University of North Carolina at Charlotte

Hospital Database in SQL

Final Report

Melissa Rivera-Franco

ITCS 3160: Database Design and Implementation

Dr. Sara Riazi

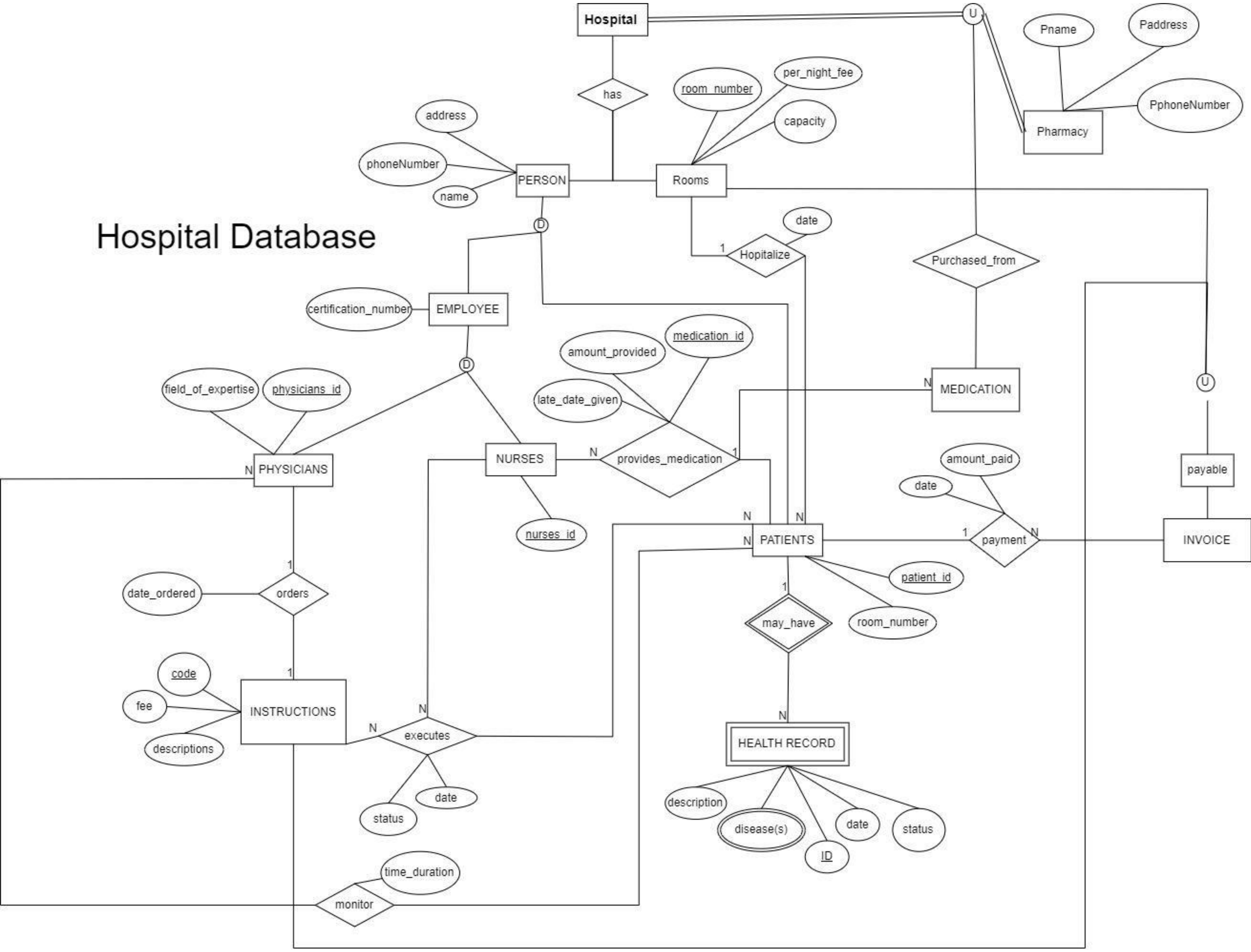
December 1, 2023

TABLE OF CONTEXT

1 E(ER) Diagram

1.1 Diagram

No Assumptions were made.



