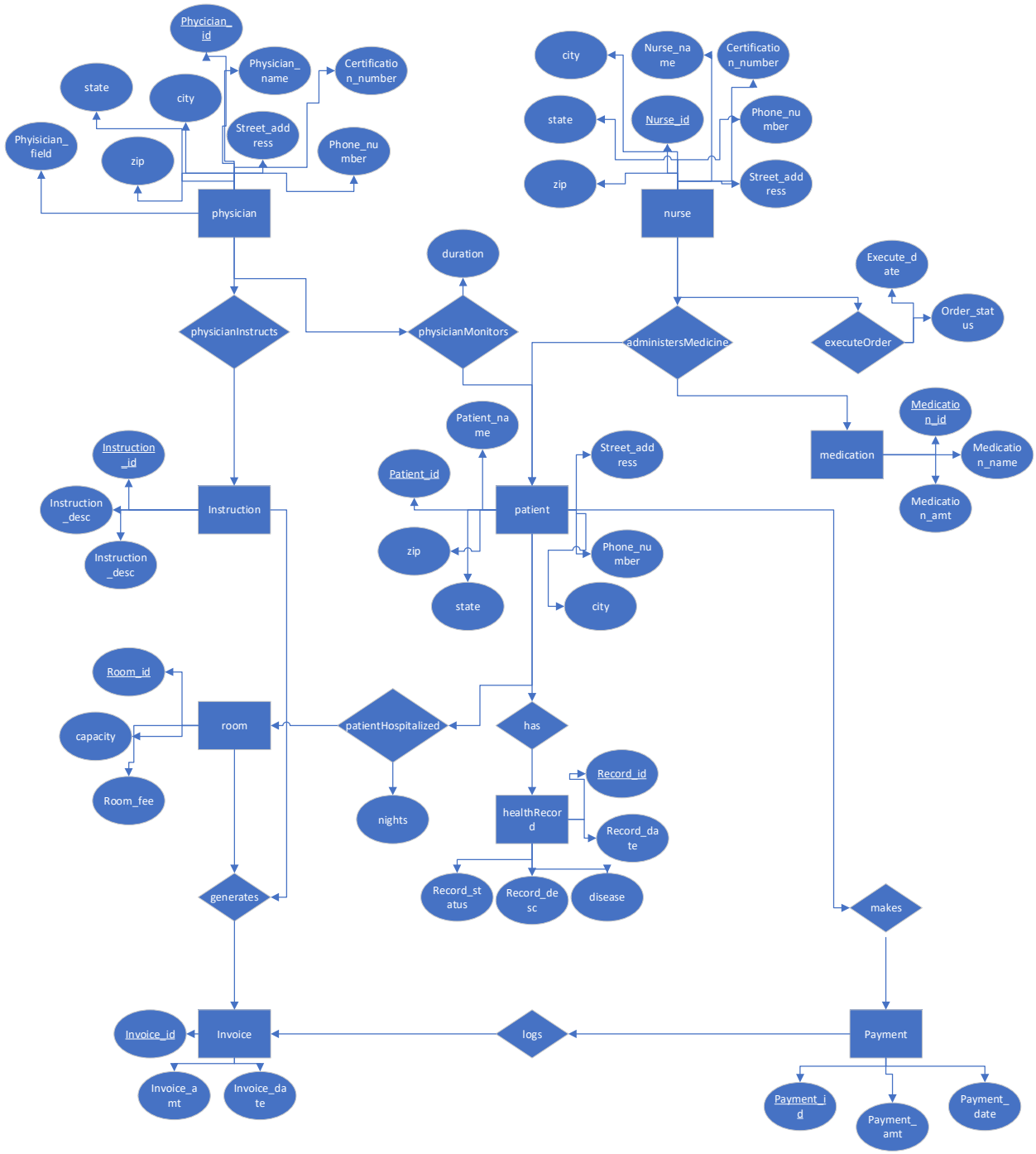
1. EERD:



I wrote down the cardinality ratios and entity relationships and participations below so that way the ERD did not become too convoluted.

Cardinality:

- patient table:
  - One patient can have one or many health records (1:N).

