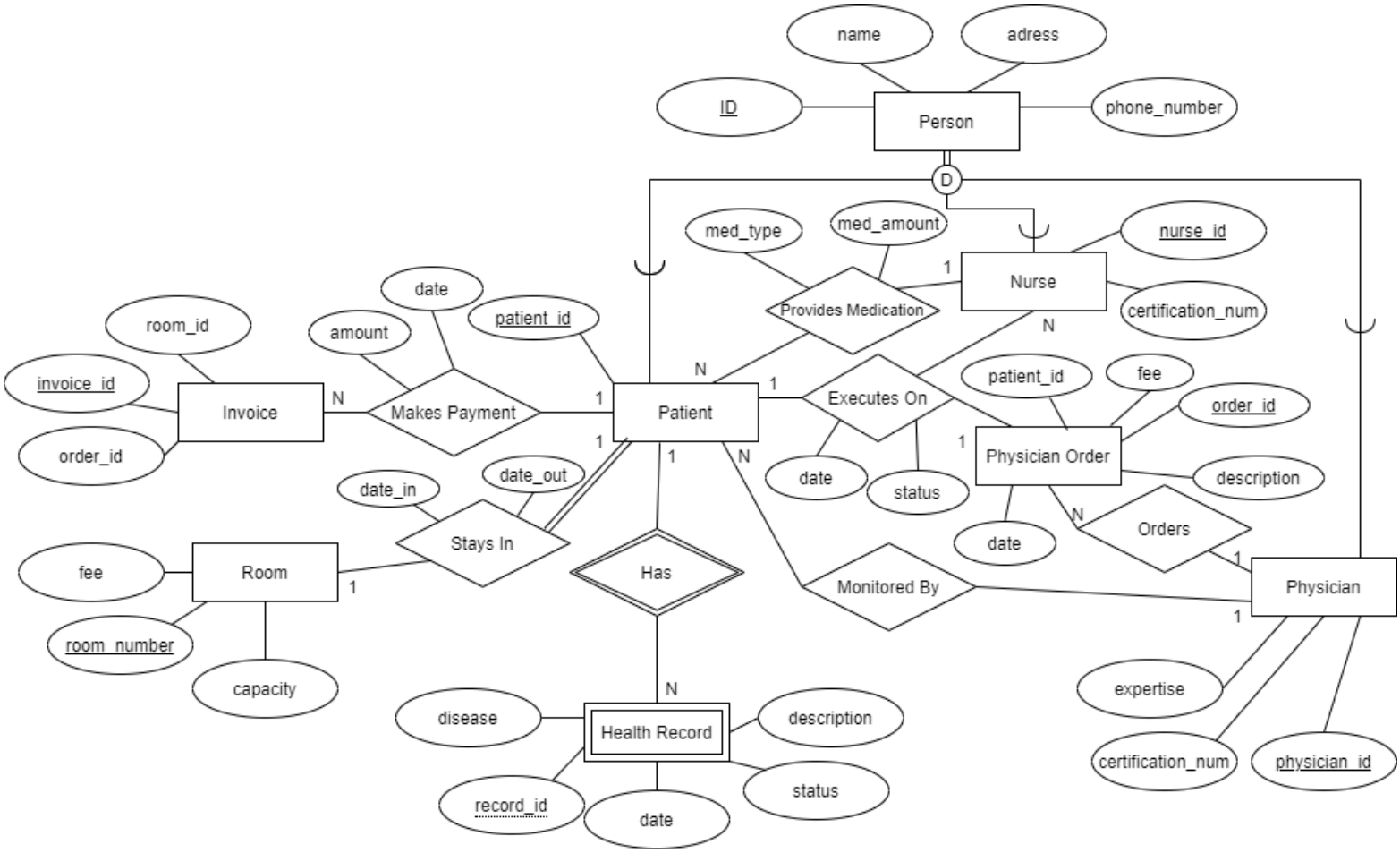

ASSUMPTIONS

A health record cannot exist without an associated patient

A patient must stay in a room

A person must be a patient, nurse, or physician

Each nurse may provide medication for multiple patients

Each physician may provide care for multiple patients

SET OF RELATIONS

rooms (room_number, capacity, room_fee)
primary key: {room_number}

contacts (id, name, phone_number, address)
primary key: {id}

physicians (physician_id, certification_num, expertise)
primary key: {physician_id}
foreign key: {physician_id references contacts(id)}

nurses (nurse_id, certification_num)
primary key: {nurse_id}
foreign key: {nurse_id references contacts(id)}

