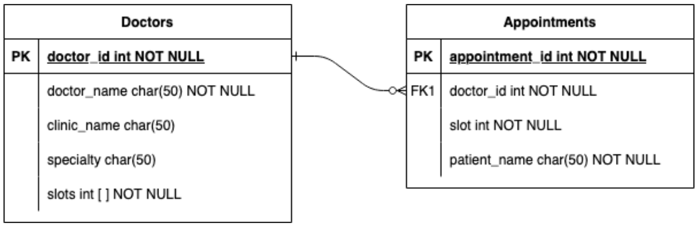
17-625: API Design
Assignment 2.1 - GraphQL

Schema

My design of the API has the following simple schema, represented by an ER diagram -

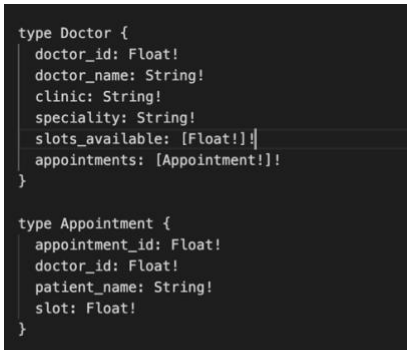


Doctors	
PK	doctor_id int NOT NULL
	doctor_name char(50) NOT NULL
	clinic_name char(50)
	specialty char(50)
	slots int [ ] NOT NULL

Appointments	
PK	appointment_id int NOT NULL
FK1	doctor_id int NOT NULL
	slot int NOT NULL
	patient_name char(50) NOT NULL

Slots is a list of integers from 0 to 15, which are mapped to a timeslot from 9:00 AM to 4:30 PM. For example, slot 2 represents a slot of 10:00 - 10:30 AM.

These are translated to GQL types like -



```
type Doctor {
  doctor_id: Float!
  doctor_name: String!
  clinic: String!
  speciality: String!
  slots_available: [Float!]!
  appointments: [Appointment!]!
}

type Appointment {
  appointment_id: Float!
  doctor_id: Float!
  patient_name: String!
  slot: Float!
}
```

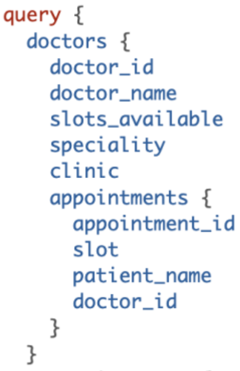
Additionally, we could also have a "*Patients*" table, but I have omitted this detail for now and only have a *patient_name* in the appointment to represent the patient.

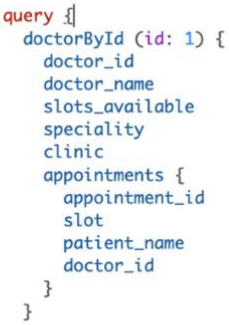
Queries

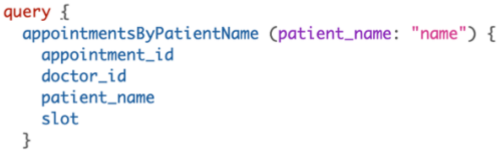
The following queries are supported by my schema -

Query	Input	Output	Description

appointments	query {
	appointments {
	appointment_id
	doctor_id
	patient_name
	slot
	}
	}
		Appointment [ ]	List of all appointments booked

doctors	
	query {
	doctors {
	doctor_id
	doctor_name
	slots_available
	speciality
	clinic
	appointments {
	appointment_id
	slot
	patient_name
	doctor_id
	}
	}
		Doctor [ ]	List of all doctors

doctorById	id: Float!
	
	query {
	doctorById (id: 1) {
	doctor_id
	doctor_name
	slots_available
	speciality
	clinic
	appointments {
	appointment_id
	slot
	patient_name
	doctor_id
	}
	}
		Doctor (with fields queried like here)	Get a doctor by their Id. This will give info like name, clinic, specialty, list of available slots and appointments.