    - One patient can be associated with one or many invoices (1:N).
    - One patient can be hospitalized in only one room (1:1).
    - One patient can be monitored by one or many physicians (1:N).
    - One patient can receive instructions from one or many physicians (1:N).
    - One patient can have one or many executed orders (1:N).
    - One patient can be administered one or many medications (1:N).
- physician table:
    - One physician can monitor one or many patients (1:N).
    - One physician can provide instructions for one or many patients (1:N).
- nurse table:
    - One nurse can execute one or many orders (1:N).
    - One nurse can administer one or many medications (1:N).
- room table:
    - One room can accommodate only 1 patient (1:1).
- medication table:
    - One medication can be administered to zero or many patients (1:N).
- instruction table:
    - One instruction can be provided to one patient (1:1).
- invoice table:
    - One invoice is associated with one patient (1:1).
    - One invoice is associated with one instruction (1:1).
    - One invoice is associated with one room (1:1).
- payment table:
    - One payment is associated with one patient (1:1).
    - One payment is associated with one invoice (1:1).
- healthRecord table:
    - One health record is associated with one patient (1:1).
- patientHospitalized table:
    - One patient can be hospitalized in one room (1:1).
- physicianMonitors table:
    - One physician can monitor multiple patients (1:N).
- physicianInstructs table:
    - One physician can provide instructions to multiple patients (1:N).
    - One instruction can be associated with multiple medications (1:N).
- executesOrder table:
    - One nurse can execute an order for multiple patients (1:N).
    - One order is associated with one patient (1:1).
    - One order is associated with one instruction (1:1).
- administerMedicine table:
    - One nurse can administer multiple medications to one patient (N:1).
    - Multiple medications can be administered to one patient (N:1).

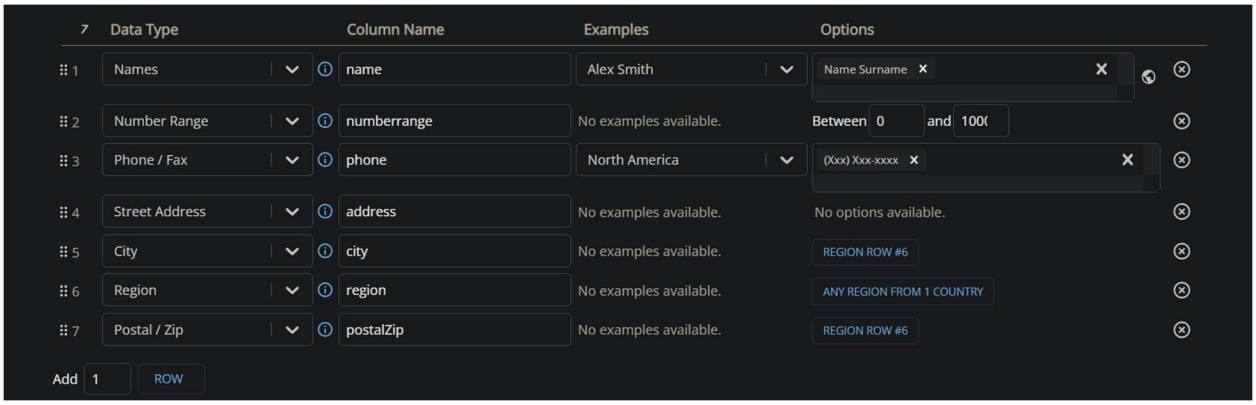
Entity relationships/participation

- patient entity:

    - Total participation in the healthRecord relationship (1:1).
    - Total participation in the invoice relationship (1:1).
    - Total participation in the patientHospitalized relationship (1:1).
    - Partial participation in the physicianMonitors relationship (1:N).
    - Partial participation in the physicianInstructs relationship (1:N).
    - Partial participation in the executesOrder relationship (1:N).
    - Partial participation in the administerMedicine relationship (1:N).
- physician entity:
    - Partial participation in the physicianMonitors relationship (1:N).
    - Partial participation in the physicianInstructs relationship (1:N).
- nurse entity:
    - Partial participation in the executesOrder relationship (1:N).
    - Partial participation in the administerMedicine relationship (1:N).
- room entity:
    - Partial participation in the patientHospitalized relationship (1:1).
- medication entity:
    - Partial participation in the physicianInstructs relationship (1:N).
    - Partial participation in the administerMedicine relationship (1:N).
- instruction entity:
    - Partial participation in the physicianInstructs relationship (1:N).
    - Partial participation in the invoice relationship (1:N).
- invoice entity:
    - Total participation in the patient relationship (1:1).
    - Total participation in the instruction relationship (1:1).
    - Total participation in the room relationship (1:1).
- payment entity:
    - Total participation in the patient relationship (1:1).
    - Total participation in the invoice relationship (1:1).
- healthRecord entity:
    - Total participation in the patient relationship (1:1).
- patientHospitalized entity:
    - Total participation in the patient relationship (1:1).
    - Total participation in the room relationship (1:1).
- physicianMonitors entity:
    - Total participation in the physician relationship (1:1).
    - Total participation in the patient relationship (1:1).
- physicianInstructs entity:
    - Total participation in the physician relationship (1:1).
    - Total participation in the instruction relationship (1:1).
    - Total participation in the medication relationship (1:1).
    - Total participation in the patient relationship (1:1).
- executesOrder entity:
    - Total participation in the nurse relationship (1:1).
    - Total participation in the patient relationship (1:1).

  - Total participation in the instruction relationship (1:1).
- administerMedicine entity:
  - Total participation in the nurse relationship (1:1).
  - Total participation in the medication relationship (1:1).
  - Total participation in the patient relationship (1:1)

2. Assumptions:
   a. Hospital Room Capacity: I assumed most hospital rooms to contain 1 patient, so I treated this field moreso as a flag. For example, if the room is vacant, capacity is set to 'Y'
   b. It is assumed that multiple nurses and physicians can treat multiple patients. It is also assumed that patients can receive more than one medication.
   c. I used a website to generate some of the basic data for the patients, physicians, and nurses. Example:



   d. I tied the duration that physician monitors the patient to the number of nights the patient has been hospitalized. For example, if the patient has been hospitalized for 3 nights, and the physician started monitoring the patient by the second night, then the physician has monitored the patient for 2 nights.
   e. Medication amounts are measured in ccs.

3. Relations and keys:
   a. Relation Patient(Patient_id, patient_name, phone_number, street_address, city, state, zip)
      i. primary key: {Patient_ID}
   b. Relation healthRecord(Record_id, Patient_ID, Disease, record_Date, record_status, record_desc)
      i. primary key: {Record_ID}
      ii. Foreign key: {Patient_ID references Patient(Patient_ID)}
   c. Relation Physician(Physician_ID, physician_name, Certification_Number, physician_field, Phone_number, street_address, city, state, zip)
      i. primary key: {Physician_ID}
   d. Relation Nurse(Nurse_ID, nurse_name, Certification_number, Phone_number, street_address, city, state, zip)

    i. primary key: {Nurse_ID}
e. Relation Room(Room_id, Capacity, room_fee)
    i. primary key: {Room_id}
f. Relation patientHospitalized(Patient_ID, Room_id, Nights)
    i. primary key: {Patient_ID, Room_id}
    ii. foreign key: {Patient_ID references Patient(Patient_ID), Room_id references Room(Room_id)}
g. Relation physicianMonitors(Physician_ID, Patient_ID, Duration)
    i. primary key: {Physician_ID, Patient_ID}
    ii. foreign key: {Physician_ID references Physician(Physician_ID), Patient_ID references Patient(Patient_ID)}
h. Relation Instruction(Instruction_id, instruction_Fee, instruction_desc)
    i. primary key: {Instruction_id}
i. Relation phsyicianInstructs(physician_id, instruction_id, medication_id, patient_id)
    i. Primary key: {physician_id, instruction_id}
    ii. Foreign key: {physician_id references Physician(physician_id), instruction_id references Instruction(instruction_id), medication_id references Medication(medication_id), patient_id references Patient(patient_id)}
j. Relation executesOrder(Nurse_ID, Patient_ID, Instruction_id, execute_Date, order_Status)
    i. primary key: {Nurse_ID, Patient_ID, Instruction_id}
    ii. foreign key: {Nurse_ID references Nurse(Nurse_ID), Patient_ID references Patient(Patient_ID), Instruction_id references Instruction(Instruction_id)}
k. Relation Medication(Medication_ID, Medication_name, medication_amt)
    i. primary key: {Medication_ID}
l. Relation administersMedicine(nurse_ID, Medication_ID, patient_id)
    i. primary key: {Physician_ID, Medication_ID,patient_id}
    ii. foreign key: {Physician_ID references Physician(Physician_ID), Medication_ID references Medication(Medication_ID), Patient_id references Patient(patient_id)}
m. Relation Payment(Payment_ID, Patient_ID, payment_Date, payment_Amt)
    i. primary key: {Payment_ID}
    ii. foreign key: {Patient_ID references Patient(Patient_ID)}
n. Relation Invoice(invoice_id, invoice_date, invoice_amt, patient_id, instruction_id, room_id)
    i. Primary key: {invoice_id}
    ii. Foreign key: {patient_id references patient(patient_id), instruction_id references Instruction(instruction_id), room_id references Room(room_id)}

