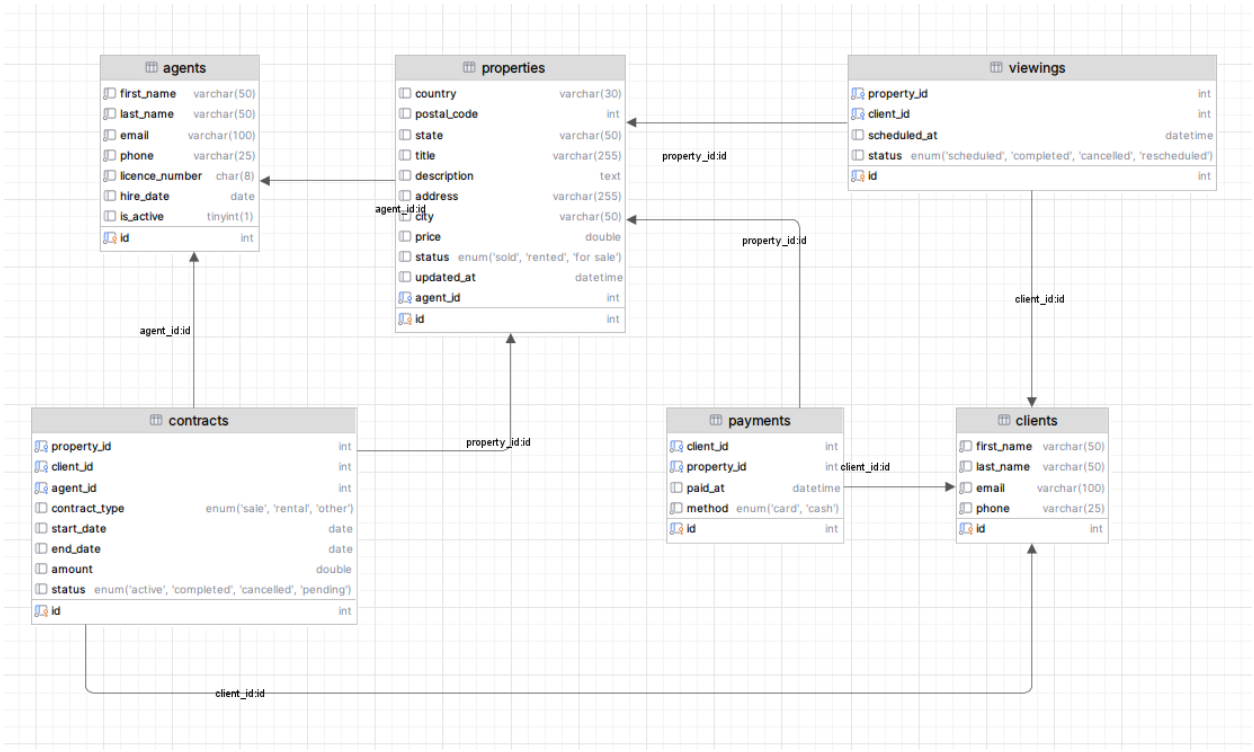
04. Database Design - SOLUTIONS

Task 1

1. A real estate agency is facing significant challenges in managing its extensive portfolio of properties, client interactions, and administrative tasks. As the agency expands its operations and clientele, the existing manual systems are proving to be inadequate for efficiently handling the increasing workload and complexities of the real estate market.

   The agency is in need of a sophisticated database system tailored to the real estate industry that can seamlessly integrate all aspects of its operations. This system must offer a user-friendly interface for agents, clients, and administrative staff, while providing robust features to streamline property management, client interactions, and administrative processes.

   Tables:
   agents - Stores information about real estate agents who manage properties and interact with clients.
   properties - Contains all information about the properties that the agency manages or sells.
   clients - Stores information about clients who buy, rent, or view properties.
   viewings - Tracks scheduled property viewings between clients and agents.
   payments - Manages all payment transactions related to contracts.
   contracts - Stores records of finalized deals (property sales or rentals) between the agency, clients, and agents.

