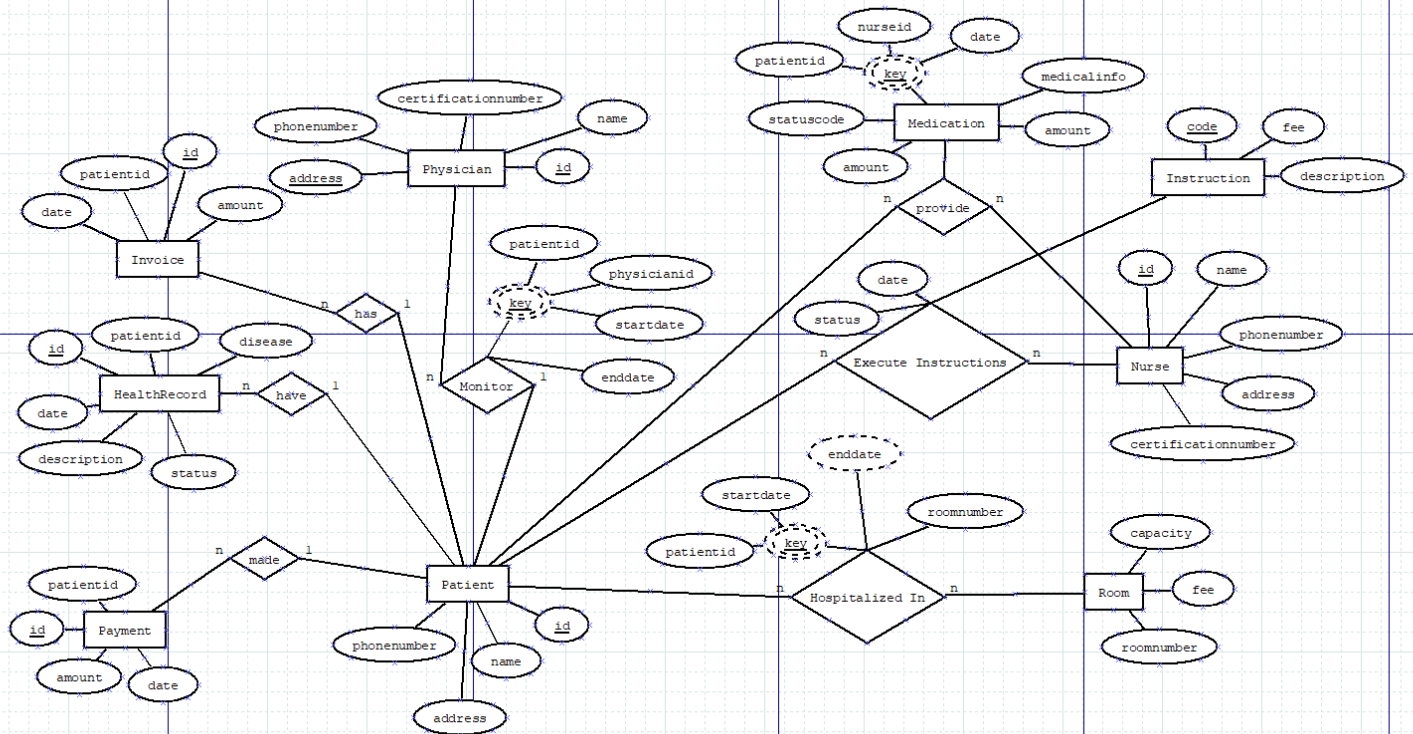


# Entity Relationship Diagram



**Assumptions / Changes from original instruction:**

### Physician and Nurse Tables:

Assumption: Both physicians and nurses have similar attributes like name, phone number, address, and certification number.

Change: Explicit fields for each attribute are created.

### Patient Table:

Assumption: Patients are identified by unique IDs, and their names, addresses, and phone numbers are essential attributes.

Change: A straightforward table is created with these fields.

### Health Record Table:

Assumption: Each health record is unique and linked to a specific patient. It includes details about disease, date, status, and a description.

Change: A table linking health records to patients via patient IDs is designed.

**Room Table:**

Assumption: Each room in the hospital has a unique number, a capacity, and a fee per night.

Change: A table for rooms is created with these specific fields.

**Hospitalized Table:**

Assumption: Patients are linked to specific rooms and have defined stay durations.

Change: A composite primary key comprising patient ID and start date is used to uniquely identify each hospitalization record.

**Monitored Table:**

Assumption: Each patient is monitored by physicians for specific durations.

Change: A table to capture this relationship with timestamps for start and end times is created.

**Medication Table:**

Assumption: Nurses administer medications to patients, which are tracked daily.