4. Views and descriptions
    a. This view shows the breakdown of each patient's invoice

```sql
-- This view shows the breakdown of each patient's invoice
DROP VIEW IF EXISTS invoiceBreakdown;

CREATE VIEW invoiceBreakdown AS
SELECT p.patient_name, i.invoice_id, r.room_fee, instr.instruction_fee,
  (i.invoice_amt - r.room_fee - instr.instruction_fee) AS miscellaneous_expenses, i.invoice_amt
FROM invoice i
JOIN room r ON r.room_id = i.room_id
JOIN instruction instr ON instr.instruction_id = i.instruction_id
JOIN patient p ON p.patient_id = i.patient_id;

SELECT * FROM invoiceBreakdown;

-- This view gives a breakdown of each patient and their associated physician, nurse, and treatment
DROP VIEW IF EXISTS patientTreatment;

CREATE VIEW patientTreatment AS
SELECT p.patient_name, ph.physician_name, n.nurse_name,hr.disease, i.instruction_desc, m.medicatio
FROM patient p
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

patient_name	invoice_id	room_fee	instruction_fee	miscellaneous_expenses	invoice_amt
Lars Duke	3001	146.93	853.36	409.65	1409.94
Shea Avery	3002	266.76	235.44	1478.79	1980.99
Melinda Fletcher	3003	103.91	565.23	792.95	1462.09
Susan Pace	3004	191.31	225.90	1015.07	1432.28
Melissa Knowles	3005	292.22	696.77	425.81	1414.80

b. This view gives a breakdown of each patient and their associated physician, nurse, and treatment plan, as well as the current status.

```sql
DROP VIEW IF EXISTS patientTreatment;

CREATE VIEW patientTreatment AS
SELECT p.patient_name, ph.physician_name, n.nurse_name,hr.disease, i.instruction_desc, m.medication_name, m.medication_amt, hr.record_status
FROM patient p
JOIN physicianInstructs pi ON pi.patient_id = p.patient_id
JOIN physician ph ON ph.physician_id = pi.physician_id
JOIN administerMedicine am ON am.patient_id = p.patient_id
JOIN medication m ON m.medication_id = am.medication_id
JOIN instruction i ON i.instruction_id = pi.instruction_id
JOIN nurse n ON n.nurse_id = am.nurse_id
JOIN healthRecord hr ON hr.patient_id = p.patient_id;

SELECT * FROM patientTreatment;
-- This view details the amount due that a patient may owe to the hospital, if any.
DROP VIEW IF EXISTS paymentRecord;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

patient_name	physician_name	nurse_name	disease	instruction_desc	medication_name	medication_amt	record_status
Shea Avery	Alexander Lopez	Aquila Palmer	Strep Throat	Administer medicine, patient has a bacterial infe...	Amoxicillin	400	Open
Lars Duke	Venus Bradley	Rhonda Ballard	Migraine	Administer medicine, patient has severe migraines	Acetaminophen	200	Open
Melinda Fletcher	Whitney Henry	Ifeoma Ryan	Broken Arm	Administer medicine, patient has a broken arm	Ibuprofen	800	Open
Susan Pace	Brock Walls	Joseph Webb	Sprained Ankle	Refer to specialist	Hydrocodone	200	Closed
Melissa Knowles	Moses Williamson	Griffith Hoffman	Laceration	Administer stitches to laceration	Ibuprofen	400	Open