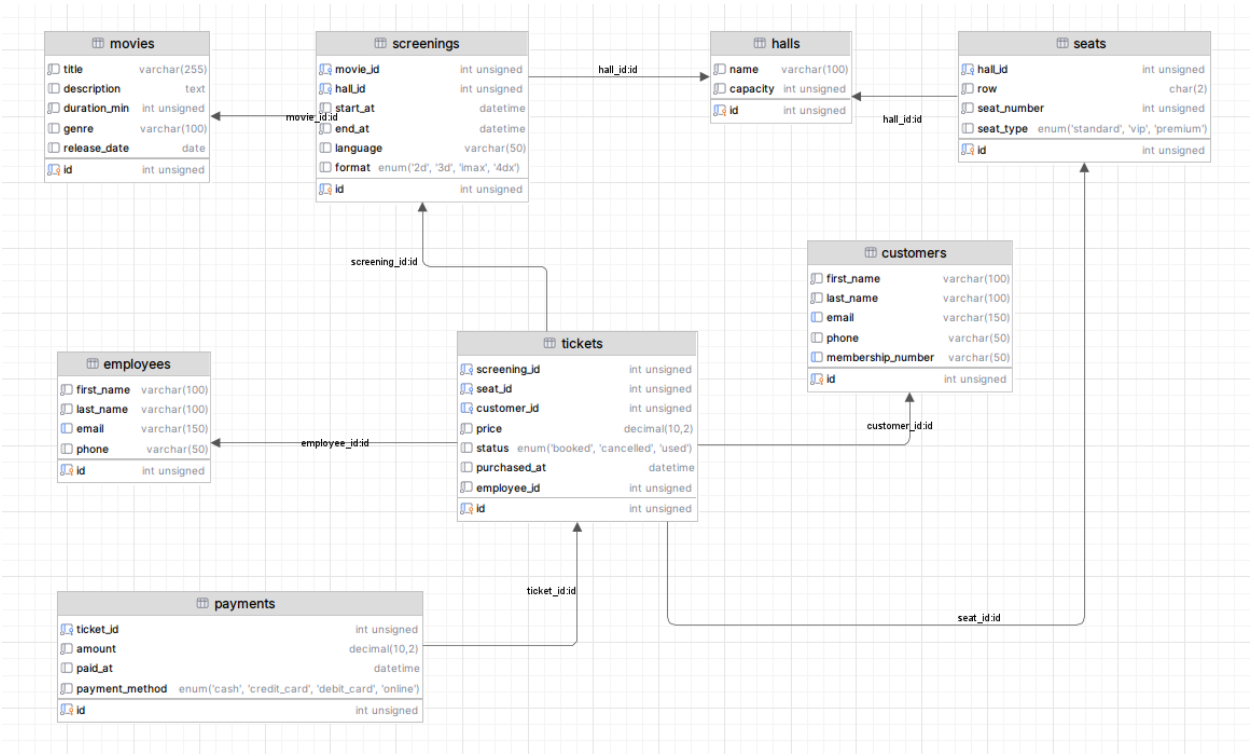

agents
first_name varchar(50)
last_name varchar(50)
email varchar(100)
phone varchar(25)
licence_number char(8)
hire_date date
is_active tinyint(1)
id int
properties
country varchar(30)
postal_code int
state varchar(50)
title varchar(255)
description text
address varchar(255)
city varchar(50)
price double
status enum('sold', 'rented', 'for sale')
updated_at datetime
agent_id int
id int
viewings
property_id int
client_id int
scheduled_at datetime
status enum('scheduled', 'completed', 'cancelled', 'rescheduled')
id int
contracts
property_id int
client_id int
agent_id int
contract_type enum('sale', 'rental', 'other')
start_date date
end_date date
amount double
status enum('active', 'completed', 'cancelled', 'pending')
id int
payments
client_id int
property_id int
paid_at datetime
method enum('card', 'cash')
id int
clients
first_name varchar(50)
last_name varchar(50)
email varchar(100)
phone varchar(25)
id int
agent_id:id
property_id:id
property_id:id
property_id:id
client_id:id
client_id:id
client_id:id

Task 2

2. A well-known cinema company is struggling to keep up with the growing intricacy of managing its administrative responsibilities, ticket sales, movie screenings, and client interactions. As the demand for online ticket booking grows and the number of movie releases escalates, the existing manual systems are struggling to keep up with the pace of modern cinema operations.

   The cinema company wants to put in place a strong database system designed specifically for the entertainment sector that can integrate every aspect of its business operations with simplicity. In addition to providing sophisticated capabilities to speed up ticket booking, manage movie schedules, track customer needs, and improve communication within the cinema community, this system should have an easy-to-use interface for consumers, employees, and administrators.