This chart chart is the updated version of the original one submitted for part one of the final projects. The updates made were cordialities corrections. The relationship of the health record to the patient was updated to a N:1. The relationship of the physician to the patient (orders) updated to 1:1 and (monitors) to N:N. The nurse entity relationship with the patient was also updated to a N:N.

2 Relations and Keys

PERSON (IdNum, name, phoneNumber, address)

Primary key: {IdNum}

EMPLOYEE (IdNum, certificatin_number)

Primary key { IdNum }

Foreign key {IdNum references PERSON(IdNum)}

PATIENTS (patient_id, room_number, hospitalized_date)

Primary key {patient_id}

Foreign key {patient_id references PERSON(IdNum), room_number references ROOMS(room_number)}

HEALTH_RECORD(patient_id, record_id, date, status, description)

Primary key{patient_id, record_id}

Foreign key{ patient_id references PATIENTS(patient_id)}

RECORD_DISEASES (record_id, disease)

Primary key{record_id, disease}

Foreign key{record_id references HEALTH_RECORD(record_id)}

PHYSICIANS(physicians_id, field_of_expertise, certification_number)

Primary key {physicians_id}

Foreign key{ physicians_id references PERSON(IdNum)}

INSTRUCTIONS (code, fee, descriptions, date_ordered , physician_id)

Primary key {code}

Foreign key{ physician_id references PHYSICIANS(phycian_id)}

EXECUTES(code, nurses_id, patient_id, status, date)

Primary key{code, nurses_id, patient_id}

Foreign key{ code references INSTRUCTION(code), nurses_id references NURSES(nurses_id), patient_id references PATIENTS(patient_id)}

MONITORS(time_duration, patient_id, physician_id)

Primary key{patient_id, physician_id}

Foreign key {patient_id references PATIENTS(patient_id), physician_id references PHYSICIANS(phycian_id)}

NURSES (nurses_id)

Primary key {nurses_id}

Foreign key {nurses_id references PERSON(IdNum)}

PROVIDES_MEDICATION(nurses_id, medication_id, amount_provided, last_date_given)

Primary key{nurses_id, medication_id}

Foreign key{nurses_id references NURSES(nurses_id), medication_id references MEDICATION(medication_id)}

MEDICATION(medication_id, description, pharmacy_id)

Primary key{medication_id}

Foreign key {pharmacy_id references PHARMACY(pID)}

PHARMACY(pID, pname, paddress, pphoneNumber)

Primary key{pID}

ROOMS (room_number, capacity, per_night_fee)

Primary Key {room_number}

INVOICE(patient_id, invoice_id, amount_paid, date)

