Team 9 Medical Clinic

Our medical database can be used to manage medical records, patient information, appointments, prescriptions, and diagnostic tests. It can also be used for administrative tasks such as managing employee information, office addresses, and login credentials.

Mini-world DB Operation:

Lisa is a new patient who lives in Texas and has never visited our medical clinic before. To schedule her first appointment, she creates an account online by entering her personal information, such as name, age, contact details, and health insurance which will be stored in the "Patient" entity in the database. She then schedules an appointment by approving from a list of time frames with a primary care doctor working at the clinic's office located in Houston. Returning users are allowed the additional option of updating medical information by using their login.

On the day of her appointment, Lisa arrives at the clinic and is greeted by the receptionist. The receptionist gives Lisa a medical history sheet to fill out. Once Lisa finishes filling out the medical sheet the receptionist enters it into the database by using her employee login for the portal.

Lisa is then assigned to a primary care doctor who works at the clinic's Houston office. The doctor is given the medical history sheet from the receptionist and then performs a General Check Up to determine the cause of her symptoms. The doctor writes down the results of the tests, determines a diagnosis, and writes a prescription for medications with instructions for Lisa.

The doctor refers Lisa to a specialist at another location. The doctor's referral is also written down and Lisa is told to make an appointment with the receptionist or online at her convenience.

After Lisa completes her consultation, she goes to the receptionist to make a payment for the services provided. The receptionist enters the details of the transaction into the web portal and provides Lisa with a receipt. The receptionist also receives the papers with Lisa's medical information from the doctor and enters the information into the database, which gets stored in their respective entities.

When Lisa arrives home, she logs in online to schedule her appointment with her specialist, which allows her to view available appointments and select a time that works for her. If Lisa tries to schedule an appointment with a specialist without approval from her primary physician, it will alert her to the requirement and direct her to contact her primary care doctor.

Entities

- Employees
- Patient
- Doctor
- Appointment
- Referrals
- Invoice
- General_CheckUp
- Blood_CBC_test
- Medical_History
- Prescription
- Diagnosis
- Office
- Login
- Availability

Relationship

- Many *Employees* work in an *Office* **(M:1)**
    - Many *Employees* work under a *Doctor* **(M:1)**
    - *Employees* can access *LOGIN* **(1:1)**
    - *Employees(admin)* can create many *Invoices* **(1:M)**
- *Doctors* approve *Appointments* **(1:M)**
    - Many *Doctors* work in an *Office* **(M:1)**
    - *Doctors* can access *LOGIN* **(1:1)**
- *Patients* access *LOGIN* **(1:1)**
    - *Patients* can be referred many *Referrals* **(1:M)**
    - *Patients* can have many Checkups **(1:M)**
    - Patients can be assigned many Invoices **(1:M)**
    - Many *Patients* can have many *Doctors* **(M:M)**
- Many *Appointments* can belong to one *Patient* **(M:1)**
    - Many *Appointments* are handled by one *Doctor* **(M:1)**
    - Many *Appointments* belong to one *Office* **(M:1)**
    - One *Appointment* can have one *Referral* **(1:1)**
- Many *Referrals* refer to one Office **(M:1)**,
    - Many Referrals refer to one *Doctor* **(M:1)**
    - Many *Prescriptions* will be written by a *Doctor* **(M:1)**
    - Many *Prescriptions* will be prescribed to one *Patient* **(M:1)**
- A *Diagnosis* refers to one *General Check Up* **(1:1)**
    - Multiple *Diagnoses* refer to one *Patient* **(M:1)**
    - Multiple Diagnoses refer to one *Doctor* **(M:1)**
- A *General Checkup* contains one *Medical History* **(1:1)**
    - *General Checkup* refers to at most one *Blood-CDC-Test* **(1:1)**
    - Multiple *General Checkups* are done by a *Doctor* **(M:1)**
- Multiple *Medical Histories* refer to one *Patient* **(M:1)**
- An Office has many Available options(**1:M**)