   Tables:
   customers - stores information about cinema customers, including their personal details and contact information for ticket bookings and communication.
   employees - contains data about cinema staff members, such as their names, roles, and contact details, to manage administrative and operational tasks.
   movies - holds details about movies being shown, including title, genre, duration, release date, and description.
   screenings - records scheduled movie showtimes, linking movies with specific halls, dates, and times.
   halls - defines the different cinema halls where movies are screened, including hall name, capacity, and location within the cinema.
   seats - represents individual seats within each hall, including seat number, row, and hall association, used for seat selection during booking.
   tickets - tracks issued tickets for screenings, linking customers, seats, and screenings, along with ticket type and price information.
   payments - stores details of payments made for tickets, including amount, payment date, and method, associated with a specific customer or ticket.

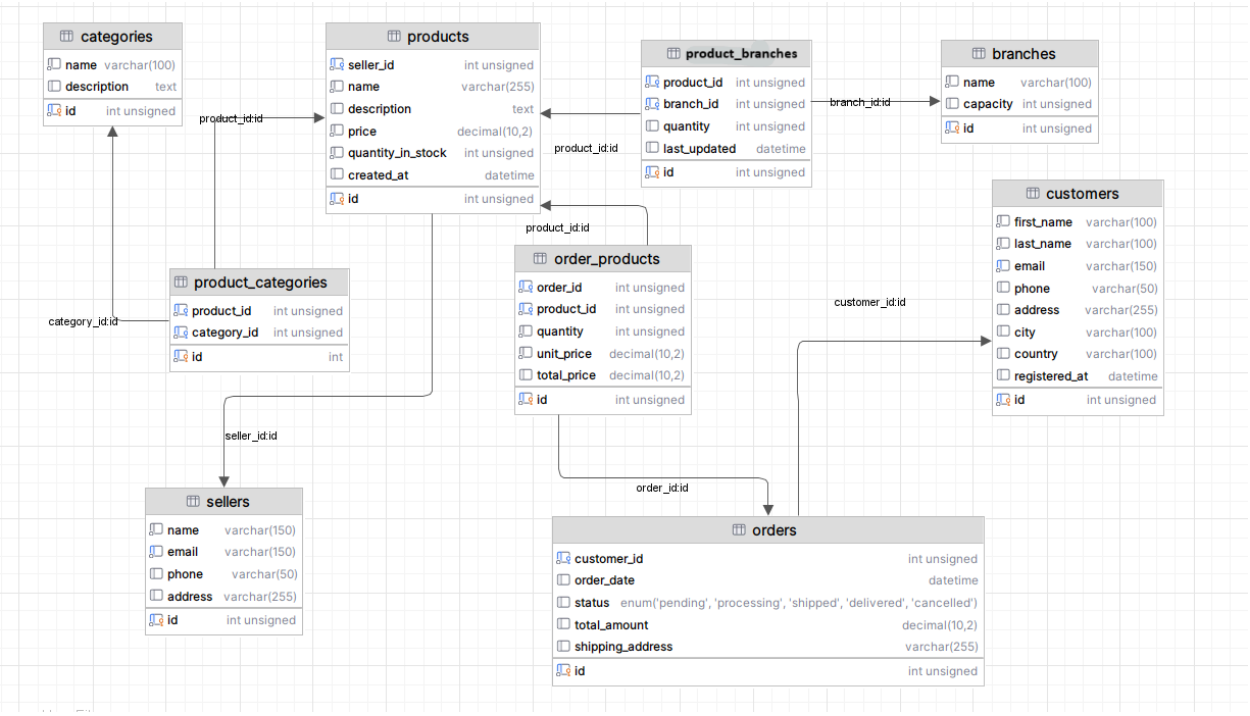

movies
title varchar(255)
description text
duration_min int unsigned
genre varchar(100)
release_date date
id int unsigned
screenings
movie_id int unsigned
hall_id int unsigned
start_at datetime
end_at datetime
language varchar(50)
format enum('2d', '3d', 'imax', '4dx')
id int unsigned
halls
name varchar(100)
capacity int unsigned
id int unsigned
seats
hall_id int unsigned
row char(2)
seat_number int unsigned
seat_type enum('standard', 'vip', 'premium')
id int unsigned
customers
first_name varchar(100)
last_name varchar(100)
email varchar(150)
phone varchar(50)
membership_number varchar(50)
id int unsigned
employees
first_name varchar(100)
last_name varchar(100)
email varchar(150)
phone varchar(50)
id int unsigned
tickets
screening_id int unsigned
seat_id int unsigned
customer_id int unsigned
price decimal(10,2)
status enum('booked', 'cancelled', 'used')
purchased_at datetime
employee_id int unsigned
id int unsigned
payments
ticket_id int unsigned
amount decimal(10,2)
paid_at datetime
payment_method enum('cash', 'credit_card', 'debit_card', 'online')
id int unsigned
hall_id:id
movie_id:id
hall_id:id
screening_id:id
customer_id:id
employee_id:id
ticket_id:id
seat_id:id

Task 3

3. An e-commerce platform is facing significant challenges in effectively managing its inventory, product listings, order fulfillment, and supply chain operations. As the platform grows its product catalog and customer base, the existing manual inventory management systems struggle to keep pace with the increasing complexity and volume of transactions.

