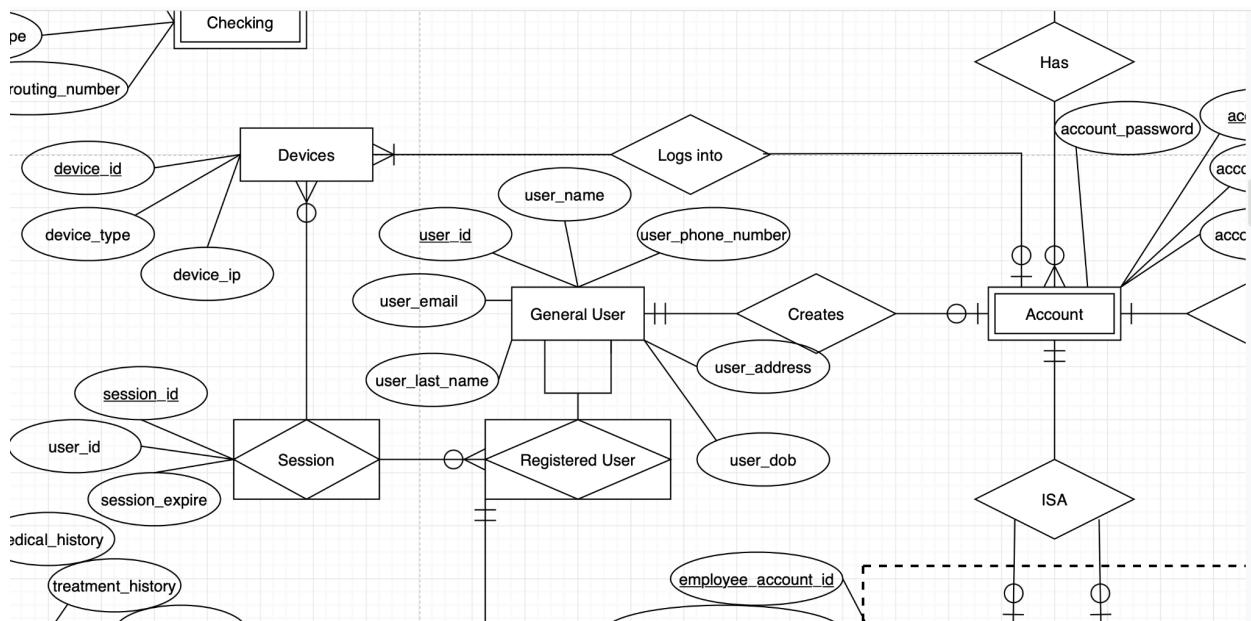
- The screenshot below shows the relationship with Patient Account and Appointment to Calendar Date and Calendar Time.



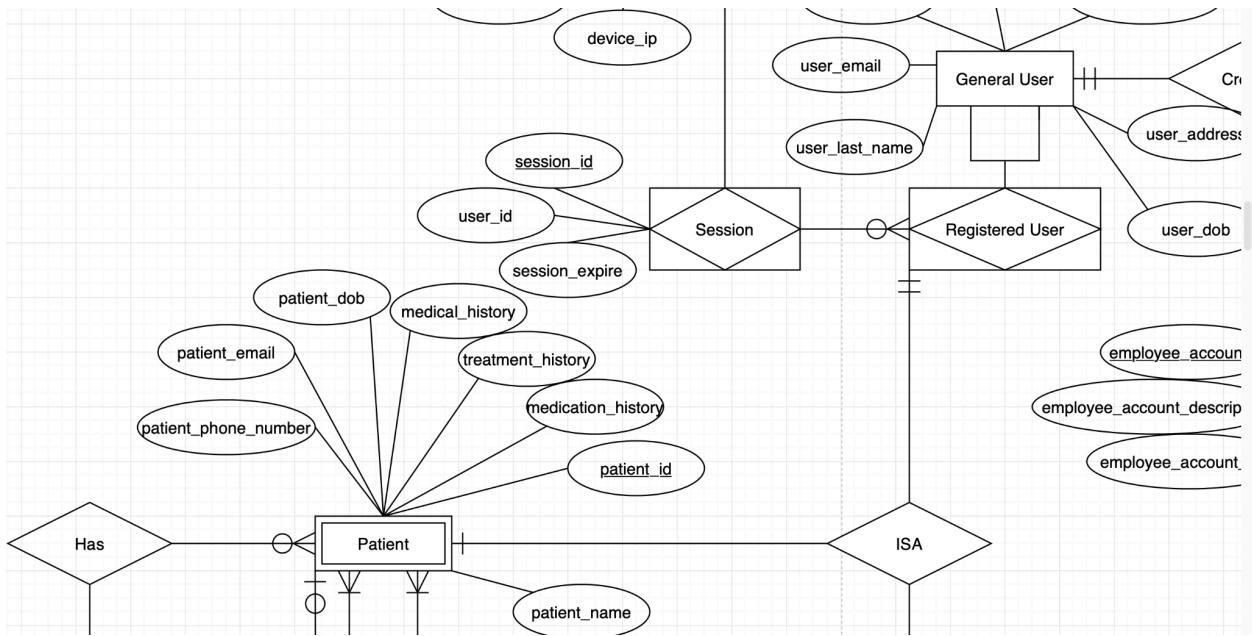
- The screenshot below shows the Appointment relationship to Calendar Date and Calendar Time.



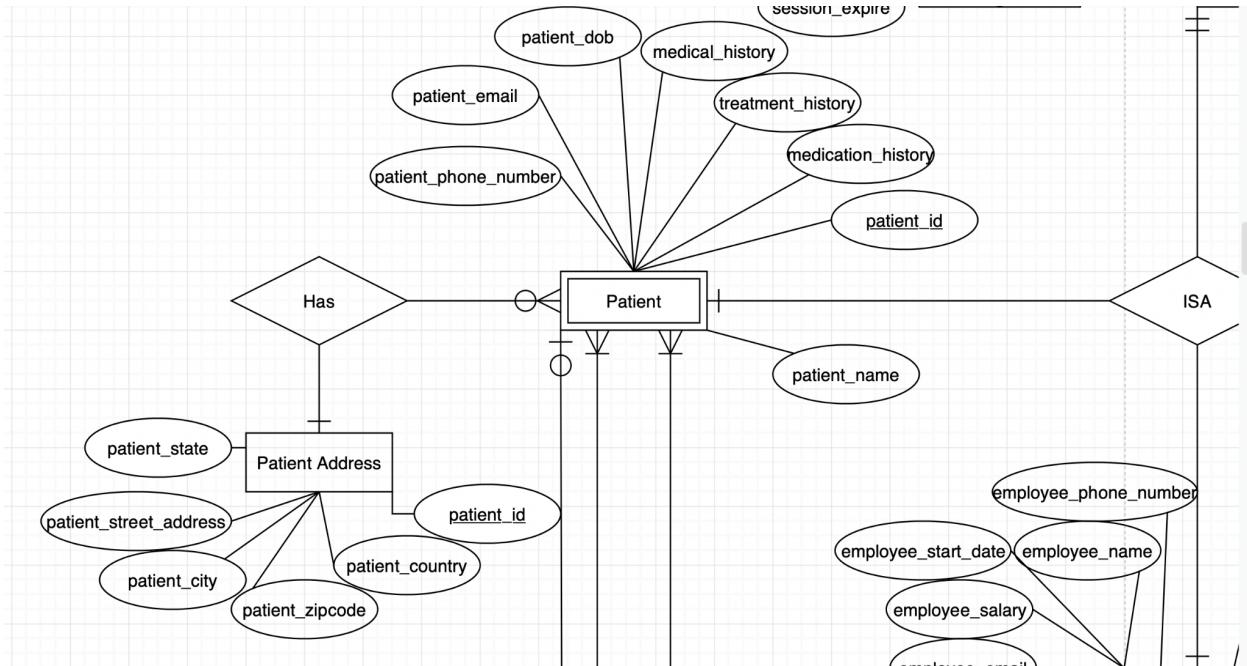
- The screenshot below continues to the left of the Account entity. It shows its relationship to General User and Registered User, as well as Devices and Sessions to Registered User.



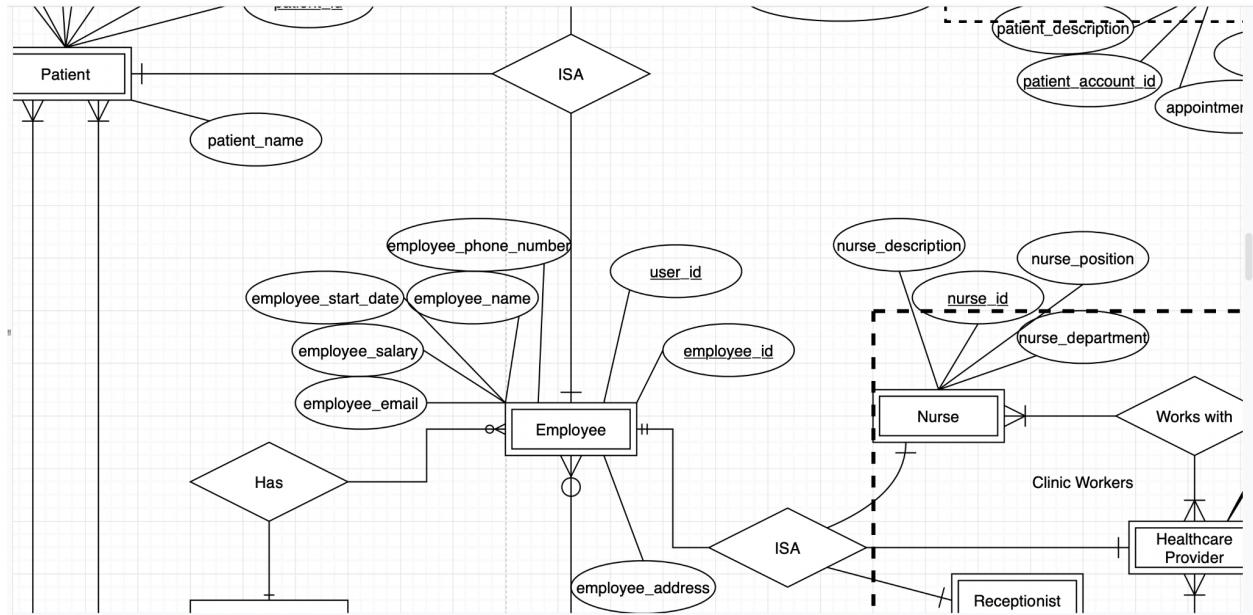
- The screenshot below continues south of the ERD, it shows the ISA relationship to Patient and Employee.



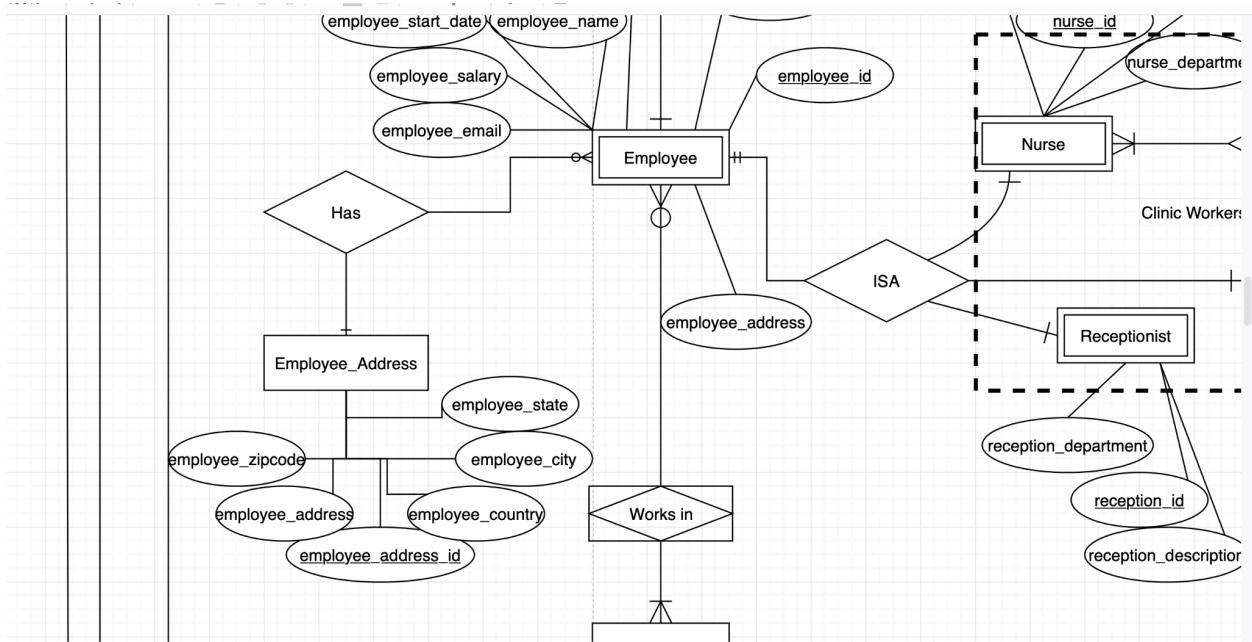
- The screenshot below shows the Patient relationship to Patient Address. Notice the three cardinalities at the bottom of the Patient Entity.