Change: A table to record this activity, including the nurse ID, patient ID, date, medication information, status code, and amount is established.

**Payment and Invoice Tables:**

Assumption: The hospital records all payable services and issues invoices.

Change: Separate tables for payments and invoices linked to patients are designed.

**Instructions Table:**

Assumption: Instructions from physicians have unique codes, fees, and descriptions.

Change: A table specifically for instructions is created.

**Executed Instructions Table:**

Assumption: Nurses execute instructions for patients, which need to be tracked.

Change: A table linking nurses, patients, and instructions is established.

**Foreign Key Constraints:**

Assumption: Ensuring data integrity and the relational aspect of the database.

Change: Foreign key constraints are added to link various entities like patients, physicians, nurses, and rooms.

**Unique IDs and Primary Keys:**

Assumption: Each entity (like a patient, physician, nurse) has a unique identifier.

Change: Primary keys are used in tables to enforce this uniqueness.

## Set of Relationships

```
drop database if exists hospital;
create database hospital;
use hospital;

create table physician (
  id int primary key,
  name varchar(255),
  phonenumber varchar(255),
  address varchar(255),
  certificationnumber varchar(255),
  fieldofexpertise varchar(255)
);

create table nurse (
  id int primary key,
  name varchar(255),
  phonenumber varchar(255),
  address varchar(255),
  certificationnumber varchar(255)
);

create table patient (
  id int primary key,
  name varchar(255),
  address varchar(255),
  phonenumber varchar(255)
);

create table healthrecord (
  id int primary key,
  patientid int,
  disease varchar(255),
  date date,
  status varchar(255),
  description text
);

create table room (
  roomnumber int primary key,
  capacity int,
  feepernight decimal(10, 2)
);

create table hospitalized (
  patientid int,
  roomnumber int,
  startdate date,
```

```

    enddate date,
    primary key (patientid, startdate)
);

create table monitored (
    physicianid int,
    patientid int,
    starttime timestamp,
    endtime timestamp,
    primary key (physicianid, patientid, starttime)
);

create table medication (
    patientid int,
    nurseid int,
    date date,
    medicationinfo text,
    statuscode varchar(255),
    amount int,
    primary key (patientid, nurseid, date)
);

create table payment (
    paymentid int primary key,
    patientid int,
    amount decimal(10, 2),
    date date
);

create table instructions (
    code int primary key,
    fee decimal(10, 2),
    description text
);

create table invoice (
    invoice_id int primary key,
    patient_id int,
    issue_date date,
    total_amount decimal(10,2),
    foreign key (patient_id) references patient(id)
);

create table executed_instructions (
    nurse_id int,
    patient_id int,
    instruction_code int,
    execution_date date,
    status_code varchar(255),
    primary key (nurse_id, patient_id, instruction_code),

```

```
foreign key (nurse_id) references nurse(id),
foreign key (patient_id) references patient(id),
foreign key (instruction_code) references instructions(code)
);
```

```
alter table healthrecord
add constraint fk_healthrecord_patient
foreign key (patientid) references patient(id);
```

```
alter table hospitalized
add constraint fk_hospitalized_patient
foreign key (patientid) references patient(id),
add constraint fk_hospitalized_room
foreign key (roomnumber) references room(roomnumber);
```

```
alter table monitored
add constraint fk_monitored_physician
foreign key (physicianid) references physician(id),
add constraint fk_monitored_patient
foreign key (patientid) references patient(id);
```

```
alter table medication
add constraint fk_medication_patient
foreign key (patientid) references patient(id),
add constraint fk_medication_nurse
foreign key (nurseid) references nurse(id);
```

```
alter table payment
add constraint fk_payment_patient
foreign key (patientid) references patient(id);
```

## Queries

```
# list of patients with their assigned physicians
select patient.name as patient_name, physician.name as physician_name
from patient
join monitored on patient.id = monitored.patientid
join physician on monitored.physicianid = physician.id;
```

```
# details of nurses and the medications they administered
select nurse.name as nurse_name, medication.medicationinfo
from nurse
join medication on nurse.id = medication.nurseid;
```

```

# patients and their room details during hospitalization:
select patient.name as patient_name, room.roomnumber, hospitalized.startdate,
hospitalized.enddate
from patient
join hospitalized on patient.id = hospitalized.patientid
join room on hospitalized.roomnumber = room.roomnumber;

# count of patients per physician
select physician.id as physician_id, physician.name as physician_name,
count(monitored.patientid) as patient_count
from physician
join monitored on physician.id = monitored.physicianid
group by physician.id;

# total fees collected per room
select room.roomnumber, sum(payment.amount) as total_fees
from payment
join hospitalized on payment.patientid = hospitalized.patientid
join room on hospitalized.roomnumber = room.roomnumber
group by room.roomnumber;

# average medication amount prescribed by nurses:
select nurse.id as nurse_id, nurse.name as nurse_name, avg(medication.amount) as
average_medication_amount
from nurse
join medication on nurse.id = medication.nurseid
group by nurse.id;

# physicians treating patients with a specific disease
select physician.name
from physician
where exists (
  select 1
  from monitored
  join healthrecord on monitored.patientid = healthrecord.patientid
  where healthrecord.disease = 'diabetes' and monitored.physicianid = physician.id # replace
diabetes with whatever you're searching for
);

# patients who have never been hospitalized
select patient.name
from patient
where not exists (
  select 1
  from hospitalized
  where hospitalized.patientid = patient.id
);

# nurses who have administered a certain medication:
select nurse.name