   The e-commerce platform is in need of a comprehensive database system tailored to inventory management that can seamlessly integrate all aspects of its operations. Along with having powerful tools to optimize product listings, expedite order processing, and streamline inventory tracking, this system must also have an intuitive user interface for sellers, warehouse employees, and administrators.

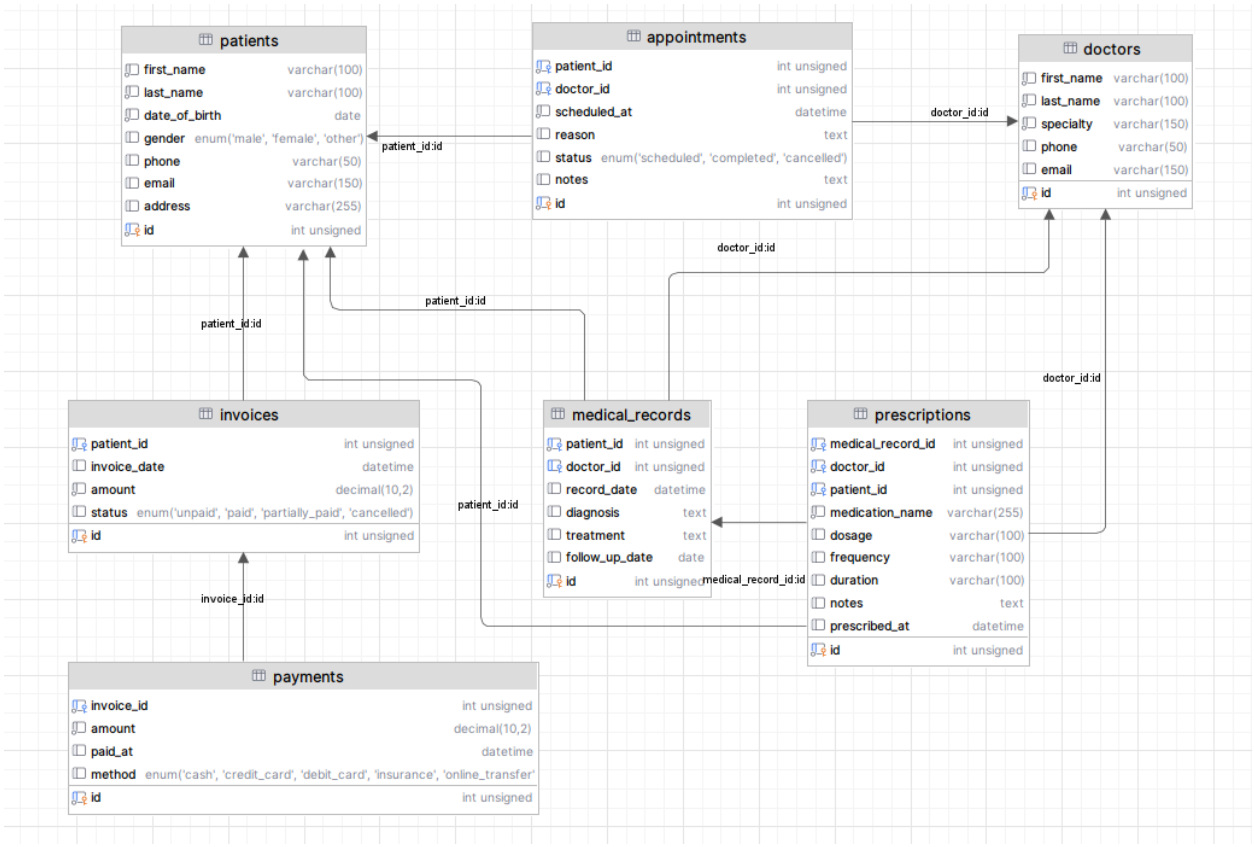
   Tables:
   customers - Stores information about registered platform users who purchase products.
   sellers - Stores information about the businesses or individuals selling products on the platform.
   products - Contains details about individual items available for sale, such as name, price, and description.
   categories - Defines different groups or classifications used to organize the products.
   product_categories - Links products to their respective categories (many-to-many relationship).
   orders - Records details of customer purchases, including date, total amount, and status.
   order_products - Details the specific products and quantities included in each order (links orders and products).
   branches - Stores information about various operational locations, such as physical stores or warehouses.
   product_branches - Tracks the inventory level of specific products at each operational branch.