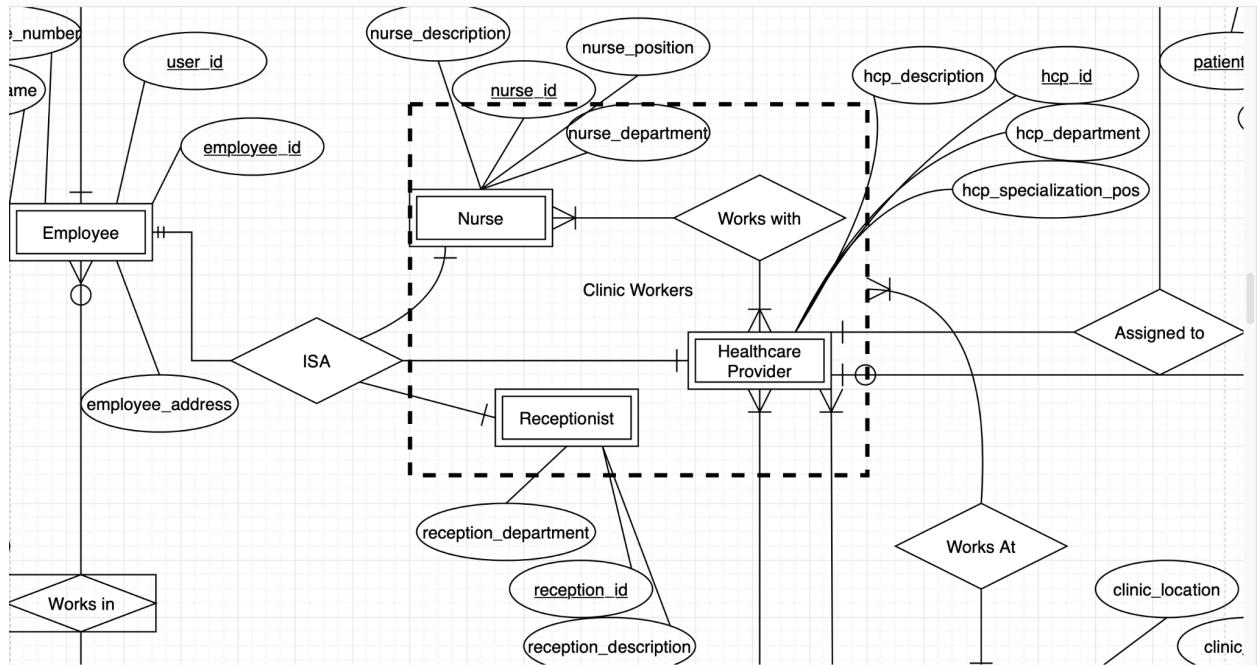
- The screenshot below shows the Registered/General User relationship to Patient AND Employee. Also shows the Employee's relationship to Nurse, Healthcare Provider, and Receptionist to the right.



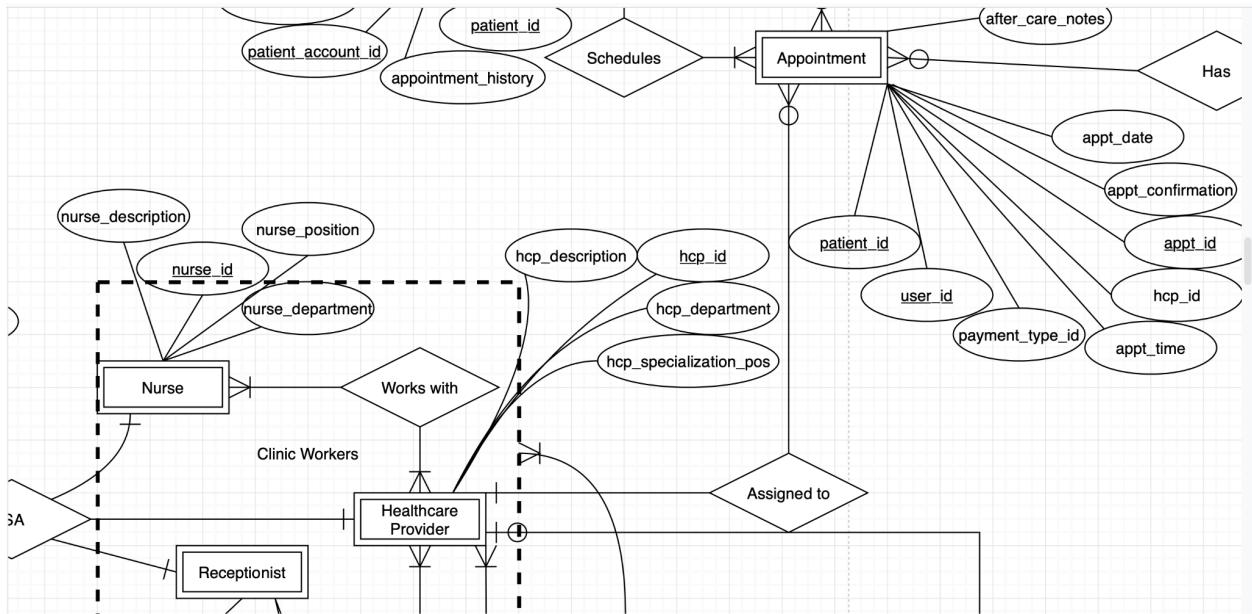
- The screenshot below shows the Employee's relationship to Employee Address.



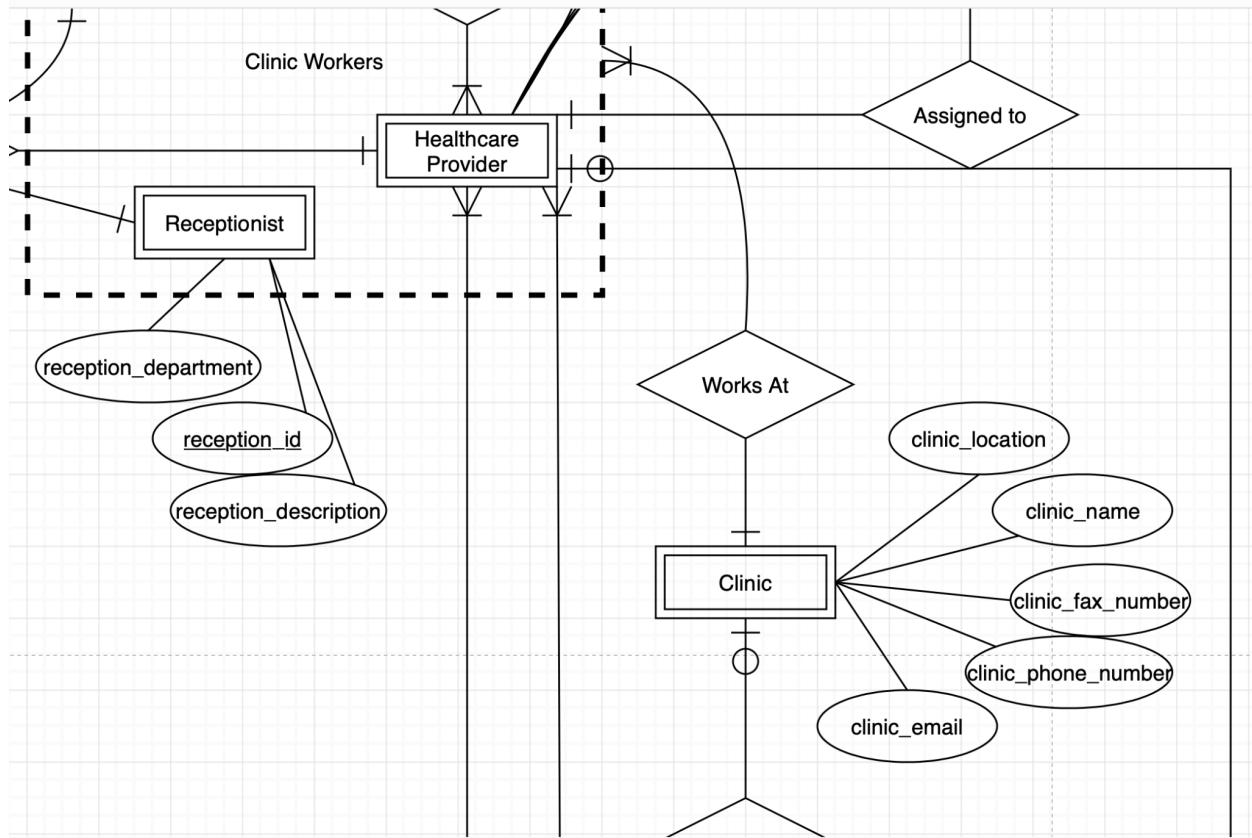
- The screenshot below shows the Employee's ISA relationship to Nurse, Healthcare Provider and Receptionist.



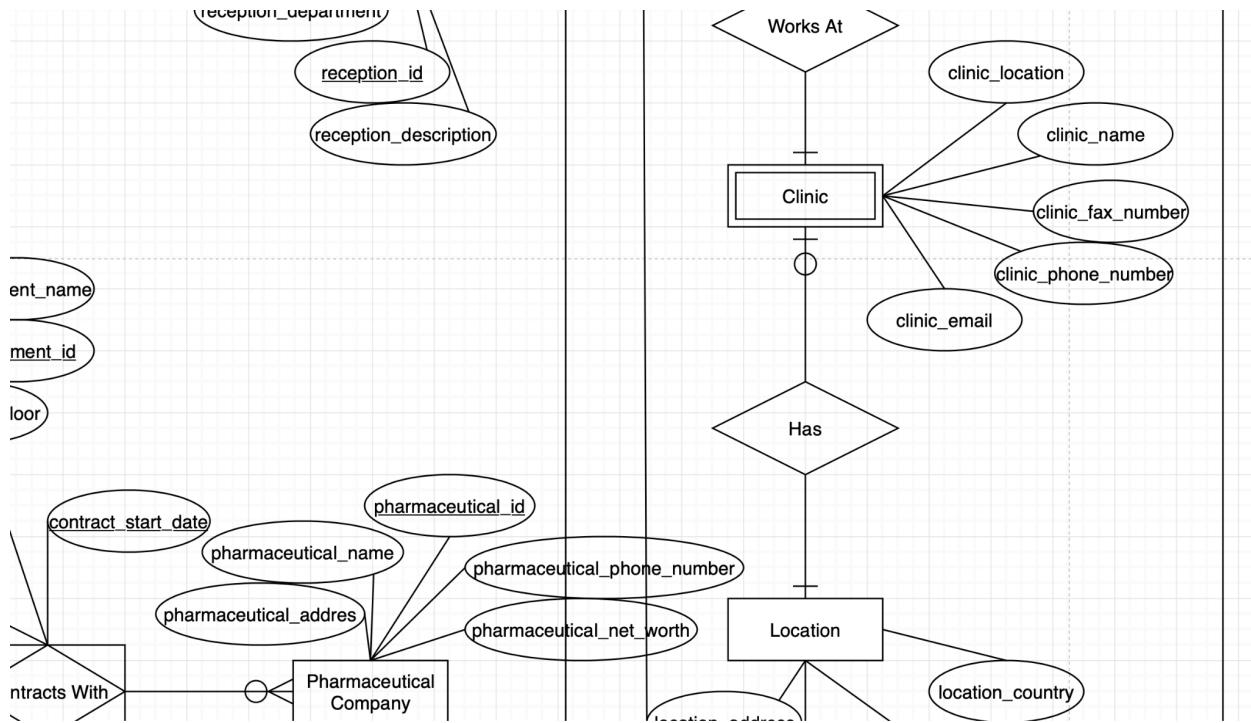
- The screenshot below shows that Appoint is connected to Healthcare Provider.



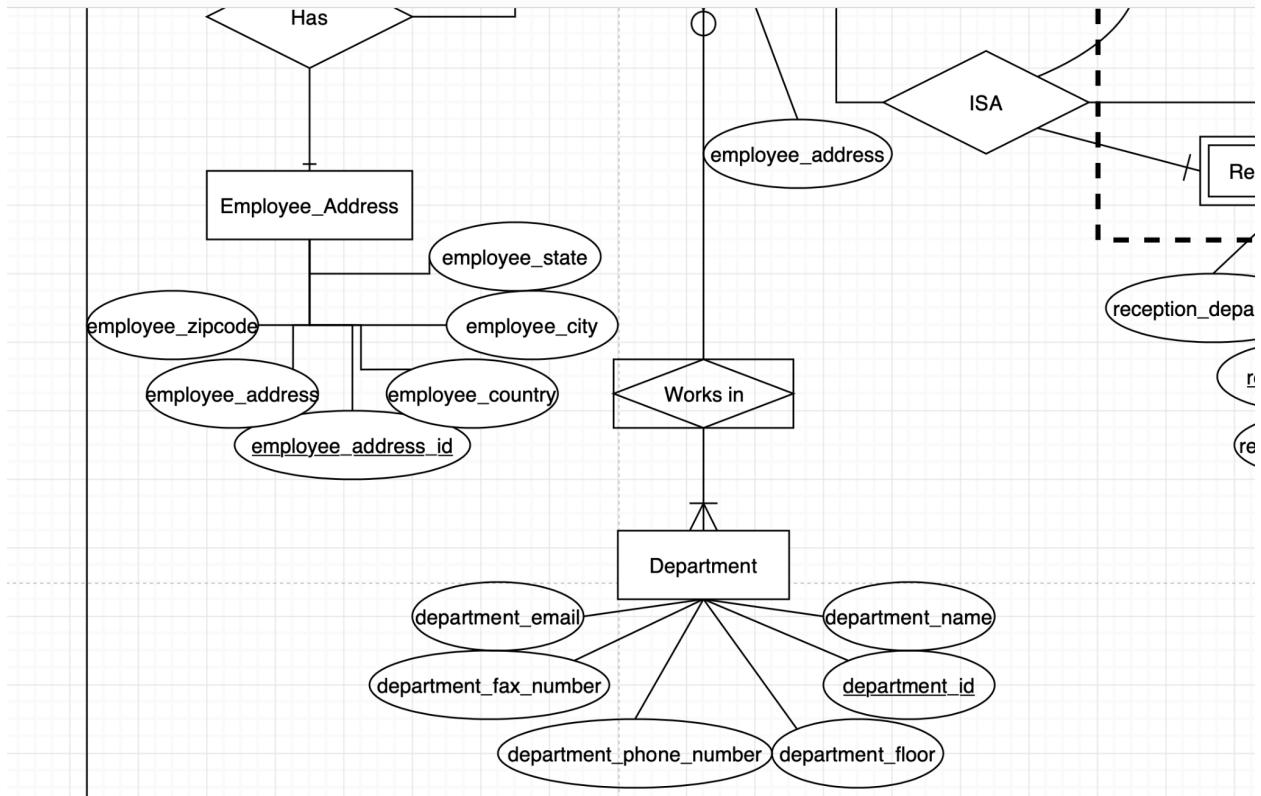
- The screenshot below continues south of the ERD. Notice the cardinality of Clinic Workers to Clinic. Notice the two cardinalities at the bottom of Healthcare Provider.