Constraints

Primary Keys

- Employees - Employee_ID
- Patient - Patient_ID
- Doctor - Doctor_ID
- General_CheckUp - Checkup_ID
- Availability - Availiabitliy_ID
- Appointment - app_ID
- Prescription - Prescription_ID
- Diagnosis - Diag_ID
- Office - Office_ID
- Blood_CBC_test - Blood_ID
- Login - Log_username
- Referrals - Ref_ID
- Medical History - Med_h_ID
- Invoice - Invoice_ID

Foreign Keys

- Employees.office_ID = Office.office_id
- Employees.Doctor_ID = Doctor.doctor_id
- Employees.log_username = Login.log_username
- General_Checkup.gc_medical_history = med_h_ID
- General_Checkup.gc_blood_test_ID = blood_ID
- General_Checkup.doctor_ID = Doctor.doctor_ID
- Patient.Checkup_ID = General_Checkup.checkup_id
- Patient.log_username = Login.log_username
- Patient.Patient_ref_ID = Referrals.ref_ID
- Patient.patient_doctor = Doctor.doctor_id
- Doctor.log_username = Login.log_username
- Medical_History.patient_ID = Patient.patient_ID
- Diagnosis.diag_checkup_ID = General_Checkup.checkup_ID
- Diagnosis. doctor_ID = Doctor.doctor_ID
- Diagnosis.patient_ID = Patient.patient_ID
- Prescription.doctor_id = Doctor.doctor_ID
- Prescription.patient_id = Patient.patient_id
- Appointment.appt_doctor_id = Doctor.doctor_id
- Appointment.appt_patient_id = Patient.patient_id
- Appointment.appt_office_id = Office.office_ID
- Appointment.ref_id = Referrals.ref_id
- Referrals.office_id = Office.office_id
- Referrals.doctor_specialization = Doctor.doctor_specialization
- Invoice.patient_ID = patient.patient_ID
- Availability.Office_ID = Doctor.Office_ID
- Office.Doctor_ID = Doctor.Doctor_ID

- All Primary Keys - NOT NULL, UNIQUE
- All attributes in the following except for primary keys and foreign keys = **NULL**:
    - **General_checkup, Medical_history, Bood_CBC_test, Diagnosis, Medication, Prescription**
- All attributes in **Patient** except for primary keys and foreign keys are **NOT NULL**
- All attributes in **Doctor** except for primary keys and foreign keys are **NOT NULL**
- All attributes in **Office** except for primary keys and foreign keys are **NULL**
- All attributes in **Employee** except for primary keys and foreign keys are **NOT NULL**
- All attributes in **Referrals** except for primary keys and foreign keys are **NULL**
- All attributes in **Appointment** except for primary keys and foreign keys are **NULL**
- **Log_pass** - check >=8 characters

Triggers

- Deny patient from requesting an appointment with a specialist type Doctor
- Deny doctors from updating or changing information of a patient that does not have themselves as the patient's primary doctor
- Deny Employees without Admin attribute status "TRUE" when updating, creating anything outside of personal information

Reports

- The administration can retrieve medical information about patients. Aggregate: Ratio of male to female patients, average age
- The administration can view complete invoice information. Aggregate: Amount owed and Amount paid for each clinic
- Patients can view their medical history: Aggregate: Total # of visits in the past 6 months and which doctor performance which diagnosis