Primary key {patient_id, invoice_id}

Foreign key {patient_id references PATIENTS(patient_id)}

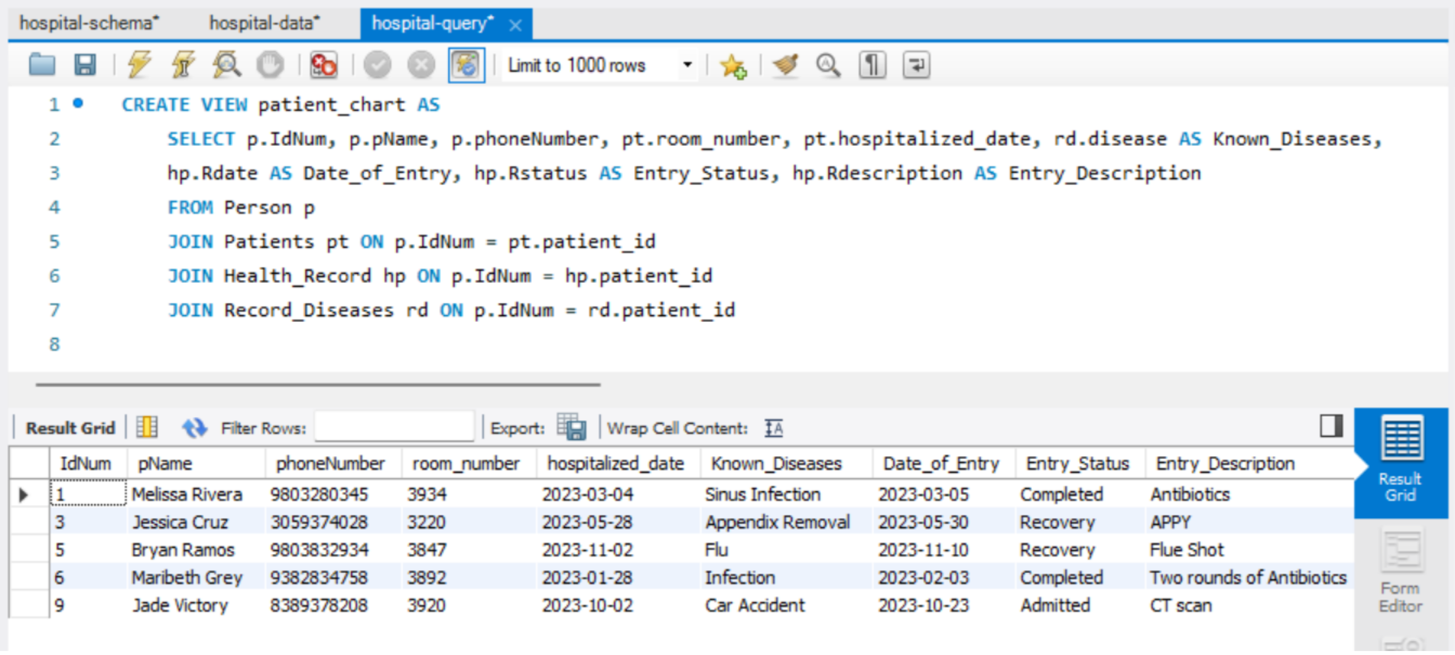
PAYABLE(room_id, code, invoice_id, amount_due)

Primary key(room_id, code)

Foreign key{room_id references ROOM(room_id), code references INSTRUCTION(code), invoice_id references INVOICE(invoice_id)}

4 Views and Descriptions

View 1:



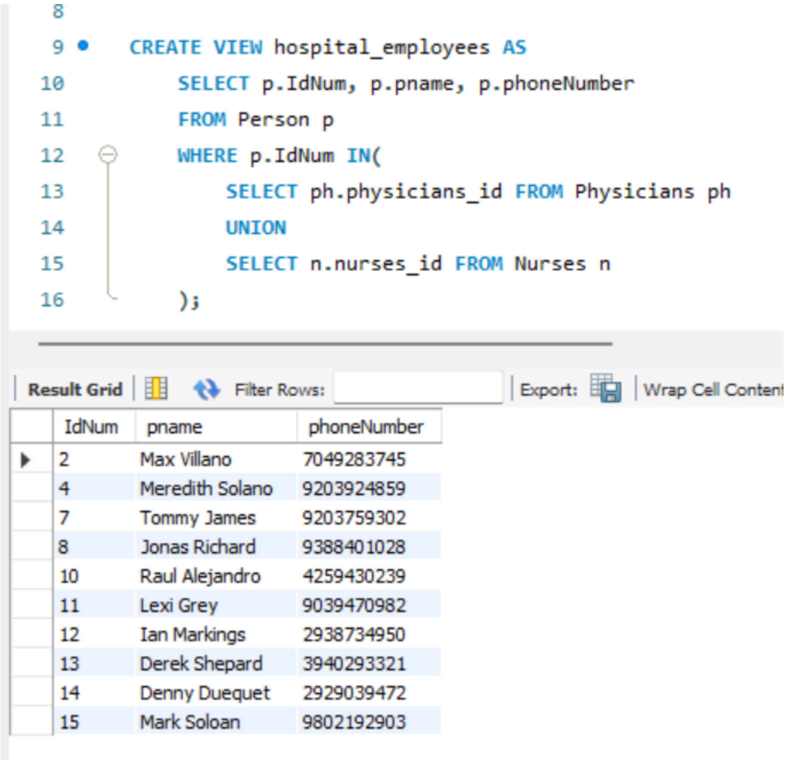
```sql
CREATE VIEW patient_chart AS
    SELECT p.IdNum, p.pName, p.phoneNumber, pt.room_number, pt.hospitalized_date, rd.disease AS Known_Diseases,
    hp.Rdate AS Date_of_Entry, hp.Rstatus AS Entry_Status, hp.Rdescription AS Entry_Description
    FROM Person p
    JOIN Patients pt ON p.IdNum = pt.patient_id
    JOIN Health_Record hp ON p.IdNum = hp.patient_id
    JOIN Record_Diseases rd ON p.IdNum = rd.patient_id
```