c. This view details the amount due that a patient may owe to the hospital, if any.

```sql
DROP VIEW IF EXISTS paymentRecord;

CREATE VIEW paymentRecord AS
SELECT
    p.patient_name,
    i.invoice_amt,
    pmt.payment_amt,
    CASE WHEN (i.invoice_amt - pmt.payment_amt) < 0 THEN 0 ELSE (i.invoice_amt - pmt.payment_amt) END AS amount_due
FROM
    patient p
    INNER JOIN invoice i ON p.patient_id = i.patient_id
    INNER JOIN payment pmt ON i.invoice_id = pmt.invoice_id;


SELECT * FROM paymentRecord;

-- Queries
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

patient_name	invoice_amt	payment_amt	amount_due
Lars Duke	1409.94	1074.11	335.83
Shea Avery	1980.99	1850.77	130.22
Melinda Fletcher	1462.09	1674.50	0
Susan Pace	1432.28	0.00	1432.28
Melissa Knowles	1414.80	1613.36	0

5. Queries, descriptions, and results
    a. Query to find which rooms are occupied and who occupies that room.

```
48    -- Query to find which rooms are occupied and who occupies that room.
49    SELECT r.room_id, r.capacity, p.patient_name
50    FROM room r
51    JOIN patientHospitalized ph ON r.room_id = ph.room_id
52    JOIN patient p ON ph.patient_id = p.patient_id;
53
54    -- Query to find which patients have not yet paid their invoice
55    SELECT patient_name
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

room_id	capacity	patient_name
100	N	Lars Duke
101	N	Shea Avery
102	N	Melinda Fletcher
103	N	Susan Pace
104	N	Melissa Knowles

b. Query to find which patients have not yet paid their invoice

```
54    -- Query to find which patients have not yet paid their invoice
55    SELECT patient_name
56    FROM patient
57    WHERE patient_id NOT IN (
58        SELECT patient_id
59        FROM payment
60        WHERE payment_amt > 0
61    );
62
63    -- Query to find the amount of past due payments
64    SELECT COUNT(*) AS remaining_payments
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

patient_name
Susan Pace

c. Query to find the amount of past due payments

```
64 •  SELECT COUNT(*) AS remaining_payments
65    FROM paymentRecord
66    WHERE amount_due > 0;
67
68    -- Query to find the total outstanding amount due to the hosp
69 •  SELECT SUM(amount_due) AS remaining_amt
70    FROM paymentRecord
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

remaining_payments
3

d. Query to find the total outstanding amount due to the hospital

```
68    -- Query to find the total outstanding amount due to the hospital
69 •  SELECT SUM(amount_due) AS remaining_amt
70    FROM paymentRecord
71    WHERE amount_due > 0;
72
73    -- Query that shows which room each nurse treats each patient in
74 •  SELECT n.nurse_name, p.patient_name, r.room_id
75    FROM nurse n
76    JOIN administerMedicine am on n.nurse_id = am.nurse_id
```

e. Query that shows which room each nurse treats each patient in

```
74 •  SELECT n.nurse_name, p.patient_name, r.room_id
75    FROM nurse n
76    JOIN administerMedicine am on n.nurse_id = am.nurse_id
77    JOIN patientHospitalized ph ON am.patient_id = ph.patient_id
78    JOIN patient p ON p.patient_id = ph.patient_id
79    JOIN room r ON ph.room_id = r.room_id;
80
81    -- Query that selects the longest amount of time a patient is stay
82 •  SELECT MAX(ph.nights) AS Nights
83    FROM  patientHospitalized ph;
84
85    -- Query that lists all hospital employees and their IDs
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nurse_name	patient_name	room_id
Aquila Palmer	Shea Avery	101
Rhonda Ballard	Lars Duke	100
Ifeoma Ryan	Melinda Fletcher	102
Joseph Webb	Susan Pace	103
Griffith Hoffman	Melissa Knowles	104

f. Query that selects the longest amount of time a patient is staying at the hospital.