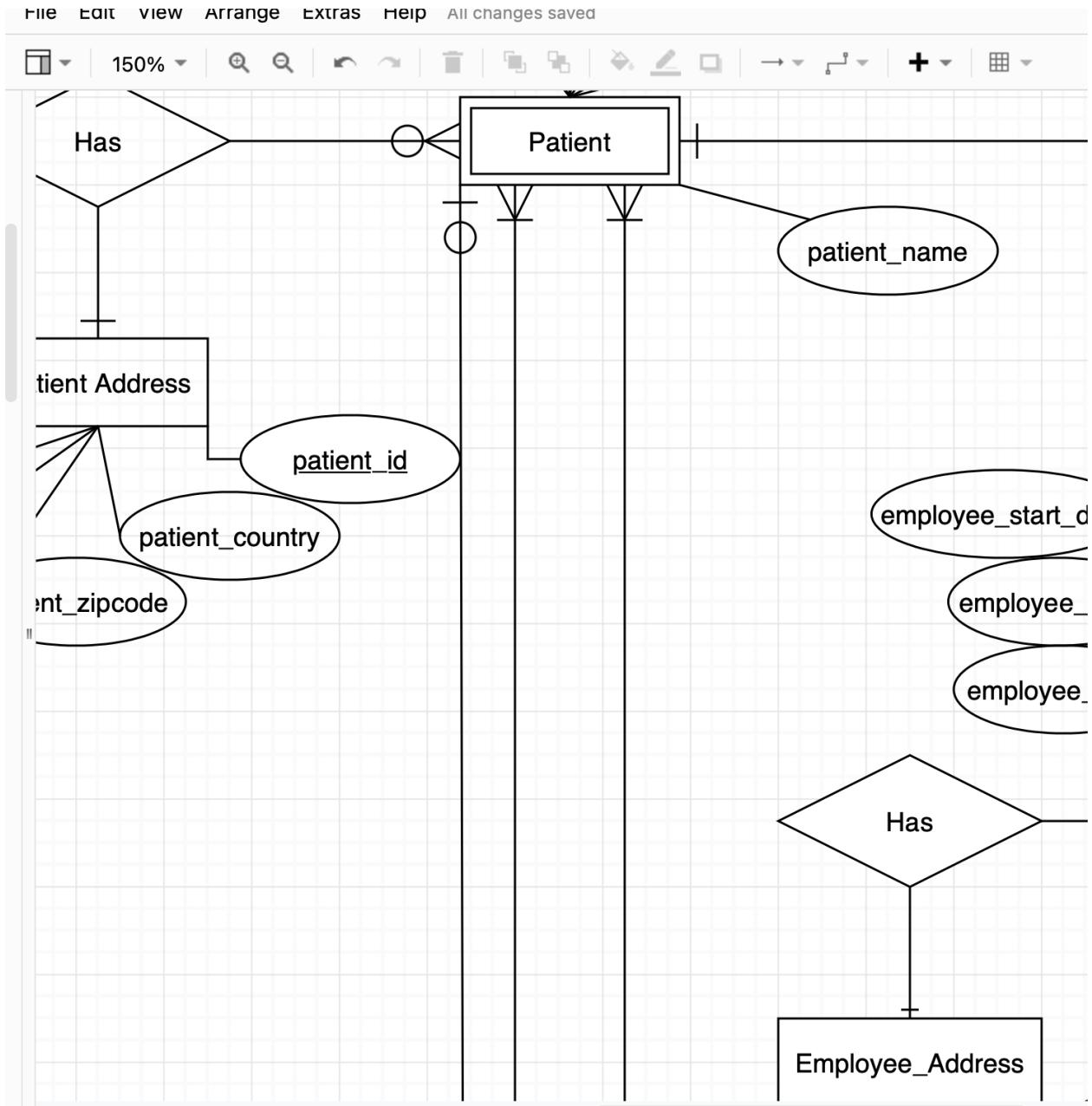
- The screenshot below continues south of the ERD, showing Clinic to Location.



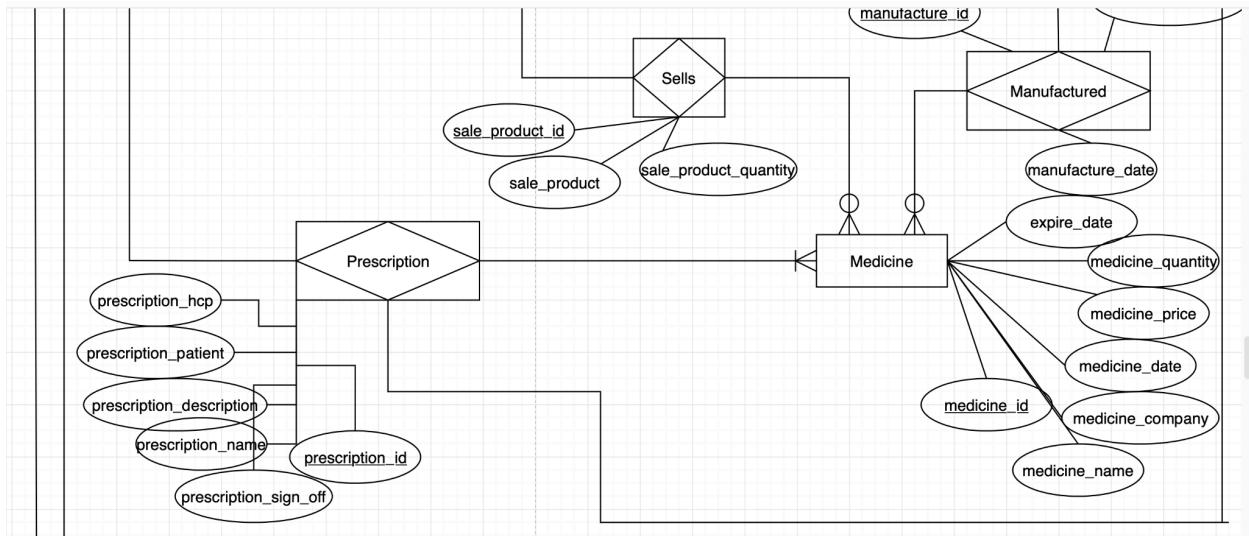
- The screenshot below continues from the Employee entity, showing its relationship to Department.



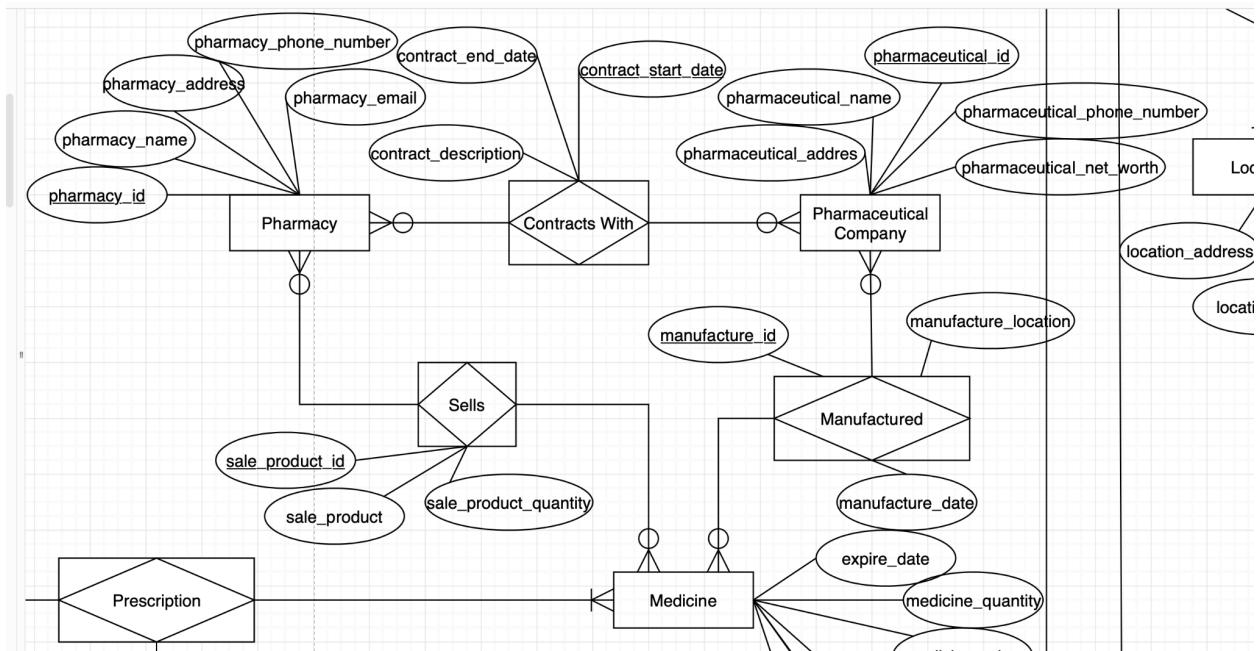
- The screenshot shows the Patient entity and the three cardinalities connecting to it. From left to right of the bottom cardinalities, the cardinalities are for Testing Service (zero or one), Procedure (at least one), and Prescription (at least one)



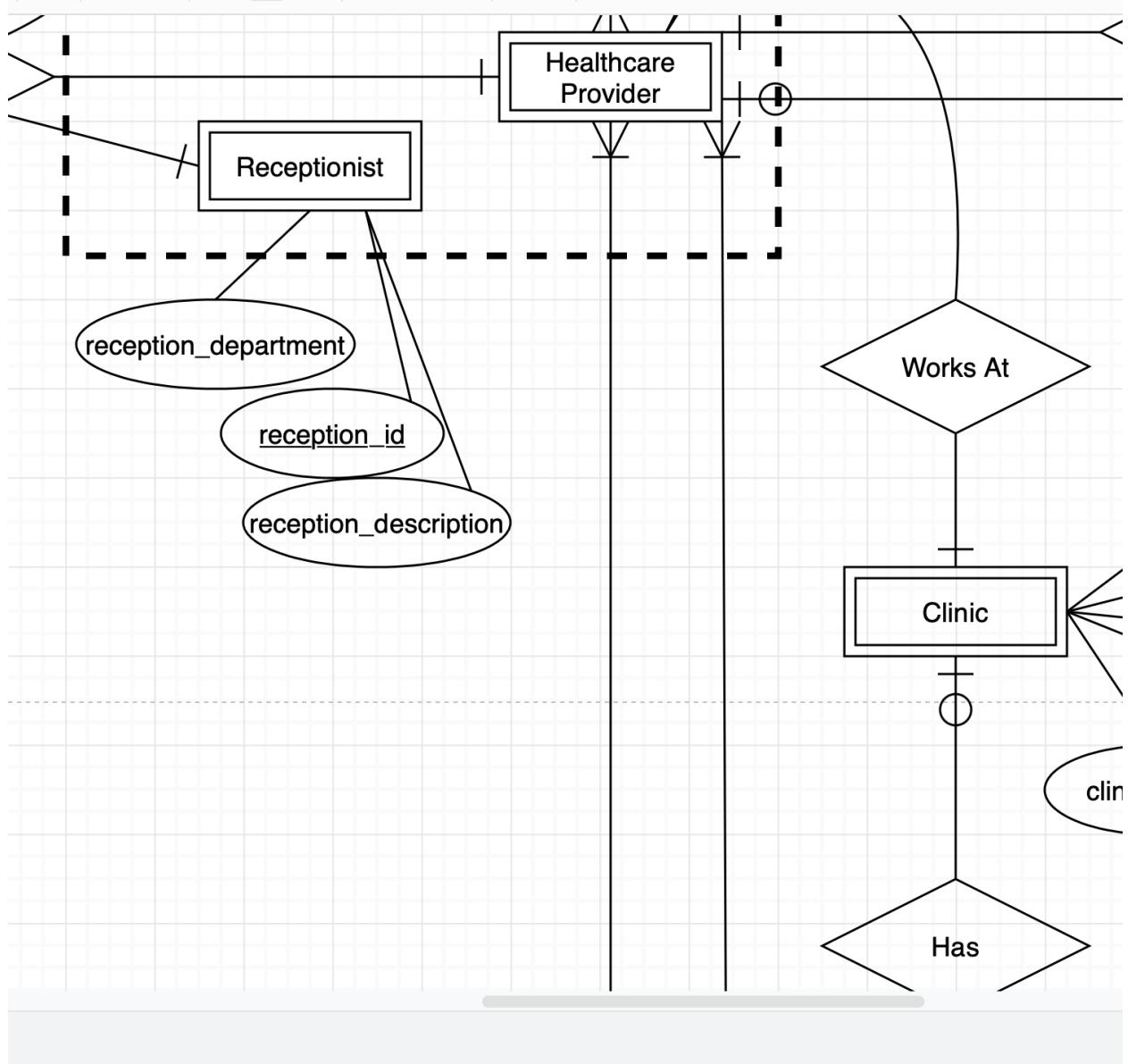
- The screenshot below shows the right most of Patient cardinality from the above screenshot, Prescription to Medicine to Manufacturing.