IdNum	pName	phoneNumber	room_number	hospitalized_date	Known_Diseases	Date_of_Entry	Entry_Status	Entry_Description
1	Melissa Rivera	9803280345	3934	2023-03-04	Sinus Infection	2023-03-05	Completed	Antibiotics
3	Jessica Cruz	3059374028	3220	2023-05-28	Appendix Removal	2023-05-30	Recovery	APPY
5	Bryan Ramos	9803832934	3847	2023-11-02	Flu	2023-11-10	Recovery	Flue Shot
6	Maribeth Grey	9382834758	3892	2023-01-28	Infection	2023-02-03	Completed	Two rounds of Antibiotics
9	Jade Victory	8389378208	3920	2023-10-02	Car Accident	2023-10-23	Admitted	CT scan

Description:

This view is helpful because it simplifies finding a patient's chart. It pulls all the information that the patient may have on their medical chart. This excludes invoices. I find this view helpful since it avoids having to join three tables together when trying to reach a simple result.

View 2:



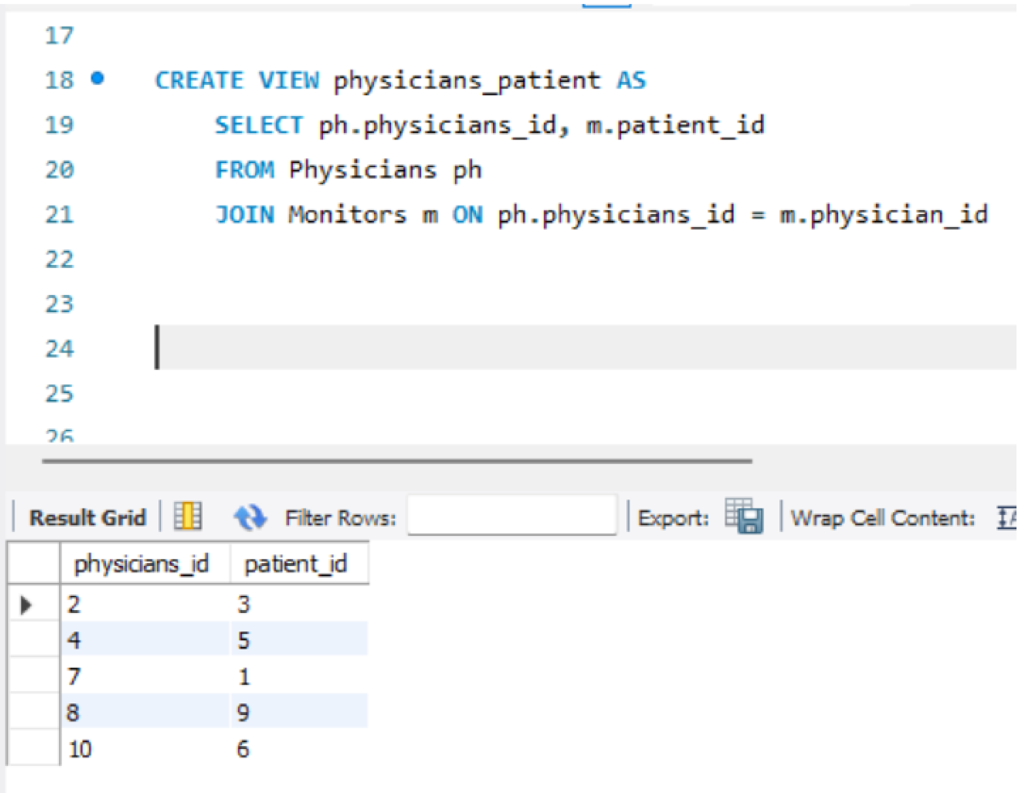
```sql
CREATE VIEW hospital_employees AS
    SELECT p.IdNum, p.pname, p.phoneNumber
    FROM Person p
    WHERE p.IdNum IN(
        SELECT ph.physicians_id FROM Physicians ph
        UNION
        SELECT n.nurses_id FROM Nurses n
    );
```