```
82 •  SELECT MAX(ph.nights) AS Nights
83    FROM  patientHospitalized ph;
84
85    -- Query that lists all hospital employees and
```

Result Grid | Filter Rows: | Export: | Wrap

Nights
10

g. Query that lists all hospital employees and their IDs

```
SELECT nurse_id, nurse_name
FROM nurse
UNION
SELECT physician_id, physician_name
FROM physician
;

-- Query that selects the average room f
SELECT AVG(room_fee) AS Average_Cost
```

esult Grid | Filter Rows: | Export:

nurse_id	nurse_name
21	Aquila Palmer
22	Rhonda Ballard
23	Ifeoma Ryan
24	Joseph Webb
25	Griffith Hoffman
11	Venus Bradley
12	Alexander Lopez
13	Whitney Henry
14	Brock Walls
15	Moses Williamson

h. Query that selects the average room fee in the hospital

```
SELECT AVG(room_fee) AS Average_Cost
FROM room;

-- Query that shows which nurses adminis
SELECT nurse_id, nurse_name
FROM nurse
WHERE nurse_id IN (
```

esult Grid | Filter Rows: | Export:

Average_Cost
185.570000

i. Query that shows which nurses administer Hydrocodone

```
SELECT nurse_id, nurse_name
FROM nurse
WHERE nurse_id IN (
    SELECT nurse_id
    FROM administerMedicine
    WHERE medication_id = (
        SELECT medication_id
        FROM medication
        WHERE medication_name = 'Hydrocodone'
    )
);
```

esult Grid | Filter Rows: | Edit:

nurse_id	nurse_name
24	Joseph Webb
NULL	NULL

j. Query that shows how long each physician has been monitoring each patient

```
SELECT p.physician_name, pt.patient_name, pm.duration
FROM physician p
JOIN physicianMonitors pm ON p.physician_id = pm.physician_id
JOIN patient pt ON pt.patient_id = pm.patient_id;

-- Query that shows the patients that need further care from a spec
SELECT patient_name
FROM patient
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

physician_name	patient_name	duration
Venus Bradley	Shea Avery	6
Alexander Lopez	Melinda Fletcher	7
Whitney Henry	Melissa Knowles	4
Brock Walls	Susan Pace	2
Moses Williamson	Lars Duke	3

k. Query that shows the patients that need further care from a specialist

```
SELECT patient_name
FROM patient
WHERE patient_id IN (
    SELECT DISTINCT pi.patient_id
    FROM physicianInstructs pi
    JOIN instruction i ON pi.instruction_id = i.instruction_id
    WHERE i.instruction_desc LIKE '%specialist%'
);

-- Query that provides a list of vacant rooms
SELECT *
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

patient_name
Susan Pace

l. Query that provides a list of vacant rooms

```
SELECT *
FROM room
WHERE capacity = 'Y';
```

Result Grid | Filter Rows:

room_id	capacity	room_fee
105	Y	120.26
106	Y	187.48
107	Y	190.34
108	Y	226.54
109	Y	129.95
NULL	NULL	NULL

m. Query that provides a list of patients that are instructed to be administered medicine by a physician

```
132 •  SELECT patient_name
133    FROM patient
134    WHERE patient_id IN (
135        SELECT DISTINCT pi.patient_id
136        FROM physicianInstructs pi
137        JOIN instruction i ON pi.instruction_id = i.instruction_id
138        WHERE i.instruction_desc LIKE '%medicine%'
139    );
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

patient_name
Lars Duke
Shea Avery
Melinda Fletcher

n. Query that shows when each patient began treatment

```
142 •  SELECT p.patient_id, p.patient_name, eo.execute_date
143    FROM patient p
144    JOIN executesOrder eo ON p.patient_id = eo.patient_id;
145
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

patient_id	patient_name	execute_date
1	Lars Duke	2023-06-05
2	Shea Avery	2023-06-12
3	Melinda Fletcher	2023-06-15
4	Susan Pace	2023-06-19
5	Melissa Knowles	2023-06-08

o. Query that shows patients that need continued treatment

```
147 •  SELECT * FROM healthRecord
148    WHERE record_status = 'Open';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

record_id	patient_id	record_date	disease	record_status	record_desc
2001	1	2023-06-18	Migraine	Open	Patient suffering from severe migraines
2002	2	2023-06-20	Strep Throat	Open	Patient has sore throat and flu like symptoms
2003	3	2023-06-20	Broken Arm	Open	Patient has intense arm pain
2005	5	2023-06-11	Laceration	Open	Patient has a deep cut on hand
NULL	NULL	NULL	NULL	NULL	NULL