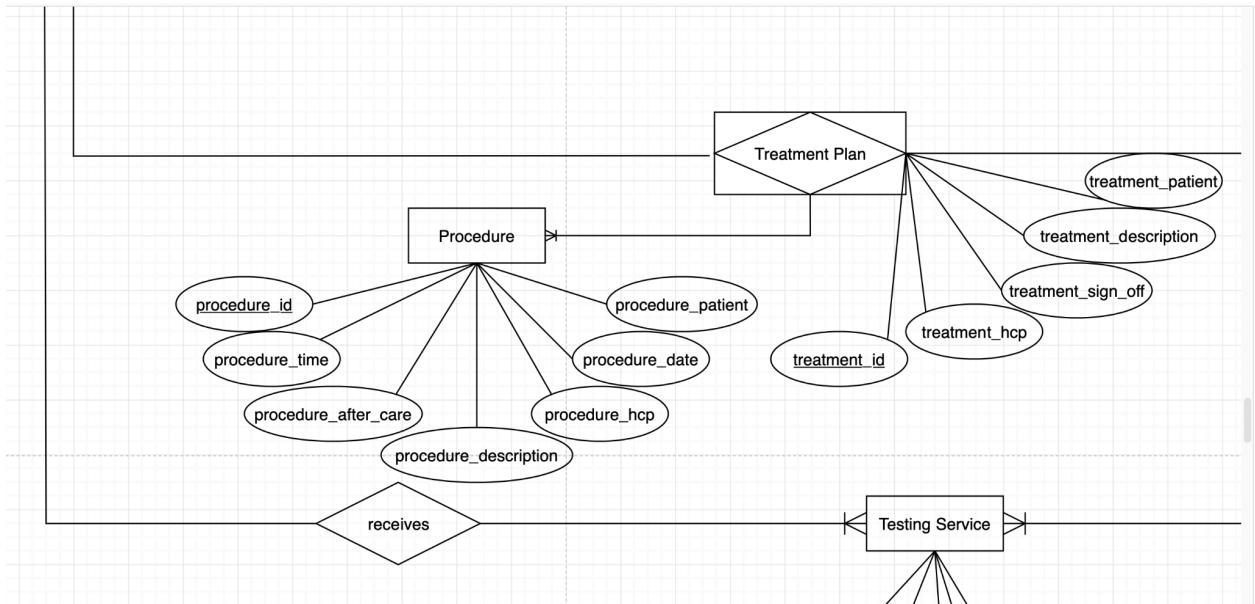
- The screenshot below shows the relationships between Medicine to Manufacturing, Pharmaceutical Company to Contract, Pharmacy to Pharmacy Sales, and finally back to Medicine.



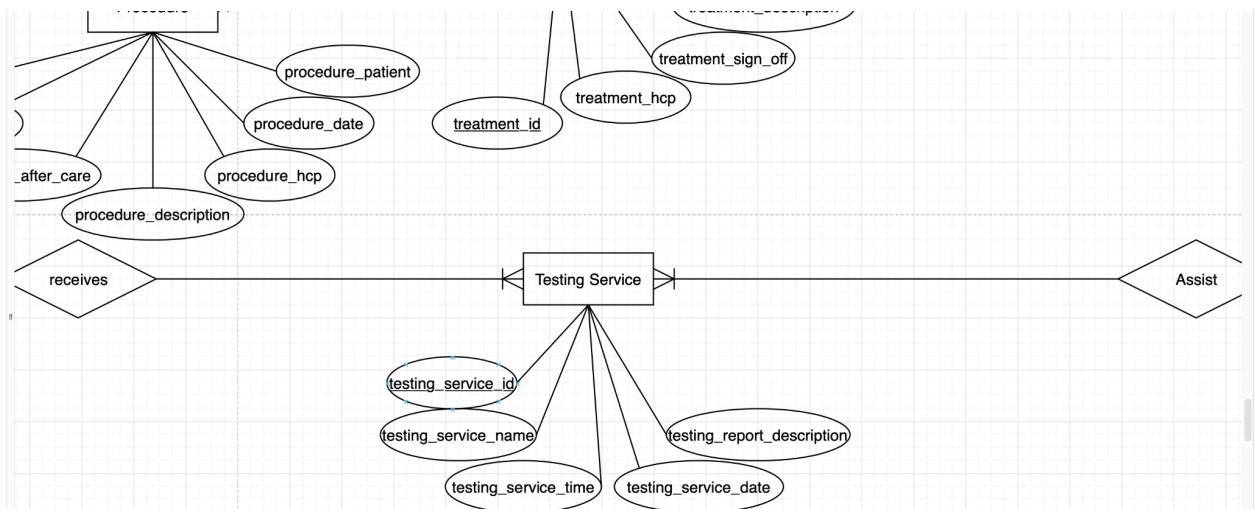
- The screenshot shows the cardinalities at the bottom of Healthcare Provider, those cardinalities point to Prescription and Treatment Plan

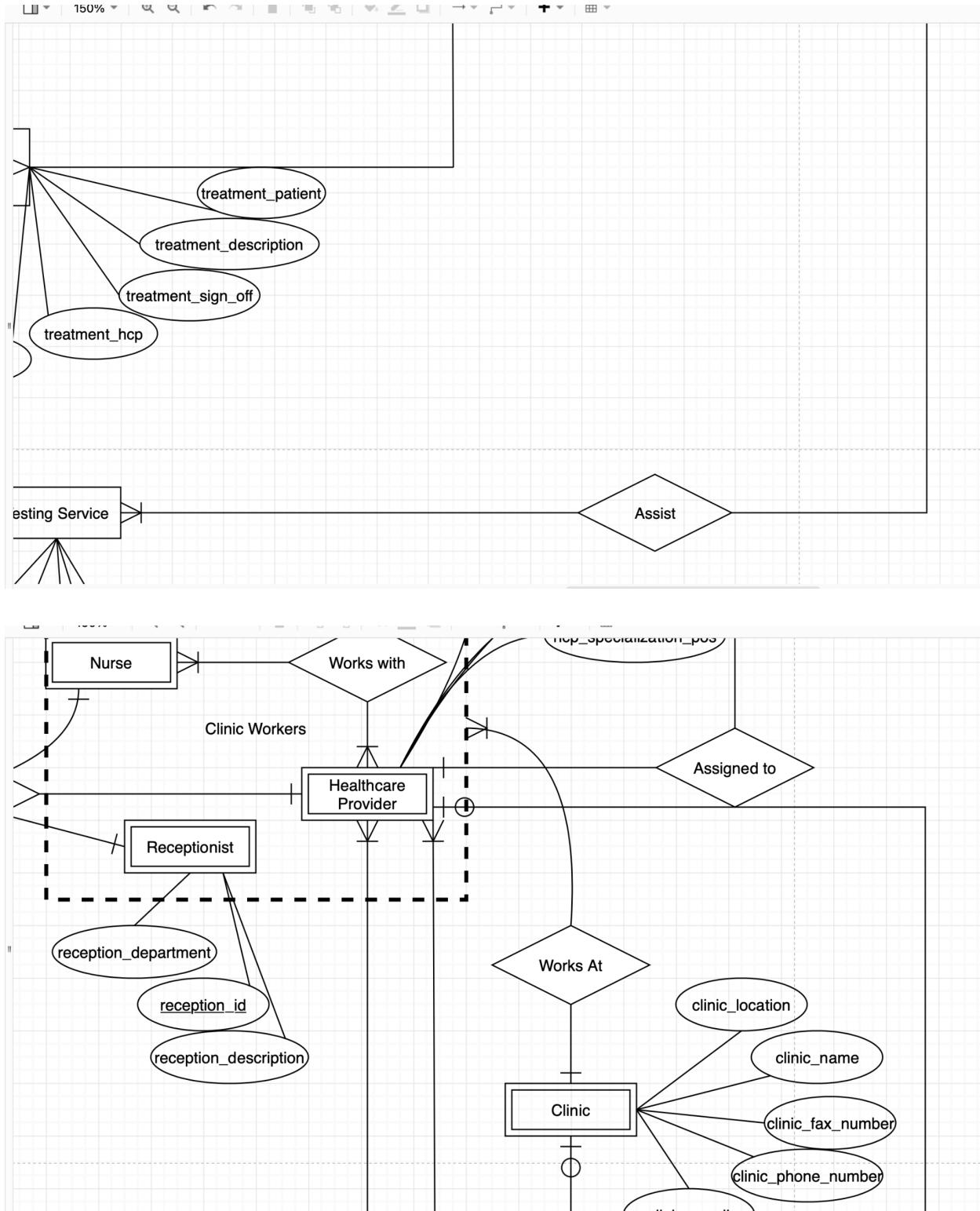


- The screenshot below shows the two Patient cardinalities to Treatment Plan to Healthcare Provider, and Patient to Testing Service to Healthcare Provider.



- The screenshot below shows the Testing Service entity and relationship Assist to Healthcare Provider. The next two screenshots show the relationship line to Healthcare Provider.





- The right outermost relationship line connects the Testing Service entity.
- ERD file shall be on Github as well in case of clarifications.

Section VI: Testing Table

Rule	Entity A	Relation	Entity B	Cardinality	Pass/Fail	Error
1	Patient	Has	Patient Address	M-to-1	Fail	A patient can have multiple addresses
2	Patient	ISA	Registered User	1-to-1	Pass	None
3	Patient	Takes	Medicine	M-to-N	Pass	None
4	Patient	Receives	Procedure	M-to-N	Pass	None
5	General User	Recursive	Registered User	1-to-1	Pass	None
6	Registered User	ISA	Employee	1-to-1	Pass	None
7	Employee	ISA	Nurse	1-to-1	Pass	None
8	Employee	ISA	Healthcare Provider	1-to-1	Pass	None
9	Employee	ISA	Receptionist	1-to-1	Pass	None
10	Employee	ISA	Admin	1-to-1	Pass	None
11	Employee	ISA	CEO	1-to-1	Pass	None
12	Nurse	Works with	Healthcare Provider	M-to-N	Pass	None
13	Employee	Works in	Company	M-to-1	Fail	An employee can work for many companies
14	Company	Has	Department	1-to-M	Pass	None
15	Department	Has	Clinic Workers*	M-to-N	Fail	Clinic workers must work for one department
16	Registered User	Creates Session	Device	M-to-N	Pass	None

17	Device	Logs into	Account	M-to-N	Pass	None
18	General User	Creates	Account	1-to-1	Pass	None
19	Account	Has	Payment Type	1-to-1	Fail	A Payment Type can belong to another account. Ex. User can use their payment type for their child's payments type
20	Account	Has	Account Type	1-to-1	Pass	None
21	Account Type	ISA	Employee Account	1-to-1	Pass	None
22	Account Type	ISA	Patient Account	1-to-1	Pass	None
23	Payment Type	ISA	Bank Account	1-to-1	Pass	None
24	Payment Type	ISA	Insurance Plan	1-to-1	Pass	None
25	Payment Type	Pays for	Billing	M-to-N	Fail	A bill can only be paid by one payment type on an account
26	Bank Account	ISA	Saving Account	1-to-1	Pass	None
27	Bank Account	ISA	Checking Account	1-to-1	Pass	None
28	Insurance Plan	Has	Insurance Company	M-to-1	Pass	None
31	Account	Has	Profile	1-to-1	Pass	None
32	Profile	Creates	Action	M-to-N	Fail	An action can only be created by at

						most one profile
33	Action	Permissions	Employee/ Patient account aggregation	M-to-N	Pass	None
34	Patient Account	Schedules	Appointment	1-to-M	Pass	None
35	Appointment	Has	Date	M-to-1	Pass	None
36	Appointment	Has	Time	M-to-1	Pass	None
37	Appointment	Assigned to	Healthcare Provider	M-to-1	Pass	None
38	Healthcare Provider	Prescribes	Medicine	M-to-N	Pass	None
39	Healthcare Provider	Treatment Plan	Procedure	M-to-N	Pass	None
40	Healthcare Provider	Assist	Test Services	M-to-N	Pass	None
41	Medicine	Manufactured	Pharmaceutical Company	M-to-N	Pass	None
42	Pharmaceutical Company	Contracts With	Pharmacy	M-to-N	Pass	None
43	Pharmacy	Sells	Medicine	M-to-N	Pass	None
44	Clinic Workers ¹	Works At	Clinic	M-to-1	Pass	None
45	Clinic	Has	Location	M-to-N	Fail	It is not possible for a location to have multiple clinics