IdNum	pname	phoneNumber
2	Max Villano	7049283745
4	Meredith Solano	9203924859
7	Tommy James	9203759302
8	Jonas Richard	9388401028
10	Raul Alejandro	4259430239
11	Lexi Grey	9039470982
12	Ian Markings	2938734950
13	Derek Shepard	3940293321
14	Denny Duequet	2929039472
15	Mark Soloan	9802192903

Description:

This view returns the personal information of every hospital employee. Whether they may be a nurse or physician. It's helpful in the sense of retrieving contact information in case a message was needed to be sent out to all employees. Also helpful if an employee verification was needed.

View 3:



```
CREATE VIEW physicians_patient AS
    SELECT ph.physicians_id, m.patient_id
    FROM Physicians ph
    JOIN Monitors m ON ph.physicians_id = m.physician_id
```

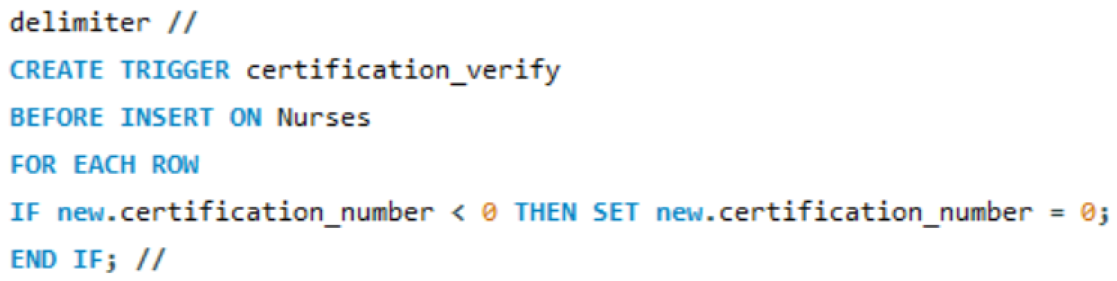
physicians_id	patient_id
2	3
4	5
7	1
8	9
10	6

Description:

The result of this view is the physician's id and the patient's id of the one they monitor.

5 Triggers and Descriptions

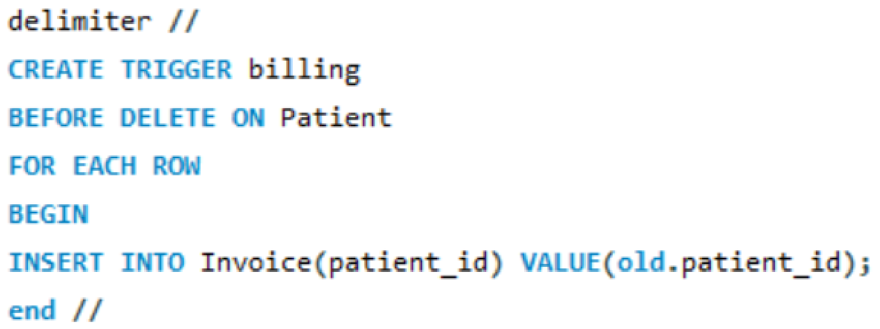
Trigger 1:



```
delimiter //
CREATE TRIGGER certification_verify
BEFORE INSERT ON Nurses
FOR EACH ROW
IF new.certification_number < 0 THEN SET new.certification_number = 0;
END IF; //
```

Description: This trigger occurs when an new nurse is entered into the system to verify their certification number is valid

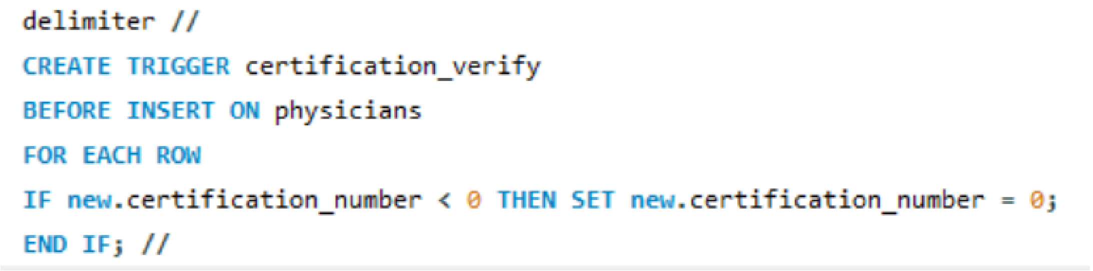
Trigger 2:



```
delimiter //
CREATE TRIGGER billing
BEFORE DELETE ON Patient
FOR EACH ROW
BEGIN
INSERT INTO Invoice(patient_id) VALUE(old.patient_id);
end //
```