appointmentsByPatientName	patient_name: String!
	
	query {
	appointmentsByPatientName (patient_name: "name") {
	appointment_id
	doctor_id
	patient_name
	slot
	}
		Appointment [ ]	List of appointments for a patient, fetched by their name, containing the info about that appointment

Mutations

The following mutations are supported by my schema -

Mutation	Input	Output	Description

createAppointment	{
	doctor_id: Float!
	patient_name: String!
	slot: Float!
	}
		Appointment	Creates an appointment with a doctor for a patient on a slot

updatePatientNameOnAppointment	{
	doctor_id: Float!
	patient_name: String!
	slot: Float!
	}
		Appointment	Updates the patient's name for an appointment with a particular doctor at a particular slot. Returns the modified appointment

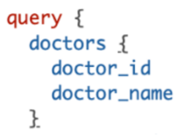
deleteAppointment	{
	doctor_id: Float!
	slot: Float!
	}
		Slot: number	Deletes an appointment slot for a particular doctor, and returns that slot number

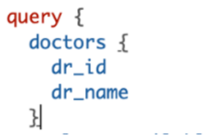
Endpoint - http://localhost:8000/graphql

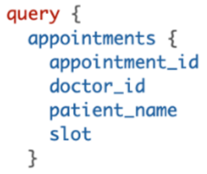
Testcases

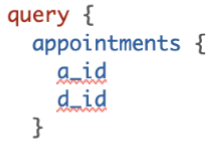
For each query and mutation, there is a happy path and an error path.

Id	Testcase Description	Inputs	Expected Output	Remarks
1	Get doctor by id, with the id existing valid	Id = 1	A valid *Doctor* object	
2	Get doctor by id, with the id not existing	Id = -1	Error with a useful message	Message suggests that doctor doesn't exist

3	Get a list of doctors with valid fields queried	
		query {
		doctors {
		doctor_id
		doctor_name
		}
			A list of valid doctor objects with those keys	

4	Get a list of doctors with an invalid field queried	
		query {
		doctors {
		dr_id
		dr_name
		}|
			Error with a useful message	Message suggests that the fields are invalid

5	Get a list of appointments with valid fields queried	
		query {
		appointments {
		appointment_id
		doctor_id
		patient_name
		slot
		}
			A list of valid appointment objects with those keys	

6	Get a list of appointments with invalid fields queried	
		query {
		appointments {
		a_id
		d_id
		}
			Error with a useful message	Message suggests that the fields are invalid

7	Get appointments for an existing *patient_name*	patient_name = “valid name”	Error with a useful message	Message suggests that8patient doesn’t exist
8	Get appointments for a non-existing *patient_name*	patient_name = “12435”	Error with a useful message	Message suggests that patient doesn’t exist

9	Create appointment with all fields valid	{
		doctor_id: 1
		patient_name: “Name”
		slot: 12
		}
			{
			doctor_id: 1
			appointment_id: 1
			patient_name: “Name”
			slot: 12
			}
				Returns the created appointment, with all the fields from the input

10	Create appointment with invalid slot	{
		doctor_id: 1
		patient_name: “Name”
		slot: -1
		}
			Error with a useful message	Message suggests that slot is invalid

11	Update patient name on an existing appointment slot with and existing doctor	{
		doctor_id: 1
		patient_name: “New”
		slot: 12
		}
			{
			doctor_id: 1
			appointment_id: 1
			patient_name: “New”
			slot: 12
			}
				Returns the created appointment, with all the fields from the input

12	Update patient name on an existing doctor's appointment, but with slot missing	{
		doctor_id: 1
		patient_name: "Name"
		}
			Error with a useful message	Message suggests that slot is missing

13	Delete appointment for a valid doctor at a valid slot	{
		doctor_id: 1
		slot: 12
		}
			12	

14	Delete appointment for a valid doctor at an invalid slot	{
		doctor_id: 1
		slot: -1
		}
			Error with a useful message	Message suggests that slot is invalid