¹ Clinic Workers consists of entities: Healthcare Provider, Nurse, and Receptionist

Section VII: Database Model/EER

Table (child table)	Foreign Key (parent table)	ON DELETE	ON UPDATE	Comment
Checking	Bank Account (CHECK_BA NK_FK)	ON CASCADE	ON CASCADE	If the bank account is deleted, then the checking account attached to the bank account is deleted.
Saving	Bank Account (SAVING_BA NK_FK)	ON CASCADE	ON CASCADE	If the bank account is deleted, then the savings account attached to the bank account is deleted.
Bank Account	Payment_Type (BANK_PAY_TYPE_FK)	ON CASCADE	NO CASCADE	If the payment type is updated, then the bank account shall propagate that update. If the payment type is deleted, then the bank account shall be deleted
Insurance Plan	Insurance Company (INSURANCE_COMPAN Y_FK)	ON CASCADE	ON CASCADE	If the insurance company is updated, then the insurance plan shall propagate that update. If the insurance company is deleted (i.e goes bankrupt), then the insurance plan shall be deleted.
Insurance Plan	Payment Type (INSURANCE_PAYMENT_FK)	ON CASCADE	ON CASCADE	If the payment type is updated, then the insurance plan shall propagate that update. If the payment type is deleted, then the insurance plan shall be deleted.
Billing	Payment Type (BILLING_PAYMENT_TYPE_FK)	NO ACTION	ON CASCADE	If the payment type is updated, then the billing shall propagate that update. If the payment type is deleted, then the billing shall stay since it has been already paid.
Patient	Account	ON	ON	If the account is updated, then

Account	(PATIENT_A CC_FK)	CASCADE	CASCADE	the patient account shall propagate that update. If it is deleted, then the patient account shall be deleted, since a patient account belongs to the account.
Employee Account	Account (EMPLOYEE _ACC_FK)	ON CASCADE	ON CASCADE	If the account is updated, then the employee account shall propagate that update. If it is deleted, then the employee account shall be deleted, since the employee account belongs to the account.
Account	Payment Type (ACC_PAYM ENT_FK)	NO ACTION	ON CASCADE	If the payment type is updated, then the account shall propagate that update. If the payment type is deleted, then the account belonging to the user is not deleted.
Device	Account (DEVICE_AC C_FK)	NO ACTION	NO ACTION	If the account is updated or deleted, then the device shall have no action done
Session	Device (SESSION_D EVICE_FK)	ON CASCADE	ON CASCADE	If the device is updated, then the session shall propagate that update. If the device is deleted then, the session is also deleted
Session	General User (SESSION_U SER_FK)	ON CASCADE	ON CASCADE	If the general user is updated, then the session shall propagate that update. If the general user is deleted, then the session shall be deleted.
Action	Profile (ACTION_PR OFILE_FK)	NO ACTION	ON CASCADE	If the profile is updated, then the action shall propagate that update. If the profile is deleted, then the action shall remain.
Permission	Action (PERMISSIO N_ACTION_	NO ACTION	ON CASCADE	If the action is updated, then the permissions shall propagate that update. If the

	FK)			action is deleted, then the permissions shall remain.
Account Type Permissions	Permissions (PERMISSIONS_ACC_TYPE_FK)	NO ACTION	ON CASCADE	If the account type is updated, then the permissions attached to the account type is also updated. If the account type is deleted, then the permissions shall remain
Account Type Permissions	Employee Account (ACC_TYPE_EMPLOY_FK)	NO ACTION	ON CASCADE	If the employee account is updated then the account type permissions shall propagate that update. If the employee account is deleted, then the Account type permissions shall remain.
Account Type Permissions	Patient Account (ACC_TYPE_PATIENT_FK)	NO ACTION	ON CASCADE	If the patient account is updated, then the account type permissions shall propagate that update. If the patient account is deleted, then the account type permissions shall remain.
Account	General User (ACC_USER_FK)	ON CASCADE	ON CASCADE	If the general/registered user is updated or deleted (i.e. the user dies), then the account shall propagate that update or delete.
Patient	General User (PATIENT_USER_FK)	ON CASCADE	ON CASCADE	If the general/registered user is updated or deleted (i.e. the user dies), then the patient associated with the user is also updated or deleted.
Patient	Patient Address (PATIENT_ADDRESS_FK)	NO ACTION	ON CASCADE	If the patient address is updated, then the patient shall propagate that update. If the patient address is deleted, then the patient shall not be deleted and shall remain.
Testing Service	Patient (TESTING_P)	NO ACTION	ON CASCADE	If the patient is updated, then the testing service shall

	ATIENT_FK)			propagate that update. If the patient is deleted (i.e dies or is no longer a patient), then the testing service shall remain since it has already been done.
Healthcare Provider	Employee (HCP_EMPL_OY_FK)	ON CASCADE	ON CASCADE	If the employee is updated or deleted (i.e is fired or dies), then the healthcare provider id shall propagate that update or delete.
Testing Service	Healthcare Provider (TESTING_HCP_FK)	NO ACTION	ON CASCADE	If the testing service is updated, then the healthcare provider shall propagate that update. If the testing service is deleted (i.e. cancelled or no longer offered to patients), then the healthcare provider shall still remain.
Treatment Plan	Healthcare Provider (TREATMENT_HCP_FK)	NO ACTION	ON CASCADE	If the healthcare provider is updated, then the treatment plan shall propagate that update. If the healthcare provider is deleted, then the treatment plan shall remain, since the healthcare provider has already authorized it for the patient.
Treatment Plan	Procedure (TREATMENT_PROCEDURE_FK)	ON CASCADE	ON CASCADE	If the procedure is updated or deleted, then the treatment plan shall propagate that update or delete, since if the procedure is no longer available to perform then the treatment plan must be canceled.
Treatment Plan	Patient (TREATMENT_PATIENT_FK)	ON CASCADE	ON CASCADE	If the patient is deleted (i.e. dies or is no longer a patient at Happy Health Medical Foundation), then their treatment plan is also deleted since it is no longer needed.

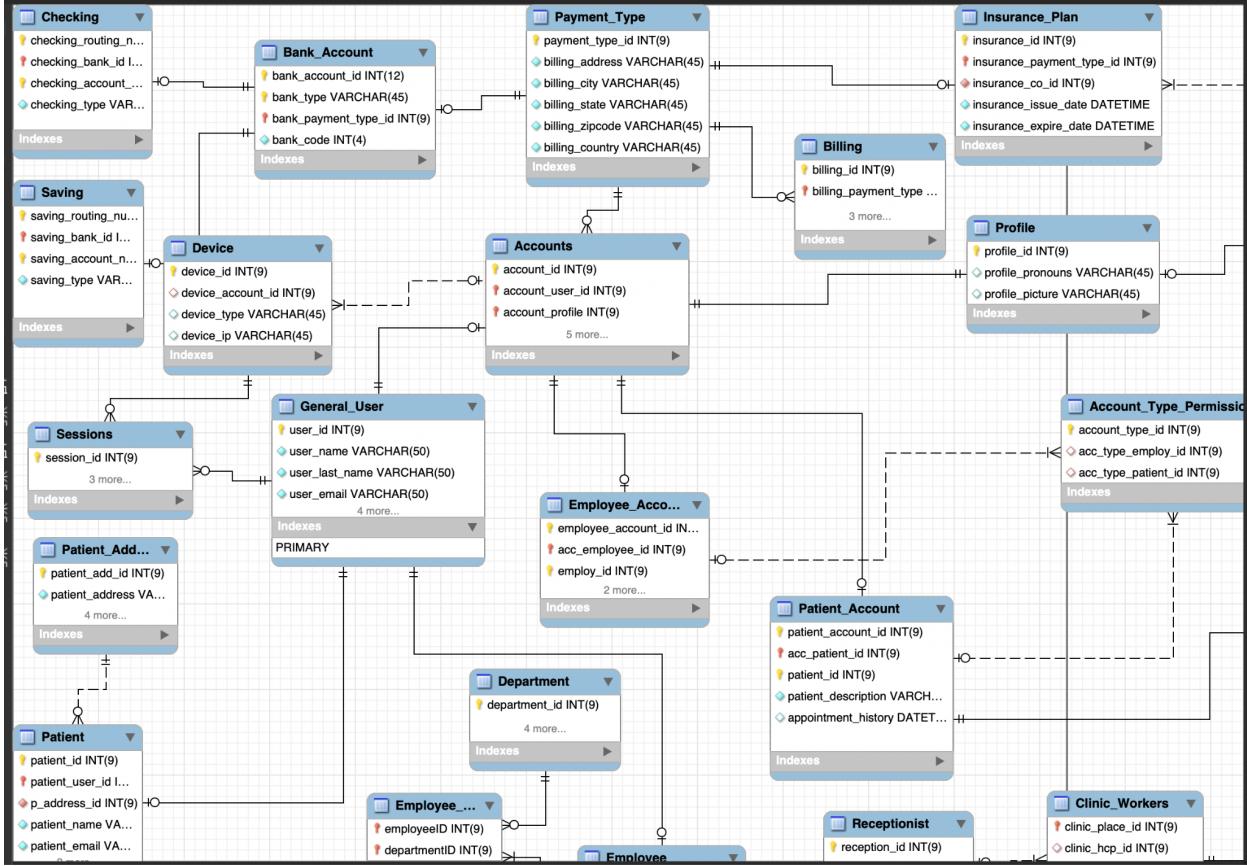
Prescription	Healthcare Provider (PRESCRIPTION_HCP_FK)	NO ACTION	ON CASCADE	If the healthcare provider is updated, then the prescription shall propagate that update. If the healthcare provider is deleted (i.e is fired or dies), then the prescription shall still remain since the healthcare provider already authorized it for the patient.
Prescription	Medicine (PRESCRIPTION_MED_FK)	ON CASCADE	ON CASCADE	If the medicine is updated or deleted, then the prescription shall propagate that update or delete (no longer authorized to prescribe or patient to take), since if there is no medicine then it cannot be prescribed.
Prescription	Patient (PRESCRIPTION_PATIENT_FK)	ON CASCADE	ON CASCADE	If the patient is deleted (i.e dies or no longer is a patient at Happy Health Medical Foundation), then their prescription is also no longer needed so it is deleted.
Pharmacy Sales	Medicine (PHARM_SALES_MED_FK)	NO ACTION	NO ACTION	If the medicine is updated or deleted, then there is no change to the pharmacy sales since it has already been sold and recorded.
Pharmacy Sale	Pharmacy (PHARM_SALES_FK)	NO ACTION	ON CASCADE	If the pharmacy is updated, then the pharmacy sales shall propagate that update. If the pharmacy is deleted (i.e bankruptcy), then the pharmacy sales shall remain, since the sale is a record of its previous transactions.
Contract	Pharmacy (CONTRACT_PHARMACY_FK)	ON CASCADE	NO ACTION	If the pharmacy is updated, then the contract does not propagate that update since the contract is final. If the pharmacy is deleted (i.e. bankruptcy) then the contract

				is also deleted since the contract is no longer binding.
Contract	Pharmaceutica l Company (CONTRACT _PHARMCO_ FK)	ON CASCADE	NO ACTION	If the pharmaceutical company is updated, then the contract does not propagate that update since the contract is final. If the pharmaceutical company is deleted (i.e bankruptcy) then the contract is deleted, since the contract is no longer binding.
Manufacturing	Pharmaceutica l Company (MANUFACT URE_PHAR MCO_FK)	ON CASCADE	ON CASCADE	If the pharmaceutical company is updated or deleted, then the manufacturing shall propagate that update or delete.
Manufacturing	Medicine (MAUFACTU RE_MED_FK)	NO ACTION	NO ACTION	If the medicine is updated or deleted, then the manufacturing shall not propagate that update or delete since what it manufactures is dependent on the pharmaceutical company.
Employee	General User (EMPLOYEE _USER_FK)	ON CASCADE	ON CASCADE	If the general user is updated or deleted, then the employee shall propagate that update or delete, since it is a ISA relationship.
Employee	Employee Address	NO ACTION	ON CASCADE	If the employee address is updated, then the employee shall propagate that update. If the employee address is deleted, then the employee shall remain.
Receptionist	Employee (RECEPTION _EMPLOY_F K)	ON CASCADE	ON CASCADE	If the employee is updated or deleted (i.e. was fired or dies), then the receptionist id associated with that employee id is also deleted.
Nurse	Employee	ON	ON	If the employee is updated or