Description: When a patient is removed from the database a new invoice row is created to store the old patient id

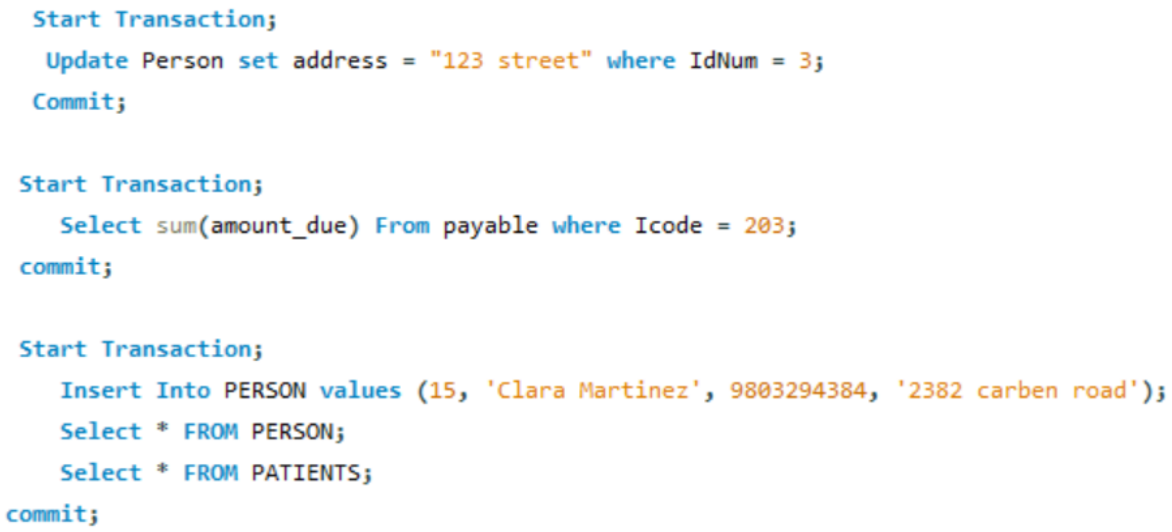
Trigger 3:



```
delimiter //
CREATE TRIGGER certification_verify
BEFORE INSERT ON physicians
FOR EACH ROW
IF new.certification_number < 0 THEN SET new.certification_number = 0;
END IF; //
```

Description: This one does the same as the first one to ensure the certification number entered is valid

6 Transactions and Descriptions



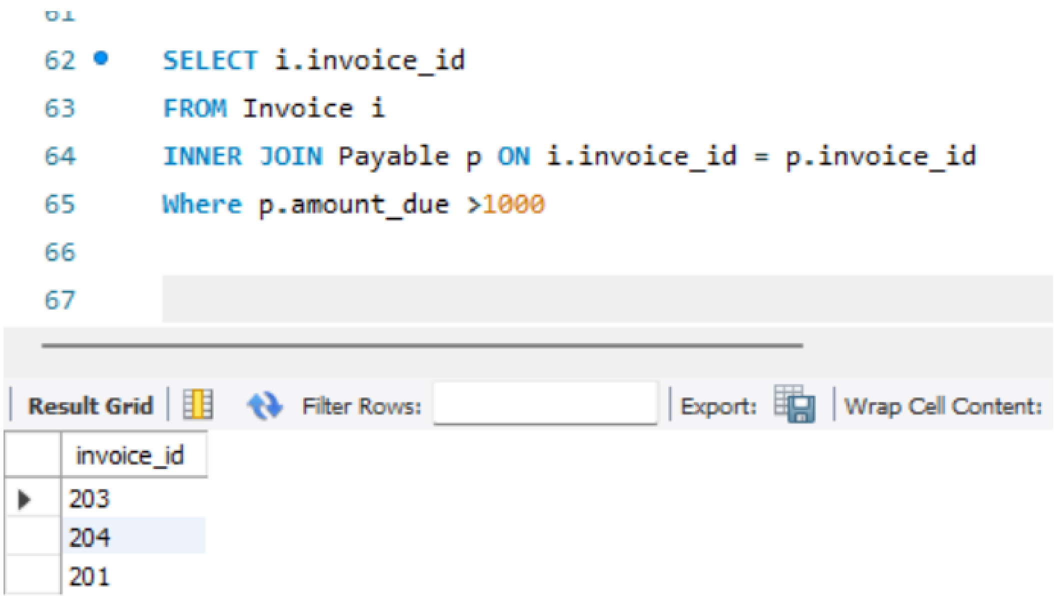
```
  Start Transaction;
   Update Person set address = "123 street" where IdNum = 3;
  Commit;

 Start Transaction;
    Select sum(amount_due) From payable where Icode = 203;
 commit;

 Start Transaction;
    Insert Into PERSON values (15, 'Clara Martinez', 9803294384, '2382 carben road');
    Select * FROM PERSON;
    Select * FROM PATIENTS;
commit;
```