patients (patient_id, room_number, physician_id, nurse_id, date_in, date_out, med_type, med_amount)
primary key: {patient_id}
foreign key: {patient_id references contacts(id), room_number references rooms(room_number), physician_id references physicians(physician_id),nurse_id references nurse(nurse_id)}

patient_records (patient_id, record_id, date, status, description, disease)
primary key: {patient_id, record_id}
foreign key: {patient_id references patients(patient_id)}

physician_orders (order_Id, description, order_fee, instruction_date, physician_id, patient_id)
primary key: {order_id}
foreign key: {physician_id references physicians(physician_id), patient_id references patients(patient_id)}

order_results (patient_id, order_id, nurse_id, date, status)
primary key: {patient_id, order_id, nurse_id}
foreign key: {patient_id references patient(patient_id), order_id references physician_orders(order_id), nurse_id references nurse(nurse_id)}

daily_medication (nurse_id, patient_id, med_type, med_amount)
primary key: {nurse_id, patient_id}
foreign key: {nurse_id references nurses(nurse_id), patient_id references patients(patient_id)}

invoices (invoice_id, patient_id, room_number, order_id)
primary key: {invoice_id}
foreign key: {patient_id references patients(patient_id), room_number references patients(room_number, order_id references physician_orders(order_id)}

patient_payments (invoice_id, payment_amount, date)
primary key: {patient_id}
foreign key: {invoice_id references invoices(invoice_id)}

QUERIES

```
-- 1 Shows contact information for each physician
SELECT name AS physician_name, phone_number
FROM contacts
WHERE id IN (SELECT physician_id FROM physicians);
```

physician_name	phone_number
Clara Bishop	165-159-1592
Walter Daniel	610-109-6390
Ann Rhodes	150-730-7302
Jeremiah Manning	373-650-6820
Dawn Clarke	739-970-1875

```
-- 2 Shows every physician order that was not a success
SELECT *
FROM order_results
WHERE status NOT IN (SELECT status FROM order_results WHERE status = 'SUCCESS');
```

patient_id	nurse_id	order_id	date	status
11	10	4	2021-11-12	COMPLICATIONS-DELAYED
NULL	NULL	NULL	NULL	NULL

```
-- 3 Shows total capacity of hospital
SELECT SUM(capacity) AS total_capacity
FROM rooms;
```

total_capacity
9

```
-- 4 Shows all physician orders in November
SELECT o.physician_id, o.patient_id, c.name, o.instruction_date
FROM physician_orders AS o
JOIN contacts AS c ON o.patient_id = c.id
WHERE o.instruction_date IN (SELECT o.instruction_date FROM physician_orders AS o
WHERE MONTH(o.instruction_date) = 11);
```

physician_id	patient_id	name	instruction_date
5	14	Javier Matthews	2021-11-07
3	11	Carla Burgess	2021-11-12
3	15	Velma Davidson	2021-11-15

-- 5 Shows total payments from a set of specific patients
SELECT SUM(p.payment_amount) AS total_payments
FROM payments AS p JOIN invoices AS i ON p.invoice_id = i.invoice_id
JOIN patients AS pa ON pa.patient_id = i.patient_id
WHERE pa.patient_id > 12 AND pa.patient_id < 15;

total_payments
800

-- 6 Shows the average room price
SELECT AVG(room_fee) AS avg_room_fee
FROM rooms;

avg_room_fee
2155

-- 7 Shows expertise for each patient's physician
SELECT c.name AS patient_name, p.physician_id, h.expertise
FROM contacts AS c
INNER JOIN patients AS p ON c.id = p.patient_id
INNER JOIN physicians AS h ON p.physician_id = h.physician_id;

patient_name	physician_id	expertise
Neil Tran	2	Pathology
Carla Burgess	3	Dermatology
Velma Davidson	3	Dermatology
Drew Olson	4	Surgery
Javier Matthews	5	Anesthesiology

-- 8 Shows room number and price for each patient
SELECT p.patient_id, p.room_number, r.room_fee
FROM patients AS p
LEFT JOIN rooms AS r ON p.room_number = r.room_number;

patient_id	room_number	room_fee
14	1001	3250
11	1002	2300
12	1003	1600
13	1004	2225
15	1005	1400

-- 9 Shows all patients who stayed in October
SELECT p.patient_id, c.name, p.date_in, p.date_out
FROM patients AS p
JOIN contacts AS c ON p.patient_id = c.id
WHERE p.date_in IN (SELECT date_in FROM patients WHERE MONTH(date_in) = 10);

patient_id	name	date_in	date_out
12	Neil Tran	2021-10-24	2021-10-26
13	Drew Olson	2021-10-28	2021-10-29