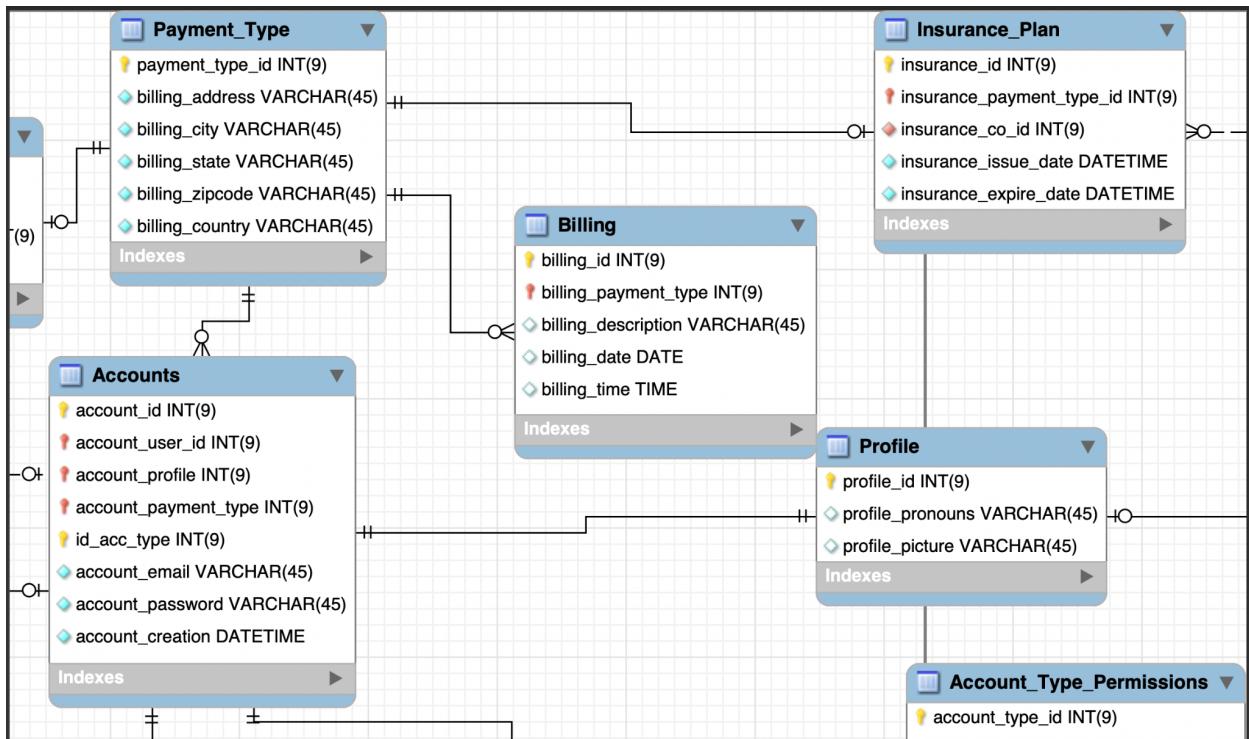
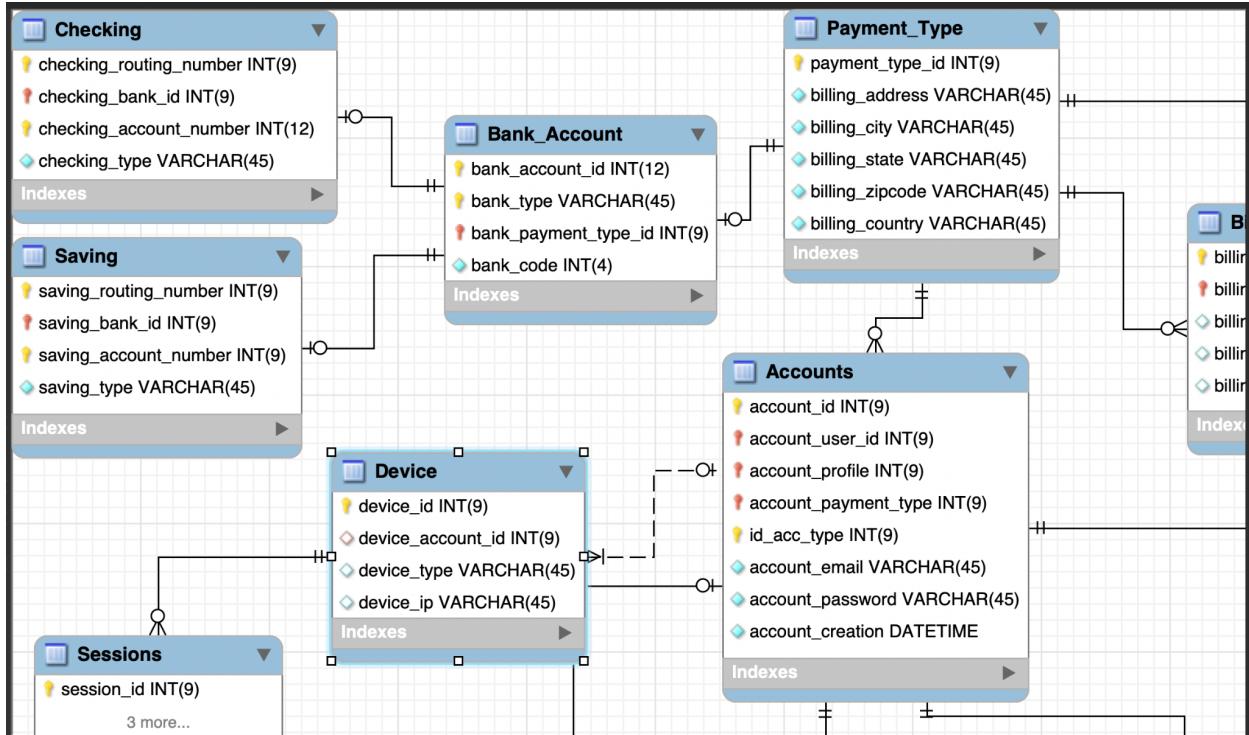
	(NURSE_EM PLOY_FK)	CASCADE	CASCADE	deleted (i.e. was fired or dies), then the nurse id associated with that employee id is also deleted.
Clinic Workers	Healthcare Provider (CLINIC_WORKER_HCP_FK)	ON CASCADE	ON CASCADE	If the healthcare provider is updated or deleted (i.e is fired or dies), then its row in the clinic workers table is also deleted, meaning that the healthcare provider no longer works at the clinic.
Clinic Workers	Nurse (CLINIC_WORKER_NURSE_FK)	ON CASCADE	ON CASCADE	If the nurse is updated or deleted (i.e is fired or dies), then its row in the clinic workers table is also deleted, meaning that the nurse no longer works at the clinic.
Clinic Workers	Receptionist (CLINIC_WORKER_RECEPTION_FK)	ON CASCADE	ON CASCADE	If the receptionist is updated or deleted (i.e is fired or dies), then its row in the clinic workers table is also deleted, meaning that the receptionist no longer works at the clinic.
Clinic Workers	Clinic (CLINIC_WORKER_FK)	ON CASCADE	ON CASCADE	If the clinic is updated or deleted (i.e. is no longer a viable place to work), then the row in the Clinic works is also updated or deleted, since the clinic workers cannot work in a place that is no longer available.
Clinic	Location (CLINIC_LOCATION_FK)	ON CASCADE	ON CASCADE	If the location is updated or deleted, then the clinic shall propagate that update or delete, since the clinic needs a location
Appointment	Patient Account (APPT_PATIENT_ACC_FK)	ON CASCADE	ON CASCADE	If the patient account is deleted, then the appointment shall also be deleted since the appointment needs to have a patient.

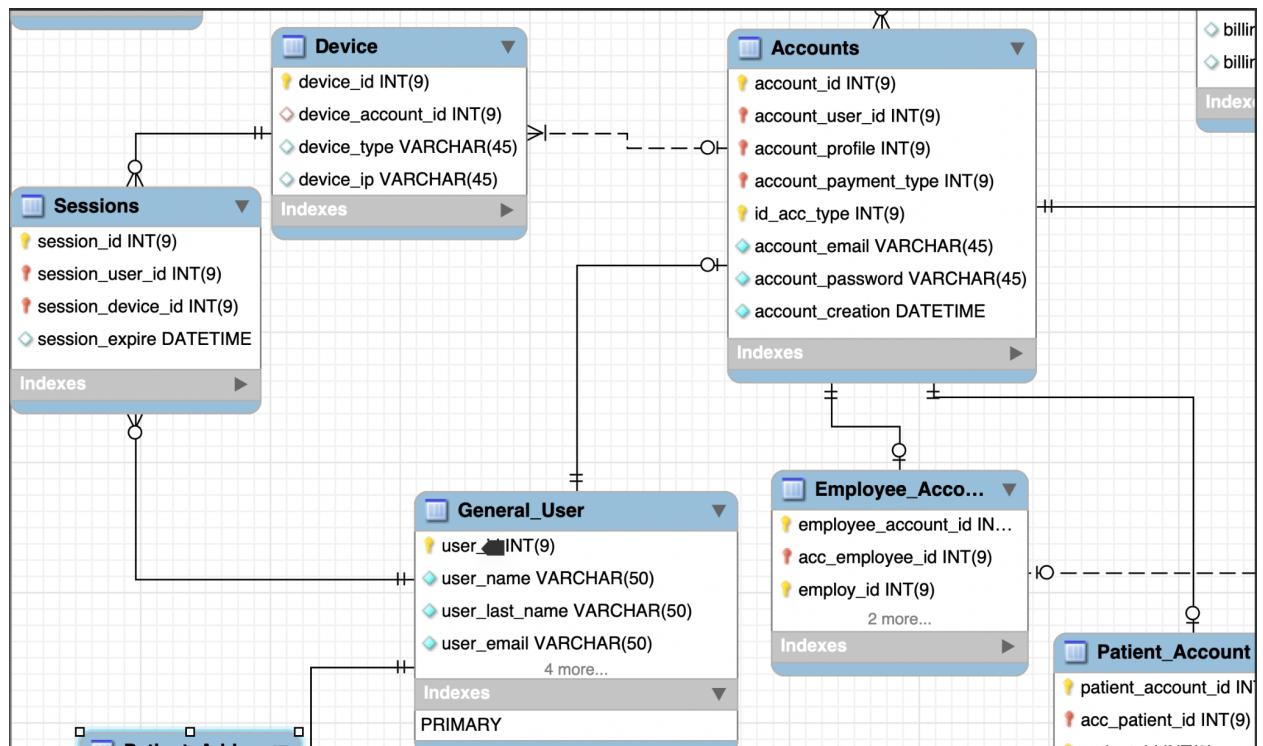
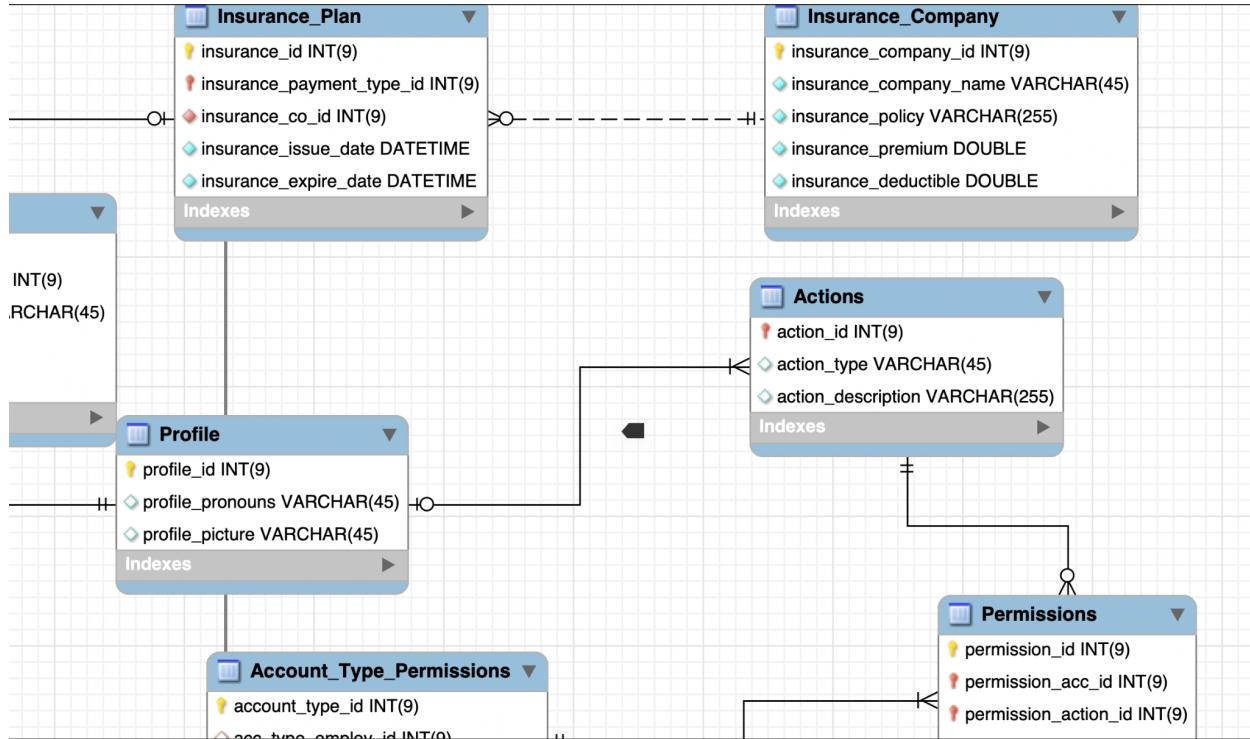
Appointment	Date (APPT_DATE_FK)	ON CASCADE	ON CASCADE	If the date is deleted (like if the date is no longer available to book or taken off), then the appointment shall be deleted since the appointment needs to have a booked date.
Appointment	Time (APPT_TIME_FK)	ON CASCADE	ON CASCADE	If the time is deleted (like if the time is no longer available to book), then the appointment is also deleted since it needs to have a booked time.
Appointment	Healthcare Provider (APPT_HCP_FK)	ON CASCADE	ON CASCADE	If the healthcare provider is deleted (i.e. is fired or no longer available), then the appointment shall also be deleted since it needs to be assigned to the healthcare provider.
Employee Department	Department	ON CASCADE	ON CASCADE	If the department is updated or deleted, then the employee department shall propagate that update or delete.
Employee Department	Employee	ON CASCADE	ON CASCADE	If the employee is updated or deleted (i.e dies or got fired), then the employee department shall propagate that update or delete, since they no longer work for the department.