```

```

from nurse
where exists (
  select 1
  from medication
  where medication.nurseid = nurse.id and medication.medicationinfo = 'ibuprofen' # replace
  ibuprofen with whatever you're searching for
);

# list all patients with their latest health record
select patient.name, healthrecord.disease, healthrecord.date
from patient
join healthrecord on patient.id = healthrecord.patientid
where healthrecord.date = (
  select max(healthrecord2.date)
  from healthrecord healthrecord2
  where healthrecord2.patientid = patient.id
);

# total amount billed to each patient
select invoice.patient_id, sum(invoice.total_amount) as total_billed
from invoice
group by invoice.patient_id;

# physicians with more than 3 patients:
select physician.name
from physician
join monitored on physician.id = monitored.physicianid
group by physician.id
having count(distinct monitored.patientid) > 3;

# patients who have been hospitalized more than once
select patient.name
from patient
join hospitalized on patient.id = hospitalized.patientid
group by patient.id
having count(hospitalized.startdate) > 1;

# most common disease among patients:
select disease, count(*) as count
from healthrecord
group by disease
order by count(*) desc
limit 1;

# rooms that have never been used:
select room.roomnumber
from room
where not exists (
  select 1

```

```

from hospitalized
where hospitalized.roomnumber = room.roomnumber
);

```

## Results

# list of patients with their assigned physicians

patient_name	physician_name
Alice Green	Dr. John Smith
Bob White	Dr. Emily Johnson
Charlie Black	Dr. William Brown
Diana Gray	Dr. Ava Davis
Ethan Stone	Dr. James Wilson

# details of nurses and the medications they administered

nurse_name	medicationinfo
Nurse Olivia Martin	Paracetamol
Nurse Liam Anderson	Ibuprofen
Nurse Sophia Thompson	Amlodipine
Nurse Lucas Garcia	Salbutamol
Nurse Mia Harris	Metformin

# patients and their room details during hospitalization

patient_name	roomnumber	startdate	enddate
Alice Green	101	2023-01-10	2023-01-15
Bob White	102	2023-02-15	2023-02-25
Charlie Black	103	2023-03-20	2023-03-30
Diana Gray	104	2023-04-25	2023-05-05
Ethan Stone	105	2023-05-30	2023-06-10

# count of patients per physician

physician_id	physician_name	patient_count
1	Dr. John Smith	1
2	Dr. Emily Johnson	1
3	Dr. William Brown	1
4	Dr. Ava Davis	1
5	Dr. James Wilson	1



# total fees collected per room

	roomnumber	total_fees
▶	101	750.00
	102	2000.00
	103	3000.00
	104	2500.00
	105	1000.00

# average medication amount prescribed by nurses:

	nurse_id	nurse_name	average_medication_amount
▶	1	Nurse Olivia Martin	2.0000
	2	Nurse Liam Anderson	1.0000
	3	Nurse Sophia Thompson	1.0000
	4	Nurse Lucas Garcia	1.0000
	5	Nurse Mia Harris	2.0000

# physicians treating patients with a specific disease (diabetes)

	name
▶	Dr. James Wilson

# patients who have never been hospitalized

	name
▶	Tyler Hudson
	Natalie Travis
	Ryan Hoover
	Phoenix Glover
	Grayson Hess
	Senna Hairdar
	Aaran Lawing
	Tatiana Dudina

# nurses who have administered a certain medication (ibuprofen)

	name
▶	Nurse Liam Anderson

# list all patients with their latest health record

	name	disease	date
▶	Alice Green	Flu	2023-01-10
	Bob White	Fracture	2023-02-15
	Charlie Black	Hypertension	2023-03-20
	Diana Gray	Asthma	2023-04-25
	Ethan Stone	Diabetes	2023-05-30