Task 4

4. A health care clinic is struggling with managing patient records, appointment scheduling, billing, and medical inventory. With the increase in patient volume and the need for accurate medical records, the existing manual systems are becoming inefficient and prone to errors. The clinic requires a sophisticated database system designed specifically for healthcare that integrates patient management, appointment scheduling, billing, and inventory control. The system should offer an intuitive interface for medical staff, patients, and administrative personnel while ensuring compliance with healthcare regulations.

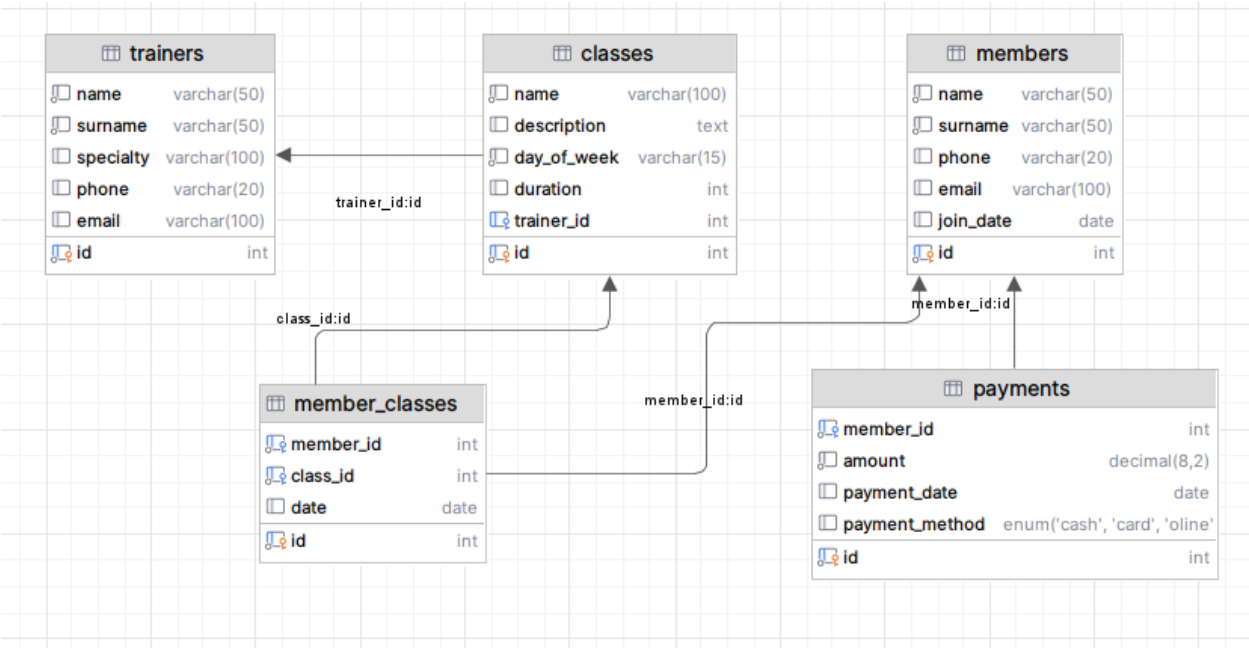
   Tables:
   patients - Stores information for individuals receiving care at the clinic.
   doctors - Contains details about the doctors working at the clinic.
   appointments - Records scheduled visits between patients and doctors.
   invoices - Stores details of charges for services and treatments provided to patients.
   payments - Tracks and records transactions made by patients against their outstanding invoices.
   medical_records - Contains a comprehensive history of a patient's health, diagnoses, treatments, and visit notes.
   prescriptions - Records details of medications ordered by doctors for patients.



Task 5

5. A Fitness Center is encountering difficulties in managing membership information, class schedules, personal training sessions, and billing. As membership numbers rise and the demand for various fitness classes increases, the existing manual systems are proving inadequate. The fitness center needs a robust database system tailored for gym management that integrates member profiles, class schedules, trainer availability, and payment processing. It should feature a user-friendly interface for members, trainers, and administrative staff.

