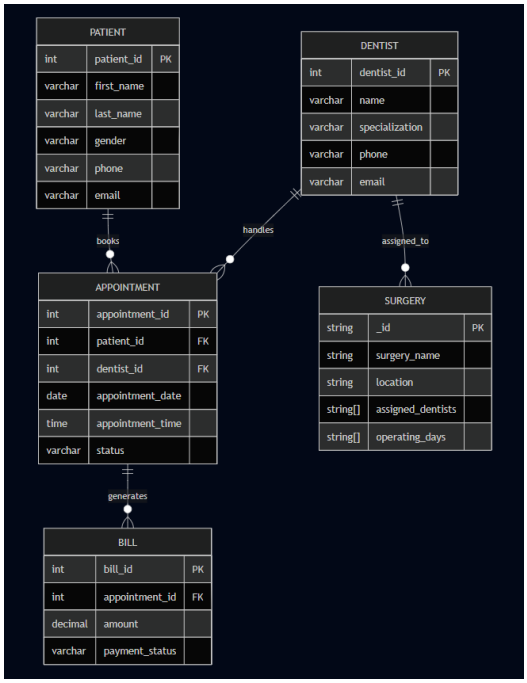
Lab 5: Data modeling, Database Design and Implementation

E–R Diagram (Entity–Relationship Model)

Entity	Attributes	Primary Key	Foreign Keys
Dentist	dentist_id, first_name, last_name, phone, email, specialization	dentist_id	—
Patient	patient_id, first_name, last_name, phone, email, address, dob	patient_id	—
Surgery	surgery_id, name, address, phone	surgery_id	—
Appointment	appointment_id, appointment_date, appointment_time, dentist_id, patient_id, surgery_id	appointment_id	dentist_id → Dentistpatient_id → Patientsurgery_id → Surgery
Bill	bill_id, appointment_id, amount, status	bill_id	appointment_id → Appointment

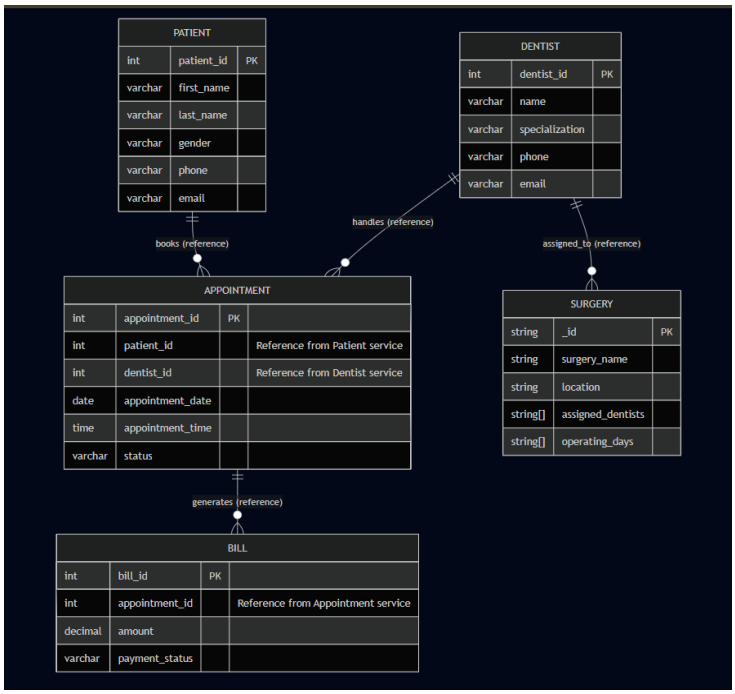
For Monolithic with single DB we shall use the ER



Relationships

- One **Dentist** → Many **Appointments**
- One **Patient** → Many **Appointments**
- One **Surgery** → Many **Appointments**
- One **Appointment** → One **Bill**
- One **Patient** → Many **Bills**

But since my architecture is based on Microservice Below is my ER.



Create my DBs in docker.

docker run -d --name dentist-db -e POSTGRES_DB=dentistdb -e POSTGRES_USER=postgres -e POSTGRES_PASSWORD=postgres -p 5433:5432 postgres:15

docker run -d --name patient-db -e POSTGRES_DB=patientdb -e POSTGRES_USER=postgres -e POSTGRES_PASSWORD=postgres -p 5432:5432 postgres:15

docker run -d --name appointment-db -e POSTGRES_DB=appointmentdb -e POSTGRES_USER=postgres -e POSTGRES_PASSWORD=postgres -p 5434:5432 postgres:15

docker run -d --name billing-db -e POSTGRES_DB=billingdb -e POSTGRES_USER=postgres -e POSTGRES_PASSWORD=postgres -p 5435:5432 postgres:15

docker run -d --name surgery-db -e MONGO_INITDB_DATABASE=surgerydb -p 27017:27017 mongo:8.0

```
CONTAINER ID   IMAGE                              COMMAND                  CREATED          STATUS          PORTS                                            NAMES
735010f69542   mongo:7                            "docker-entrypoint.s…"   8 minutes ago    Up 7 minutes    0.0.0.0:27017->27017/tcp                         surgery-db
d45ebef95831   postgres:15                        "docker-entrypoint.s…"   9 minutes ago    Up 9 minutes    0.0.0.0:5435->5432/tcp                           billing-db
0375ad47fabb   postgres:15                        "docker-entrypoint.s…"   9 minutes ago    Up 9 minutes    0.0.0.0:5434->5432/tcp                           appointment-db
5d06ea487da1   postgres:15                        "docker-entrypoint.s…"   7 hours ago      Up 7 hours      0.0.0.0:5433->5432/tcp                           dentist-db
216e8ef12d8f   apache/kafka:3.7.0                 "/__cacert_entrypoin…"   8 hours ago      Up 8 hours      0.0.0.0:9092->9092/tcp                           kafka-kraft
b90364d67340   confluentinc/cp-zookeeper:7.5.0    "/etc/confluent/dock…"   9 hours ago      Up 9 hours      2888/tcp, 0.0.0.0:2181->2181/tcp, 3888/tcp       zookeeper
d4c68f5626c6   postgres:15                        "docker-entrypoint.s…"   9 hours ago      Up 9 hours      0.0.0.0:5432->5432/tcp                           patient-db
PS D:\MIU-2024\ASD\Labs\ASD-Labs>
```

Surgery Microservice (MongoDB)

The Surgery service uses a document-oriented model implemented in MongoDB to store data about surgical rooms and schedules. Each Surgery document contains an _id, surgery_name, location, and embedded arrays for assigned_dentists and operating_days. This flexible schema supports dynamic relationships without requiring strict foreign key constraints. Dentist references are stored as IDs that correspond to records in the Dentist PostgreSQL service, allowing polyglot persistence across microservices.