The first transaction updates a person's address given their IdNum. The second transaction finds the sum of the amount due for the Instruction with a Icode of 203. The third one inserts new values in the person table and select the whole person table and patients.

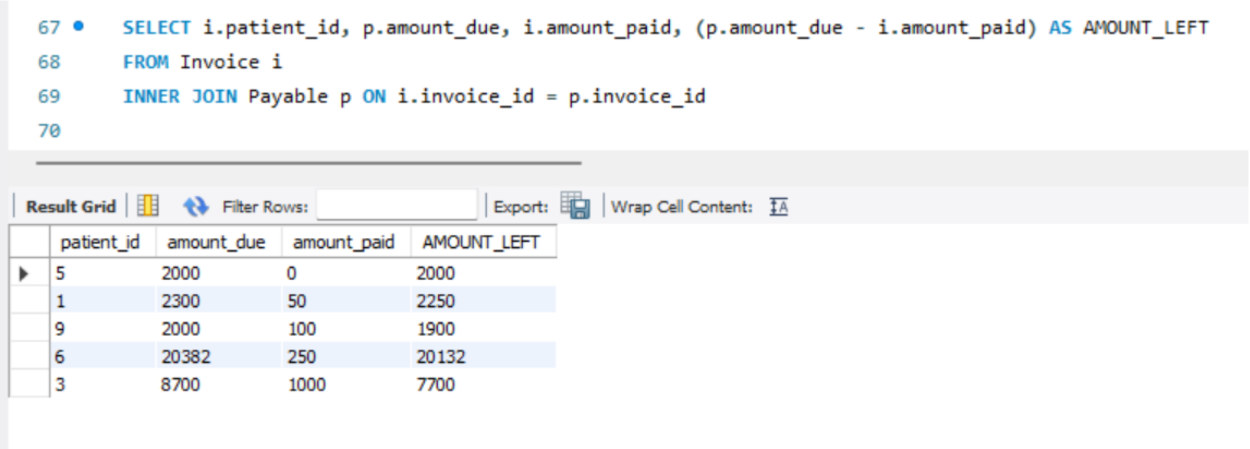
7 Database Queries



```sql
SELECT i.invoice_id
FROM Invoice i
INNER JOIN Payable p ON i.invoice_id = p.invoice_id
Where p.amount_due >1000
```