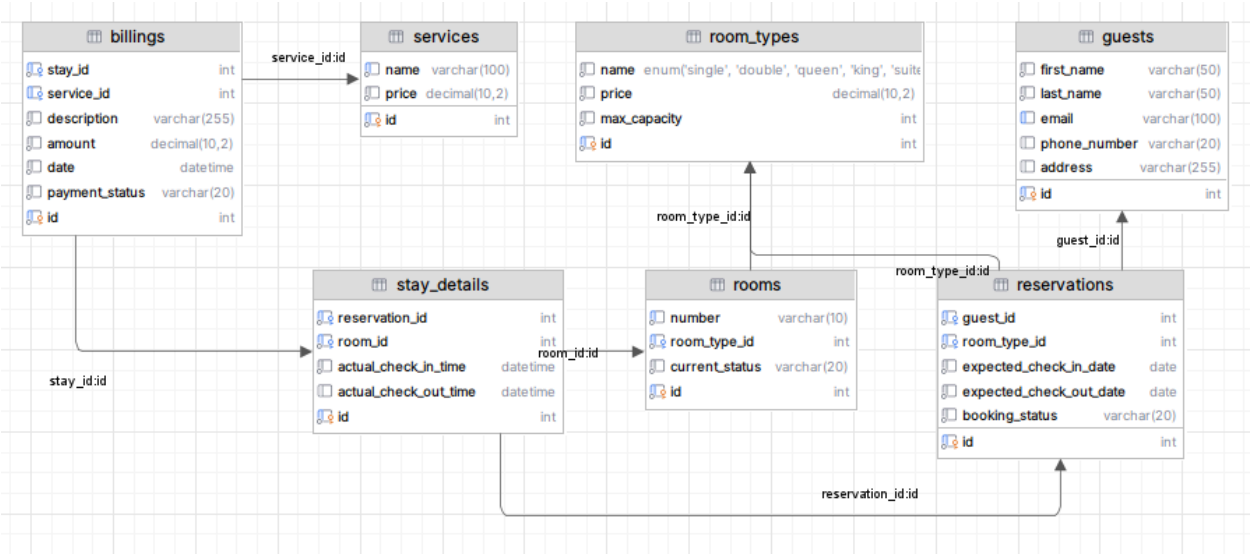
   Tables:
   trainers - Stores information about the fitness instructors employed by the center.
   members - Contains details for individuals registered at the fitness center.
   classes - Defines scheduled group fitness activities
   member_classes - Links members to the specific classes they have registered for or attended.
   payments - Tracks and records membership transactions made by members.



Task 6

6. A Hotel Chain is facing challenges in handling reservations, guest check-ins and check-outs, room assignments, and billing. With the increase in bookings and the need for efficient room management, the current manual processes are becoming cumbersome. The hotel chain needs a comprehensive database system designed for hospitality management that integrates reservations, room management, guest services, and billing. The system should provide a seamless experience for guests, hotel staff, and administrative teams.