# total amount billed to each patient

	patient_id	total_billed
▶	1	800.00
	2	2100.00
	3	3150.00
	4	2700.00
	5	1250.00

# physicians with more than 3 patients

	name
--	------

# patients who have been hospitalized more than once

	name
--	------

# most common disease among patients

	disease	count
	Flu	1

# rooms that have never been used

	roomnumber
*	NULL

## Views

```
# patient and physician information view
create view patient_physician_info as
select
```

```

    patient.id as patient_id,
    patient.name as patient_name,
    physician.id as physician_id,
    physician.name as physician_name,
    physician.fieldofexpertise
from patient
join monitored on patient.id = monitored.patientid
join physician on monitored.physicianid = physician.id;

# medication administration details view
create view medication_administration_details as
select
    medication.patientid,
    patient.name as patient_name,
    medication.nurseid,
    nurse.name as nurse_name,
    medication.date,
    medication.medicationinfo,
    medication.amount
from medication
join nurse on medication.nurseid = nurse.id
join patient on medication.patientid = patient.id;

# hospitalization summary view
create view hospitalization_summary as
select
    patient.id as patient_id,
    patient.name as patient_name,
    hospitalized.roomnumber,
    room.capacity,
    room.feepernight,
    hospitalized.startdate,
    hospitalized.enddate,
    datediff(hospitalized.enddate, hospitalized.startdate) as days_hospitalized
from hospitalized
join patient on hospitalized.patientid = patient.id
join room on hospitalized.roomnumber = room.roomnumber;

```

**Patient and Physician Information View:** This view makes it easy to see which doctor is treating which patient and what their specialty is, helping to keep track of patient care.

**Medication Administration Details View:** It shows who gave what medicine to which patient and when, which is really useful for making sure patients are getting the right care and keeping nurses organized.

**Hospitalization Summary View:** This view gives a quick look at a patient's hospital stay, like which room they're in, how long they've been there, and how much it costs, which helps with billing and managing the hospital's rooms.

# Triggers

```
delimiter //
create trigger before_hospitalized_insert
before insert on hospitalized
for each row
begin
    declare room_occupied int;

    select count(*)
    into room_occupied
    from hospitalized
    where roomnumber = new.roomnumber
    and new.startdate < enddate
    and new.enddate > startdate;

    if room_occupied > 0 then
        signal sqlstate '45000'
        set message_text = 'Room is already occupied during the selected dates';
    end if;
end; //
delimiter ;
```

```
delimiter //
create trigger before_physician_nurse_certification_insert
before insert on physician
for each row
begin
    declare cert_exists int;

    select count(*)
    into cert_exists
    from (select certificationnumber from physician
        union
        select certificationnumber from nurse) as all_certs
    where certificationnumber = new.certificationnumber;

    if cert_exists > 0 then
        signal sqlstate '45000'
        set message_text = 'Certification number already exists';
    end if;
end; //
delimiter ;
```

```
delimiter //
create trigger before_medication_insert
before insert on medication
```

```

for each row
begin
  declare patient_admitted int;

  select count(*)
  into patient_admitted
  from hospitalized
  where patientid = new.patientid
  and new.date between startdate and enddate;

  if patient_admitted = 0 then
    signal sqlstate '45000'
    set message_text = 'Patient is not currently admitted';
  end if;
end; //
delimiter ;

```

The "Before Hospitalized Insert Trigger" checks if a room is already booked for the given dates and stops double-bookings by alerting that the room is occupied.

The "Before Physician/Nurse Certification Insert Trigger" ensures each doctor and nurse has a unique certification number, preventing duplicates by flagging if the number already exists.

The "Before Medication Insert Trigger" verifies if a patient is currently admitted before allowing medication entry, preventing errors by alerting if the patient is not admitted.

## Transactions

```

# transaction 1
start transaction;

insert into hospitalized (patientid, roomnumber, startdate, enddate)
values (123, 456, '2023-11-28', NULL); # Assuming patient ID 123 is admitted to room 456 on
Nov 28, 2023

update room
set capacity = capacity - 1 # Assuming one less available spot in the room
where roomnumber = 456;

commit;

# transaction 2
start transaction;

update hospitalized
set enddate = '2023-12-04' # Setting the date to Dec 4 (my birthday), 2023

```

```
where patientid = 6 and startdate = '2023-11-28';
```

```
insert into payment (paymentid, patientid, amount, date)
```

```
values (789, 6, 1500.00, '2023-12-04'); # a payment of $1500 on Dec 4, 2023
```

```
commit;
```

Transaction 1: This transaction admits a patient (ID 123) to a room (number 456) on November 28, 2023, and then updates the room's capacity to reflect one less available spot.

Transaction 2: This transaction sets the discharge date for a patient (ID 6) to December 4, 2023, and then records a payment of \$1500 on the same date for their hospital stay.