Enhanced Entity Relationship Diagram Screenshots

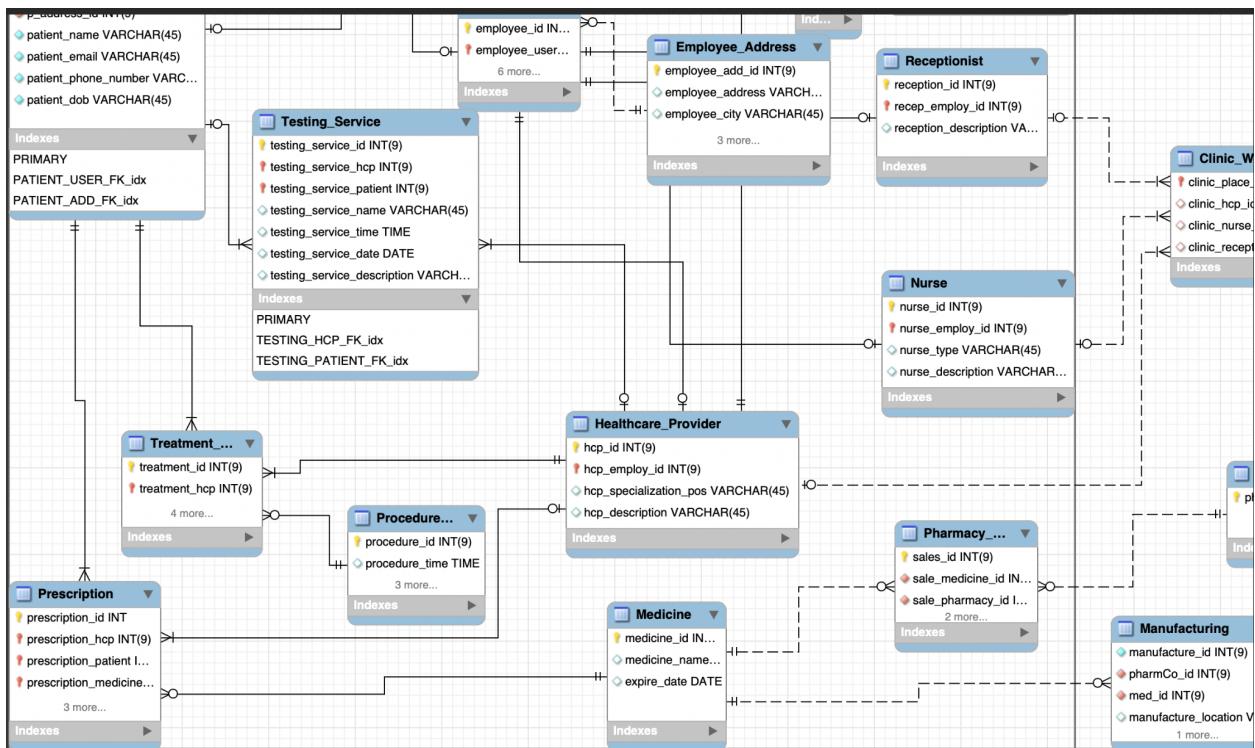
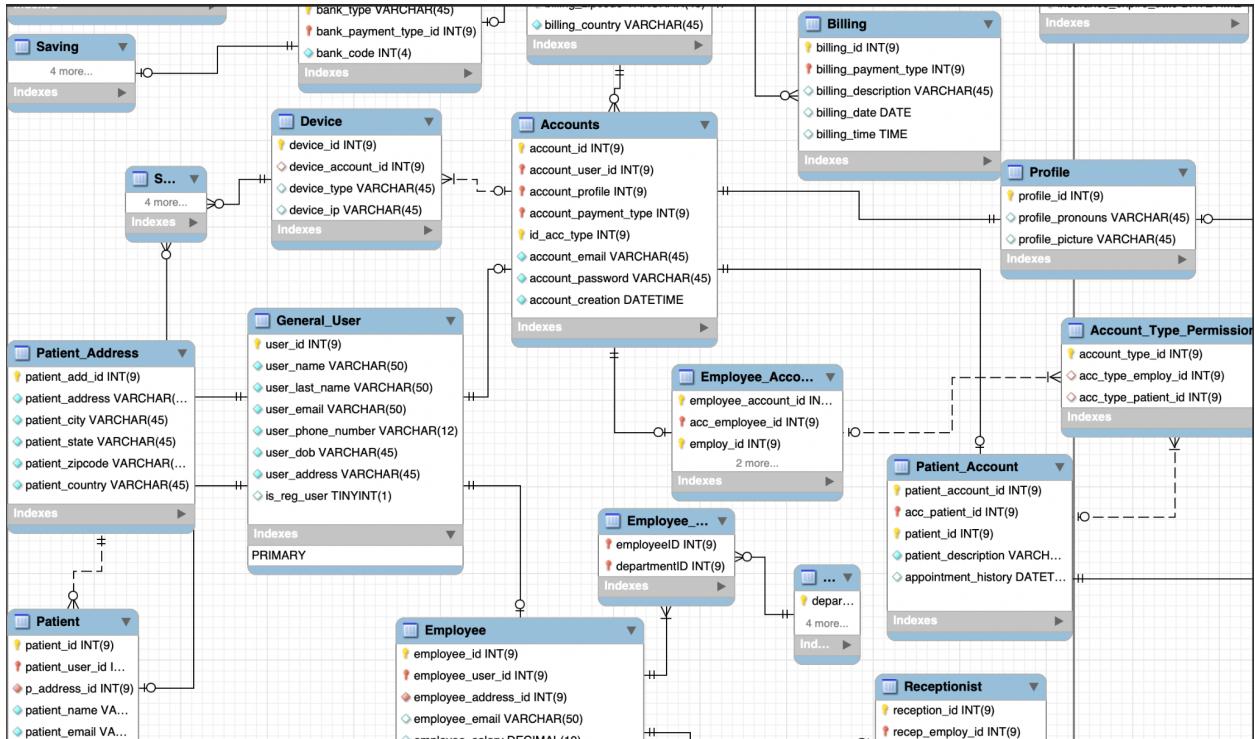


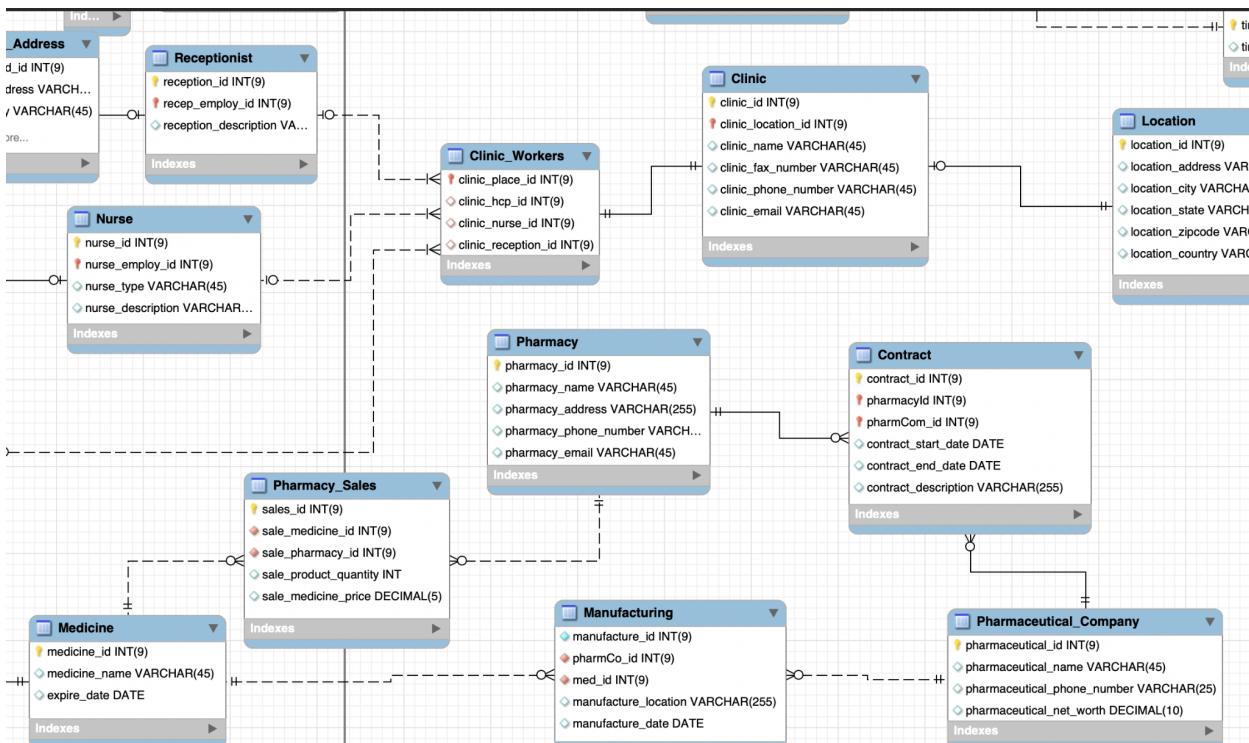
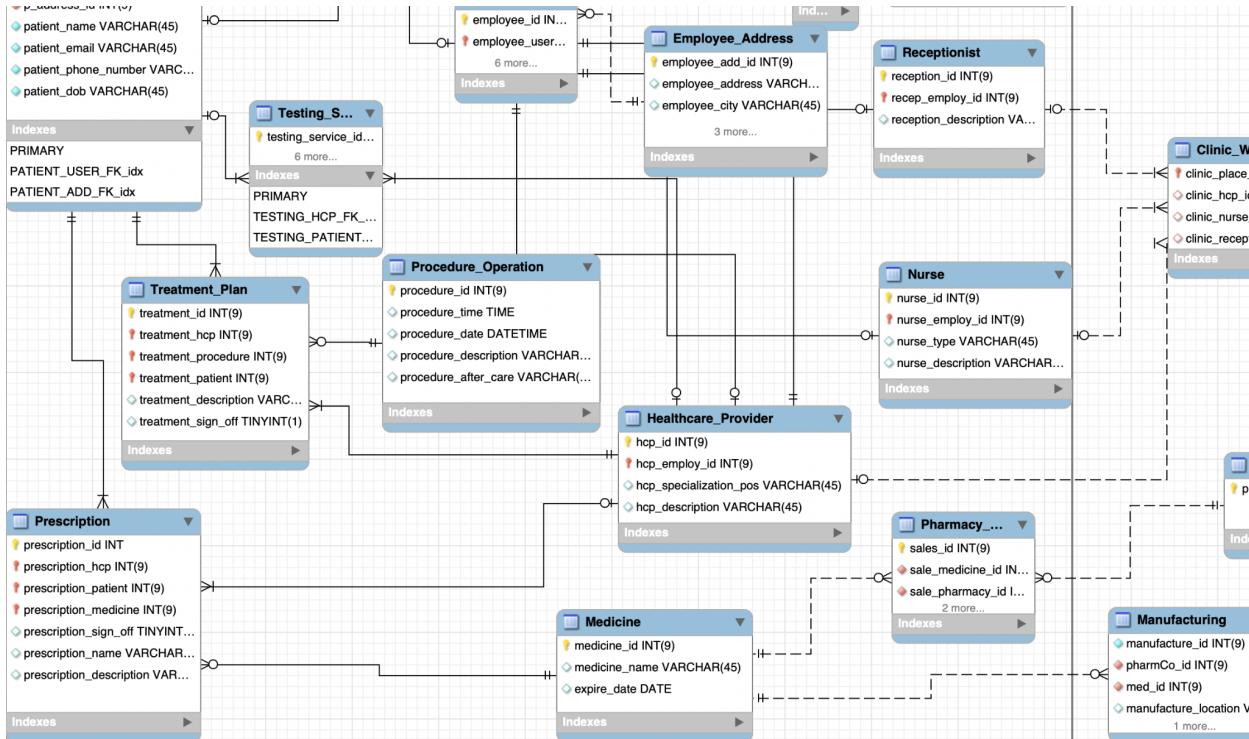
* Screenshots start from left to right and top to bottom.