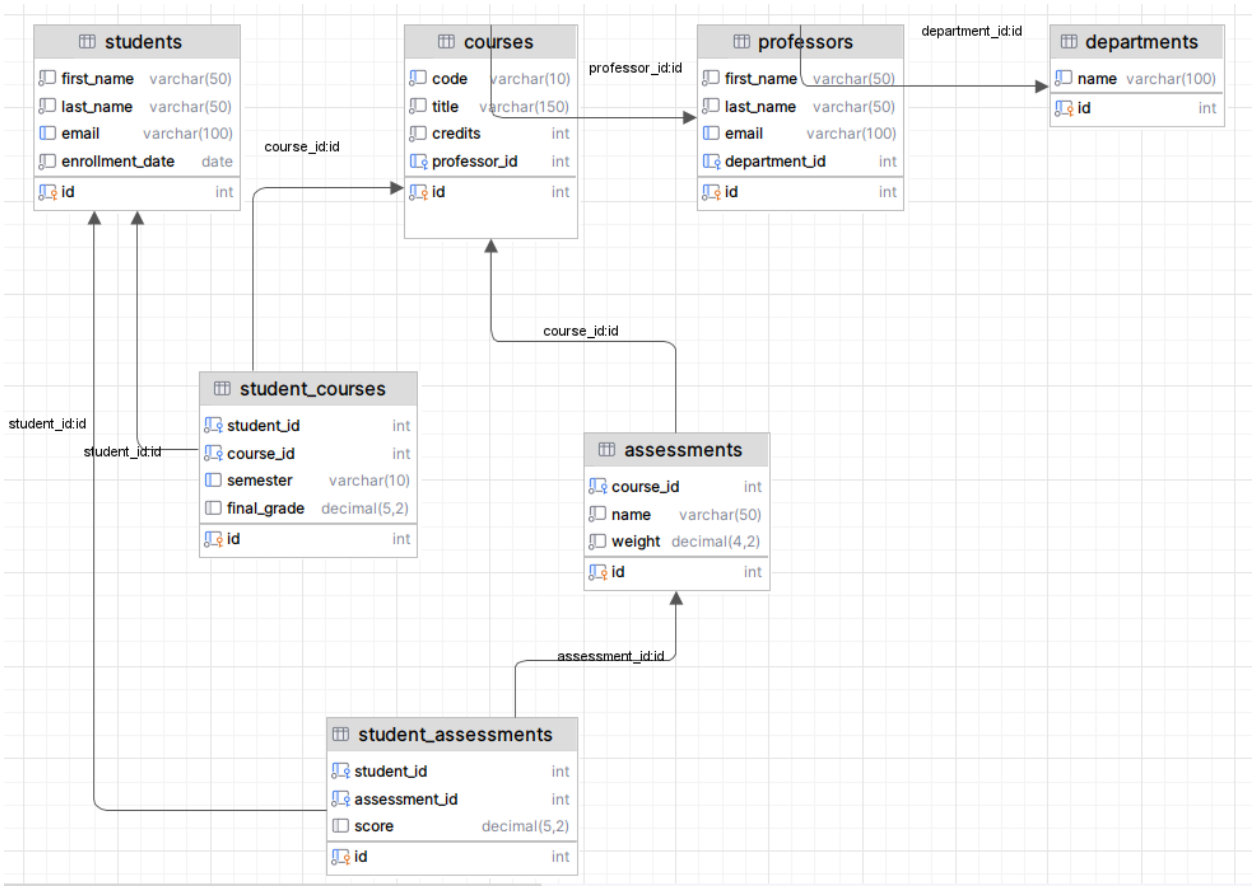
   Tables:
   guests - Stores personal and contact information for all hotel guests.
   services - Lists extra paid amenities available (e.g., parking, room service, laundry) with their prices.
   billing - Records charges for a guest's stay
   reservations - Records booking details made by guests
   stay_details - Links a specific reservation to the actual room assigned and records the actual check-in and check-out times.
   room_types - Defines different categories of rooms (e.g., Single, Double, Suite) and their standard price.
   rooms - Contains details about each physical room



Task 7

7. A University is struggling with managing student records, course registrations, faculty assignments, and grading. As the number of students and courses grows, the existing manual systems are proving inefficient and error-prone. The university needs a sophisticated database system tailored to educational institutions that integrates student information, course management, faculty scheduling, and grading. The system should offer an intuitive interface for students, faculty, and administrative staff