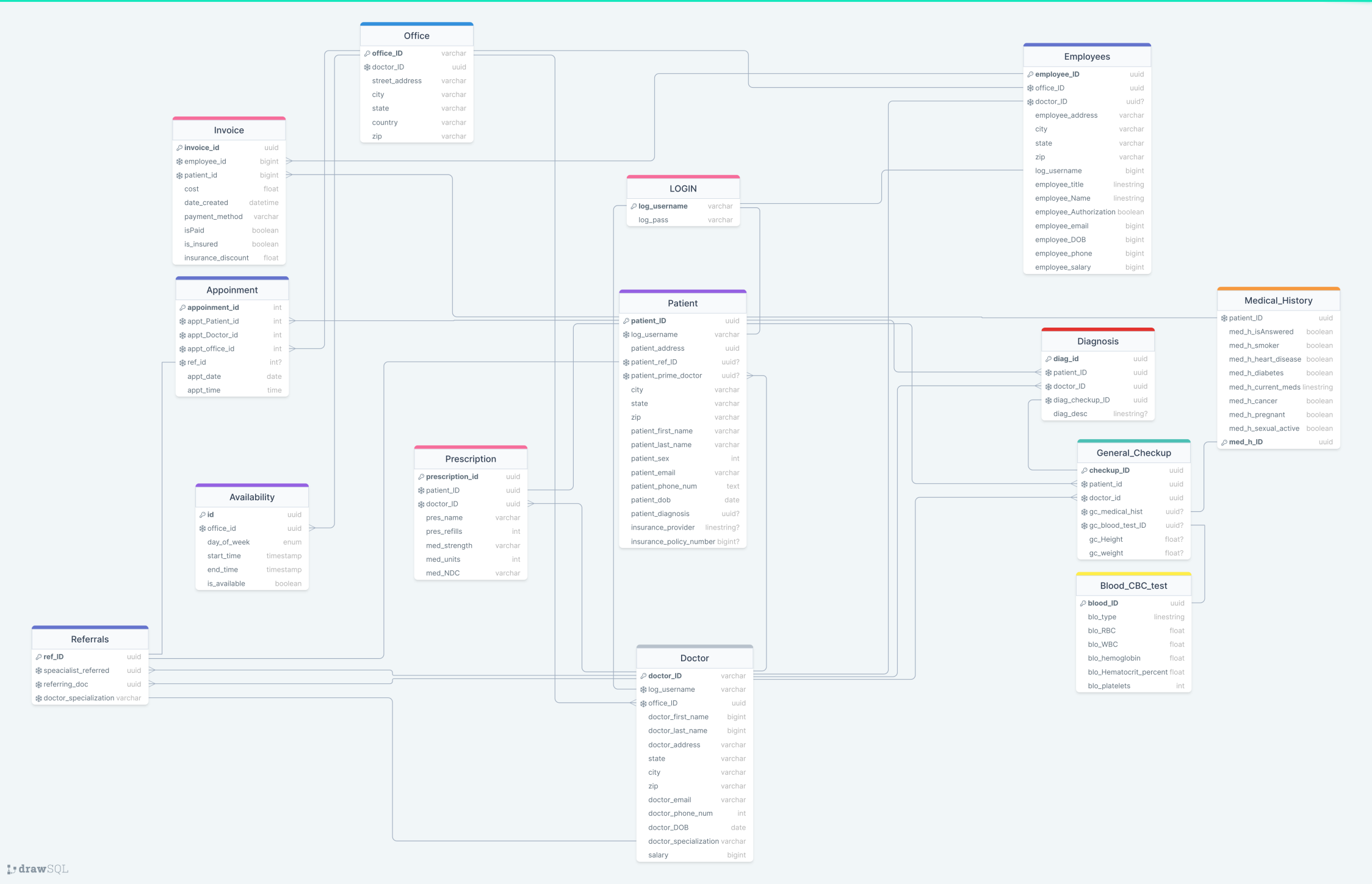
Functional Dependencies

- **Employee_ID ->** Employee_title, Employee_name, Employee_email, Employee_DOB, Employee_phone, Employee_salary, log_username, employee_authorizaton, employee_address
- **Office_ID ->** office_address
- **Availability ->** day_of_week, start_time, end_time, is_available
- **Appointment_ID ->** appt_date, appt_time
- **Prescription_ID ->** pres_name, pres_refills, med_strength, med_units, med_NDC
- **Doctor_ID ->** doctor_first_name, doctor_last_name, doctor_email, doctor_phone_num, doctor_DOB, doctor_specifilization, salary, doctor_address
- **Patient_ID ->** Patient_first_name, patient_last_name, patient_email, patient_phone_num, patient_gender, patient_DOB, insurance_provider, insurance_policy_number, patient_diagnosis, patient_address
- **Log_username ->** log_pass
- **Diag_ID ->** diag_description