-- 10 Shows all staff certification numbers
SELECT physician_id AS staff_id, certification_num
FROM physicians
UNION
SELECT nurse_id AS staff_id, certification_num
FROM nurses;

staff_id	certification_num
1	519251
2	630143
3	105630
4	690250
5	319502
6	151692
7	581905
8	528950
9	759210
10	594302

-- 11 Shows the number of rooms that cost more than 2000
SELECT COUNT(room_number)
FROM rooms
WHERE room_fee > 2000;

COUNT(room_number)
3

-- 12 Shows the total number of patients
SELECT COUNT(patient_id) AS num_patients
FROM patients;

num_patients
5

-- 13 Shows all contacts whose phone number starts with specific digits
SELECT c.name, c.phone_number
FROM contacts AS c
WHERE LEFT(c.phone_number, 3) = 610;

name	phone_number
Walter Daniel	610-109-6390
Yvette Wallace	610-529-2691

-- 14 Shows all rooms with a capacity larger than 1
SELECT r.room_number, r.capacity
FROM rooms AS r
WHERE r.capacity > 1;

room_number	capacity
1003	2
1004	2
1005	3
NULL	NULL

-- 15 Shows all payments in 2021
SELECT c.name, p.payment_amount, p.date
FROM contacts AS c
JOIN patients AS pa ON pa.patient_id = c.id
JOIN invoices AS i ON i.patient_id = pa.patient_id
JOIN payments AS p ON p.invoice_id = i.invoice_id
WHERE YEAR(date) = 2021;

name	payment_amount	date
Carla Burgess	500	2021-10-30
Neil Tran	400	2021-11-02
Drew Olson	300	2021-11-12
Javier Matthews	500	2021-11-22

VIEWS

```
-- Shows total cost for each patient's stay
CREATE VIEW invoice_total_cost AS
SELECT p.patient_id, c.name, r.room_fee, o.order_fee
FROM patients AS p LEFT JOIN contacts AS c ON p.patient_id = c.id
LEFT JOIN rooms AS r ON p.room_number = r.room_number
LEFT JOIN physician_orders AS o ON p.patient_id = o.patient_id;
```

This view allows a patient to see the two cost values for the room and physician order that are associated with their invoice.

```
-- Shows all contact information for hospital staff
CREATE VIEW staff AS
SELECT * FROM contacts
WHERE id in (SELECT physician_id FROM physicians)
UNION
SELECT * FROM contacts
WHERE id in (SELECT nurse_id FROM nurses);
```

This view allows you to select only the contact information from staff in the contacts table.

```
-- Shows total patient stay and the date of physician's order
CREATE VIEW patient_stay AS
SELECT p.patient_id, c.name, p.date_in, o.date AS order_date, p.date_out
FROM patients AS p LEFT JOIN contacts AS c ON p.patient_id = c.id
LEFT JOIN order_results AS o ON p.patient_id = o.patient_id;
```

This view shows the patient's name along with their total stay, including their date in, the date of a physician's order, and their date out.

TRIGGERS

```
DELIMITER //
-- Auto increments contacts id
CREATE TRIGGER contacts_auto_inc
BEFORE INSERT ON contacts
FOR EACH ROW
IF NEW.id IS NULL THEN
        SET @last_id = (select id FROM contacts ORDER BY id DESC LIMIT 1);
        SET NEW.id = @last_id +1;
END IF;
//
```

This trigger allows insertion into contacts without keeping track of the current max id.

```
-- Automatically adds patient id and room number to invoice when inserted
CREATE TRIGGER invoice_patient_trigger
AFTER INSERT ON patients
FOR EACH ROW
INSERT INTO invoices (patient_id, room_number) VALUES (NEW.patient_id,
NEW.room_number);
//
```

This trigger automatically begins to build invoice table by inserting patient_id and room_number values when you are inserting these values into the patients table.

```
-- Automatically adds order id to invoice when inserted
CREATE TRIGGER invoice_order_trigger
AFTER INSERT ON physician_orders
FOR EACH ROW
IF (NEW.patient_id IN (SELECT patient_id FROM invoices)) THEN
        UPDATE invoices
   SET invoices.order_id = NEW.order_id
   WHERE invoices.patient_id = NEW.patient_id;
END IF;
//
```

This trigger automatically completes the invoice table by providing the order_id for a patient whenever an order is added to the physician_orders table.