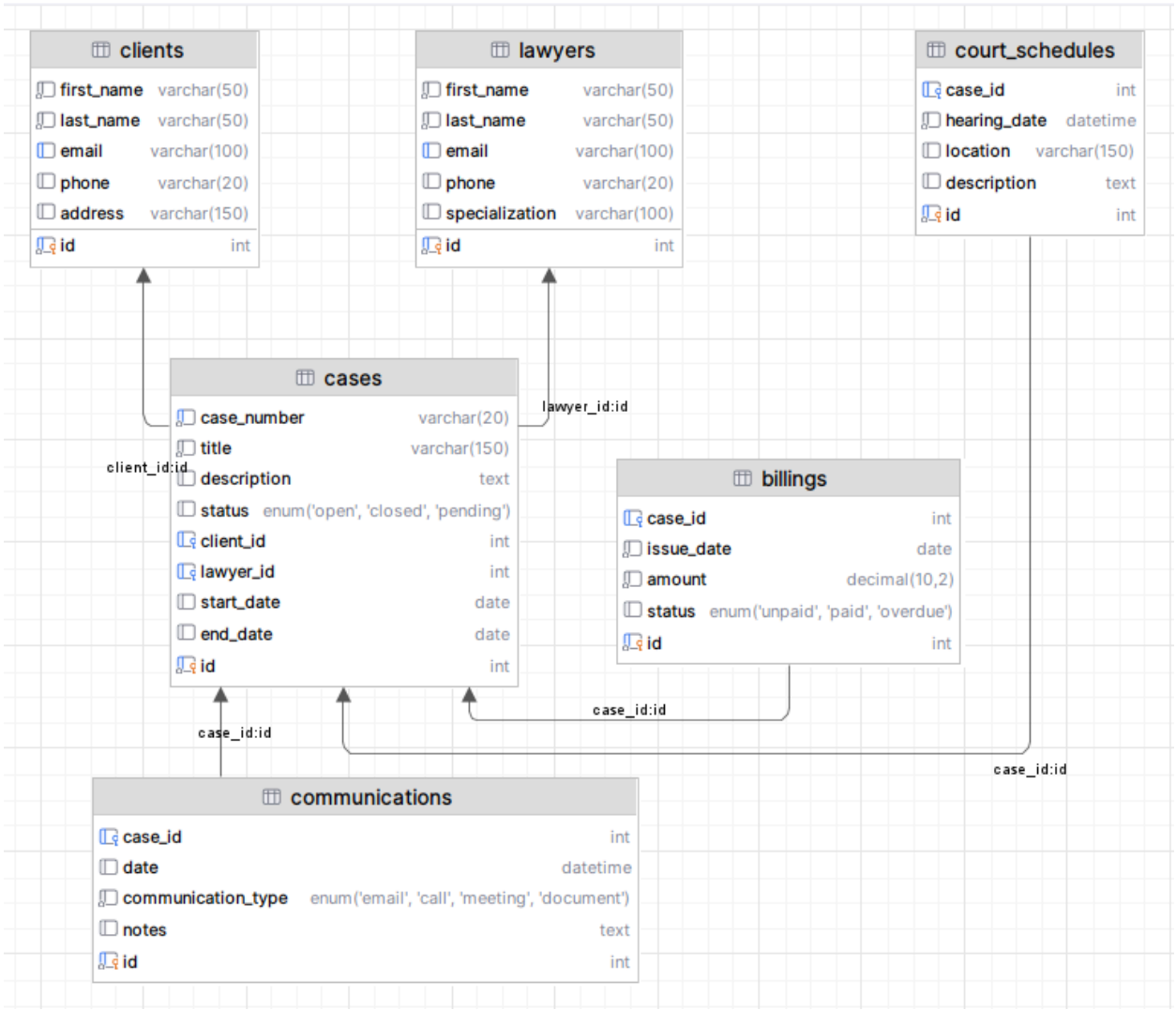
   Tables:
   students - Stores personal and enrollment information for all students
   professors - Contains details about professors
   departments - Classifies departments.
   courses - Defines all academic subjects offered
   student_courses - Links students to the specific course sections they are registered in for a given term.
   assessments - Defines various types of evaluations used in a course, such as exams, quizzes, or assignments.
   student_assessments - Records the specific score or result a student achieved on a particular assessment.



Task 8

8. A Law Firm is finding it challenging to manage case files, client communications, billing, and court schedules. With an increasing number of cases and clients, the current manual systems are becoming overwhelmed. The law firm needs a robust database system designed for legal practices that integrates case management, client interactions, billing, and scheduling. The system should feature an easy-to-use interface for attorneys, clients, and administrative staff.

   Tables:
   clients - Stores personal and contact information for individuals
   lawyers - Contains details about the lawyers employed by the firm.
   cases - Defines the specific legal matters being handled.
   communications - Logs all interactions (emails, calls, meetings) between the firm and clients or external parties related to a case.
   court_schedules - Records upcoming dates, times, and locations for court appearances or hearings related to various cases.
   billings - Tracks all time entries, expenses, and fees incurred for a case, used to generate client invoices.


clients
first_name varchar(50)
last_name varchar(50)
email varchar(100)
phone varchar(20)
address varchar(150)
id int
lawyers
first_name varchar(50)
last_name varchar(50)
email varchar(100)
phone varchar(20)
specialization varchar(100)
id int
court_schedules
case_id int
hearing_date datetime
location varchar(150)
description text
id int
cases
case_number varchar(20)
title varchar(150)
description text
status enum('open', 'closed', 'pending')
client_id int
lawyer_id int
start_date date
end_date date
id int
lawyer_id:id
client_id:id
billings
case_id int
issue_date date
amount decimal(10,2)
status enum('unpaid', 'paid', 'overdue')
id int
case_id:id
case_id:id
case_id:id
communications
case_id int
date datetime
communication_type enum('email', 'call', 'meeting', 'document')
notes text
id int