- **Checkup_ID ->** gc_height, gc_weight
- **Blood_ID ->** blo_type, blo_RBC, blo_WBC, blo_hemoglobin, blood Hemtocrit_percent, blo_platelets
- **Med_h_ID ->** med_h_isAnswered, med_h_smoker, med_h_heart_disease, med_h_current_meds, med_h_cancer, med_h_pregnant, med_h_sexual_active
- **Invoice_ID ->** Cost, Date_created, isPaid, Payment_method, is_insured, insurance_discount

Semantic Constraints

- Only Patients with a Referral ID can create an appointment for a Doctor with a specialization
- Each Patient must have at least one Doctor before seeing the doctor
- Each Patient must have a unique patient ID
- Each Appointment must have a date & start time, and a Doctor ID associated with it
- Each Patient must have a medical history
- Blood results must be associated with a Patient
- Each invoice must have payment information associated with it, including amount charged and method of payment
- Each general checkup must be associated with a Patient and a Doctor
- Each referral must be associated with a patient and a referring Doctor, and must include information on the reason for referral and the Doctor being referred to
- Employee salary cannot be greater than the Doctor's salary that they are practicing under


Office
office_ID varchar
doctor_ID uuid
street_address varchar
city varchar
state varchar
country varchar
zip varchar
Employees
employee_ID uuid
office_ID uuid
doctor_ID uuid?
employee_address varchar
city varchar
state varchar
zip varchar
log_username bigint
employee_title linestring
employee_Name linestring
employee_Authorization boolean
employee_email bigint
employee_DOB bigint
employee_phone bigint
employee_salary bigint
Invoice
invoice_id uuid
employee_id bigint
patient_id bigint
cost float
date_created datetime
payment_method varchar
isPaid boolean
is_insured boolean
insurance_discount float
LOGIN
log_username varchar
log_pass varchar
Appoinment
appoinment_id int
appt_Patient_id int
appt_Doctor_id int
appt_office_id int
ref_id int?
appt_date date
appt_time time
Patient
patient_ID uuid
log_username varchar
patient_address uuid
patient_ref_ID uuid?
patient_prime_doctor uuid?
city varchar
state varchar
zip varchar
patient_first_name varchar
patient_last_name varchar
patient_sex int
patient_email varchar
patient_phone_num text
patient_dob date
patient_diagnosis uuid?
insurance_provider linestring?
insurance_policy_number bigint?
Medical_History
patient_ID uuid
med_h_isAnswered boolean
med_h_smoker boolean
med_h_heart_disease boolean
med_h_diabetes boolean
med_h_current_meds linestring
med_h_cancer boolean
med_h_pregnant boolean
med_h_sexual_active boolean
med_h_ID uuid
Diagnosis
diag_id uuid
patient_ID uuid
doctor_ID uuid
diag_checkup_ID uuid
diag_desc linestring?
Prescription
prescription_id uuid
patient_ID uuid
doctor_ID uuid
pres_name varchar
pres_refills int
med_strength varchar
med_units int
med_NDC varchar
General_Checkup
checkup_ID uuid
patient_id uuid
doctor_id uuid
gc_medical_hist uuid?
gc_blood_test_ID uuid?
gc_Height float?
gc_weight float?
Availability
id uuid
office_id uuid
day_of_week enum
start_time timestamp
end_time timestamp
is_available boolean
Blood_CBC_test
blood_ID uuid
blo_type linestring
blo_RBC float
blo_WBC float
blo_hemoglobin float
blo_Hematocrit_percent float
blo_platelets int
Referrals
ref_ID uuid
speacialist_referred uuid
referring_doc uuid
doctor_specialization varchar
Doctor
doctor_ID varchar
log_username varchar
office_ID uuid
doctor_first_name bigint
doctor_last_name bigint
doctor_address varchar
state varchar
city varchar
zip varchar
doctor_email varchar
doctor_phone_num int
doctor_DOB date
doctor_specialization varchar
salary bigint
drawSQL