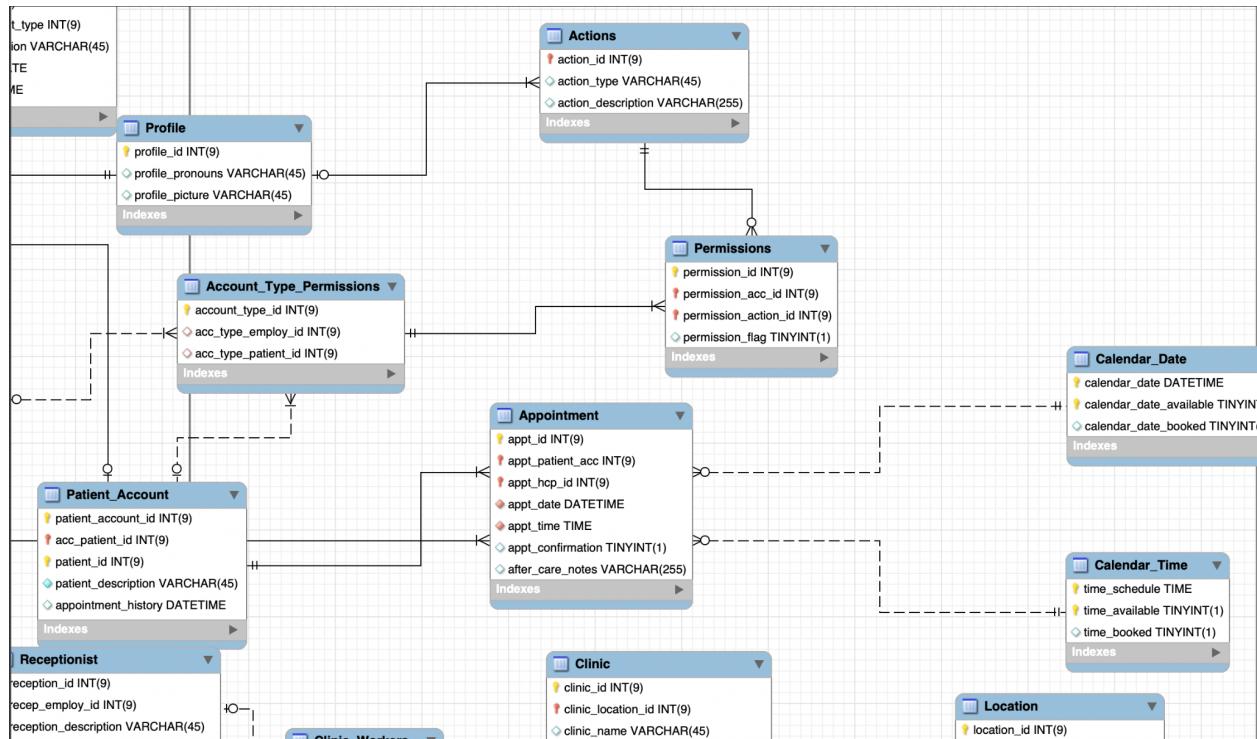
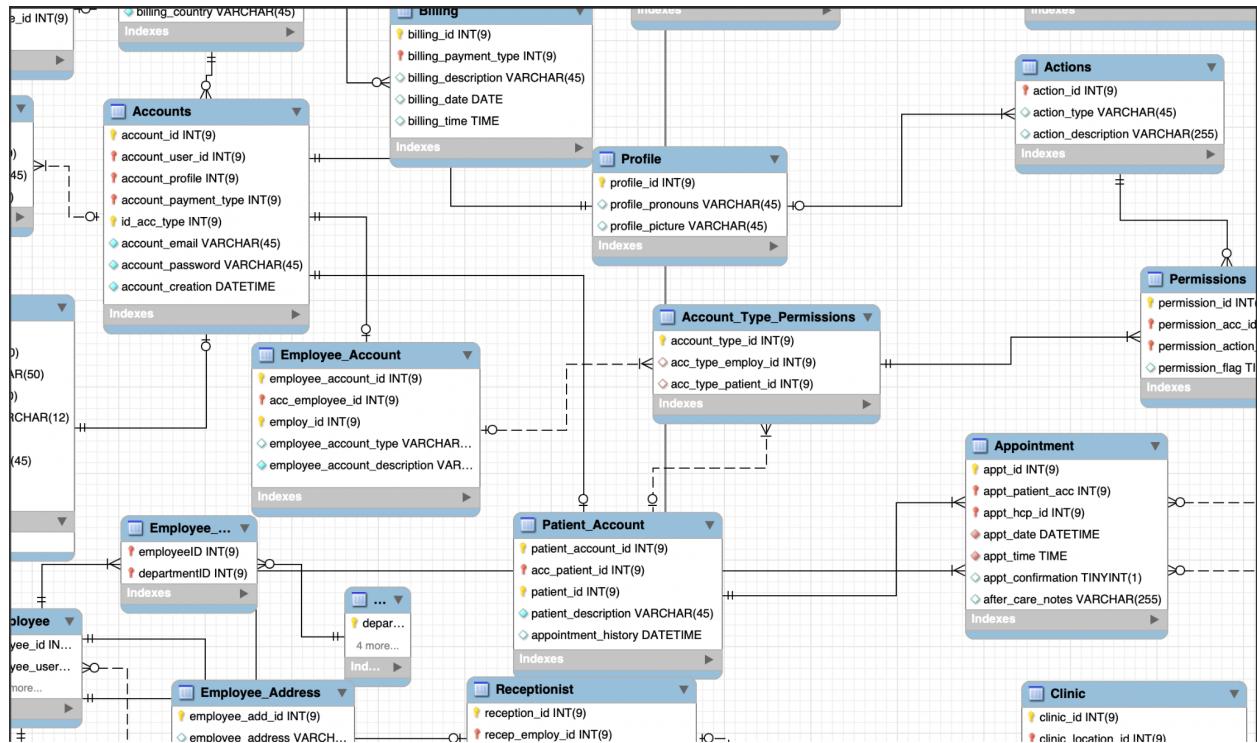




* Right of Account cardinality goes to Profile.







Section XI: Testing Table

Entity	SQL QUERY	Pass/Fail	Error Description	Possible Solution
Accounts	Delete	Pass	None	None
Accounts	Update	Pass	None	None
Account_Type_Permissions	Delete	Pass	None	None
Account_Type_Permissions	Update	Pass	None	None
Appointment	Delete	Pass	None	None
Appointment	Update	Pass	None	None
Bank_Account	Delete	Pass	None	None
Bank_Account	Update	Pass	None	None
Calendar_Date	Delete	Pass	None	None
Calendar_Date	Update	Pass	None	None
Calendar_Time	Delete	Pass	None	None
Calendar_Time	Update	Pass	None	None
Checking	Delete	Pass	None	None
Checking	Update	Pass	None	None
Clinic	Delete	Pass	None	None
Clinic	Update	Pass	None	None
Clinic_Workers	Delete	Pass	None	None
Clinic_Workers	Update	Pass	None	None
Contract	Delete	Pass	None	None
Contract	Update	Pass	None	None
Department	Delete	Pass	None	None

Department	Update	Pass	None	None
Device	Delete	Pass	None	None
Device	Update	Pass	None	None
Employee	Delete	Pass	None	None
Employee	Update	Pass	None	None
Employee_Account	Delete	Pass	None	None
Employee_Account	Update	Pass	None	None
Employee_Address	Delete	Pass	None	None
Employee_Address	Update	Pass	None	None
Employee_Department	Delete	Pass	None	None
Employee_Department	Update	Pass	None	None
General_User	Delete	Pass	None	None
General_User	Update	Pass	None	None
Healthcare_Provider	Delete	Pass	None	None
Healthcare_Provider	Update	Pass	None	None
Insurance_Company	Delete	Pass	None	None
Insurance_Company	Update	Pass	None	None
Insurance_Plan	Delete	Pass	None	None
Insurance_Plan	Update	Pass	None	None
Location	Delete	Pass	None	None

Location	Update	Pass	None	None
Manufacturing	Delete	Pass	None	None
Manufacturing	Update	Pass	None	None
Medicine	Delete	Pass	None	None
Medicine	Update	Pass	None	None
Nurse	Delete	Pass	None	None
Nurse	Update	Pass	None	None
Patient	Delete	Pass	None	None
Patient	Update	Pass	None	None
Patient_Account	Delete	Pass	None	None
Patient_Account	Update	Pass	None	None
Patient_Address	Delete	Pass	None	None
Patient_Address	Update	Pass	None	None
Payment_Type	Delete	Pass	None	None
Payment_Type	Update	Pass	None	None
Permissions	Delete	Pass	None	None
Permissions	Update	Pass	None	None
Pharmaceutical_Company	Delete	Pass	None	None
Pharmaceutical_Company	Update	Pass	None	None
Pharmacy	Delete	Pass	None	None
Pharmacy	Update	Pass	None	None
Pharmacy_Sales	Delete	Pass	None	None
Pharmacy_Sales	Update	Pass	None	None
Prescription	Delete	Pass	None	None

Prescription	Update	Pass	None	None
Procedure_Operation	Delete	Pass	None	None
Procedure_Operation	Update	Pass	None	None
Account_Profile	Delete	Pass	None	None
Account_Profile	Update	Pass	None	None
Receptionist	Delete	Pass	None	None
Receptionist	Update	Pass	None	None
Saving	Delete	Pass	None	None
Saving	Update	Pass	None	None
Sessions	Delete	Pass	None	None
Sessions	Update	Pass	None	None
Testing_Service	Delete	Pass	None	None
Testing_Service	Update	Pass	None	None
Treatment_Plan	Delete	Pass	None	None
Treatment_Plan	Update	Pass	None	None

Action Output	Time	Action	Response	Duration / Fetch Time
✓	1	13:52:35 USE OnlineAppointmentDB	0 row(s) affected	0.00044 sec
✓	2	13:52:35 SET SQL_SAFE_UPDATES = 0	0 row(s) affected	0.00092 sec
✓	3	13:52:35 DELETE FROM Device WHERE device_id = 4	1 row(s) affected	0.017 sec
✓	4	13:52:35 UPDATE Device SET device_type = 'Windows' WHERE device_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.0011 sec
✓	5	13:52:35 DELETE FROM Permissions WHERE permission_id = 2	1 row(s) affected	0.00068 sec
✓	6	13:52:35 UPDATE Permissions SET permission_flag = 0 WHERE permission_id = 3	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.00048 sec
✓	7	13:52:35 DELETE FROM Account_Type_Permissions WHERE account_type_id = 2	1 row(s) affected	0.00065 sec
✓	8	13:52:35 UPDATE Account_Type_Permissions SET acc_type_patient_id = null	2 row(s) affected Rows matched: 11 Changed: 2...	0.00053 sec
✓	9	13:52:35 DELETE FROM Patient WHERE patient_id = 1	1 row(s) affected	0.00091 sec
✓	10	13:52:35 UPDATE Patient SET patient_phone_number = '408-123-4567' WHERE patient_id = 2	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.00035 sec
✓	11	13:52:35 DELETE FROM Testing_Service WHERE testing_service_id = 1	1 row(s) affected	0.00043 sec
✓	12	13:52:35 UPDATE Testing_Service SET testing_service_time = '11:30:00' WHERE testing_service_id = 2	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.00080 sec
✓	13	13:52:35 DELETE FROM General_User WHERE user_id = 3	1 row(s) affected	0.00095 sec
✓	14	13:52:35 UPDATE General_User SET user_last_name = 'Tran' WHERE user_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.00075 sec
✓	15	13:52:35 DELETE FROM Patient_Address WHERE patient_addr_id = 3	1 row(s) affected	0.00051 sec
✓	16	13:52:35 UPDATE Patient_Address SET patient_city = 'San Mateo' WHERE patient_addr_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.00054 sec
✓	17	13:52:35 DELETE FROM Sessions WHERE session_id = 5	1 row(s) affected	0.0010 sec
✓	18	13:52:35 UPDATE Sessions SET session_expire = '2021-11-11 12:00:00' WHERE session_id = 6	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.00057 sec
✓	19	13:52:35 DELETE FROM Accounts WHERE account_id = 4	1 row(s) affected	0.00047 sec
✓	20	13:52:35 UPDATE Accounts SET account_email = 'gummybears99@yahoo.com' WHERE account_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.0013 sec
✓	21	13:52:35 DELETE FROM Patient_Account WHERE patient_account_id = 3	1 row(s) affected	0.00060 sec
✓	22	13:52:35 UPDATE Patient_Account SET patient_description = 'This is a new patient at HHMF' WHERE patient_account_id = 2	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.00067 sec
✓	23	13:52:35 DELETE FROM Employee_Account WHERE employee_account_id = 4	1 row(s) affected	0.00058 sec
✓	24	13:52:35 UPDATE Employee_Account SET employee_account_type = 'HEALTHCARE PROVIDER' WHERE employee_account...	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.00072 sec
✓	25	13:52:35 DELETE FROM Account_Profile WHERE profile_id = 5555	1 row(s) affected	0.00046 sec
✓	26	13:52:35 UPDATE Account_Profile SET profile_pronouns = 'He/His' WHERE profile_id = 8888	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.0015 sec
✓	27	13:52:35 DELETE FROM Payment_Type WHERE payment_type_id = 8	1 row(s) affected	0.00043 sec
✓	28	13:52:35 UPDATE Payment_Type SET billing_address = '1930 Grapevine Lane' WHERE payment_type_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.00045 sec
✓	29	13:52:35 DELETE FROM Actions WHERE action_id = 1	1 row(s) affected	0.00054 sec
✓	30	13:52:35 UPDATE Actions SET action_type = 'RESCHEDULE APPOINTMENT' WHERE action_id = 3	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.00100 sec
✓	31	13:52:35 DELETE FROM Billing WHERE billing_id = 1	1 row(s) affected	0.00059 sec
✓	32	13:52:35 UPDATE Billing SET billing_date = '2021-02-20' WHERE billing_id = 2	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.00068 sec
✓	33	13:52:35 DELETE FROM Bank_Account WHERE bank_account_id = 4	1 row(s) affected	0.0011 sec
✓	34	13:52:35 UPDATE Bank_Account SET bank_code = '8080' WHERE bank_account_id = 5	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.00068 sec
✓	35	13:52:35 DELETE FROM Checking WHERE checking_routing_number = 194748254	1 row(s) affected	0.00054 sec
✓	36	13:52:35 UPDATE Checking SET checking_type = 'STUDENT' WHERE checking_routing_number = 388264616	1 row(s) affected Rows matched: 1 Changed: 1 W...	0.00078 sec