invoice_id
203
204
201

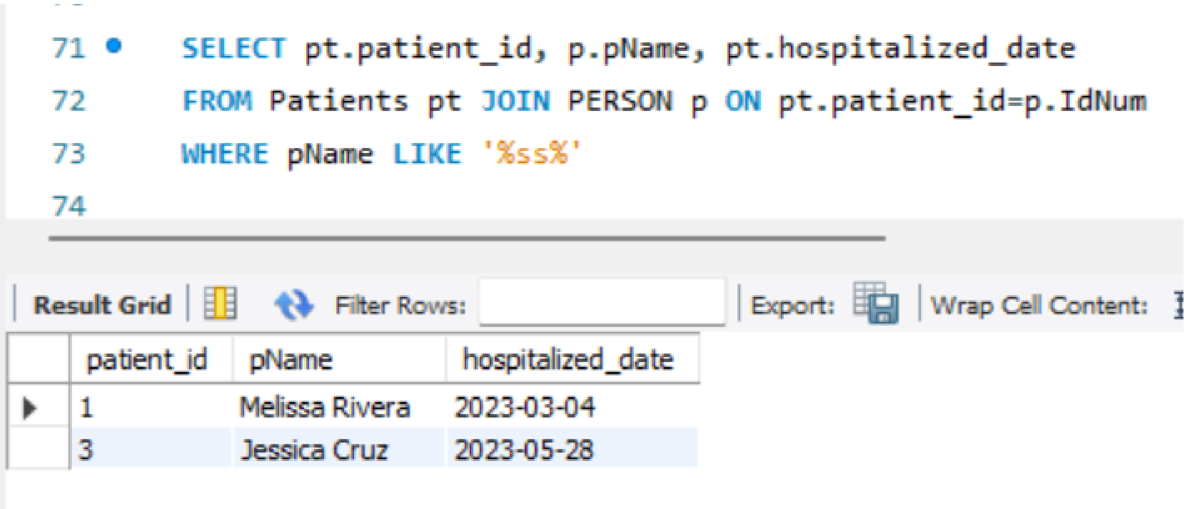
1.



```sql
SELECT i.patient_id, p.amount_due, i.amount_paid, (p.amount_due - i.amount_paid) AS AMOUNT_LEFT
FROM Invoice i
INNER JOIN Payable p ON i.invoice_id = p.invoice_id
```

patient_id	amount_due	amount_paid	AMOUNT_LEFT
5	2000	0	2000
1	2300	50	2250
9	2000	100	1900
6	20382	250	20132
3	8700	1000	7700

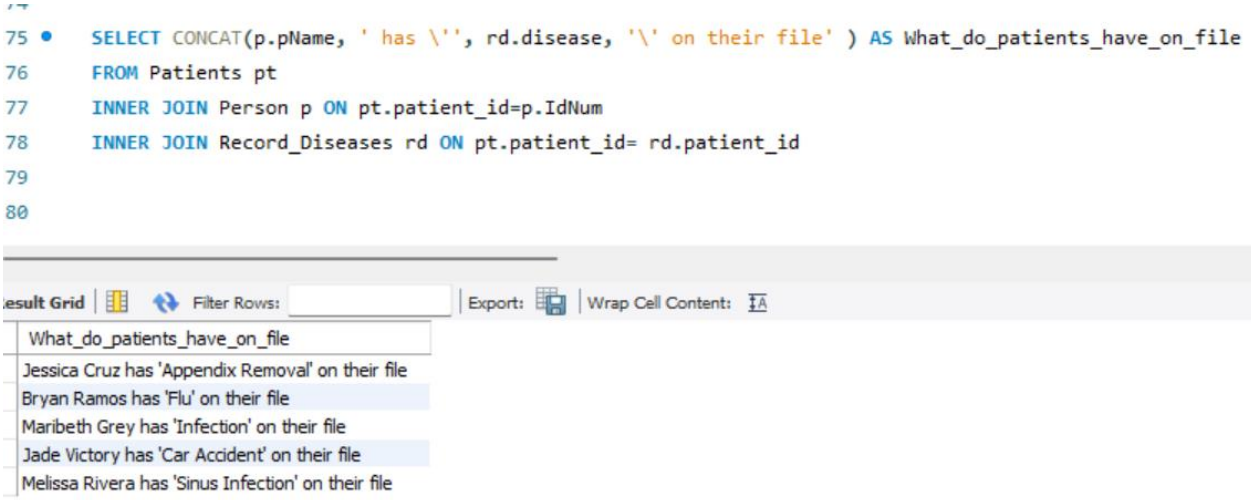
2.



```sql
SELECT pt.patient_id, p.pName, pt.hospitalized_date
FROM Patients pt JOIN PERSON p ON pt.patient_id=p.IdNum
WHERE pName LIKE '%ss%'
```

patient_id	pName	hospitalized_date
1	Melissa Rivera	2023-03-04
3	Jessica Cruz	2023-05-28

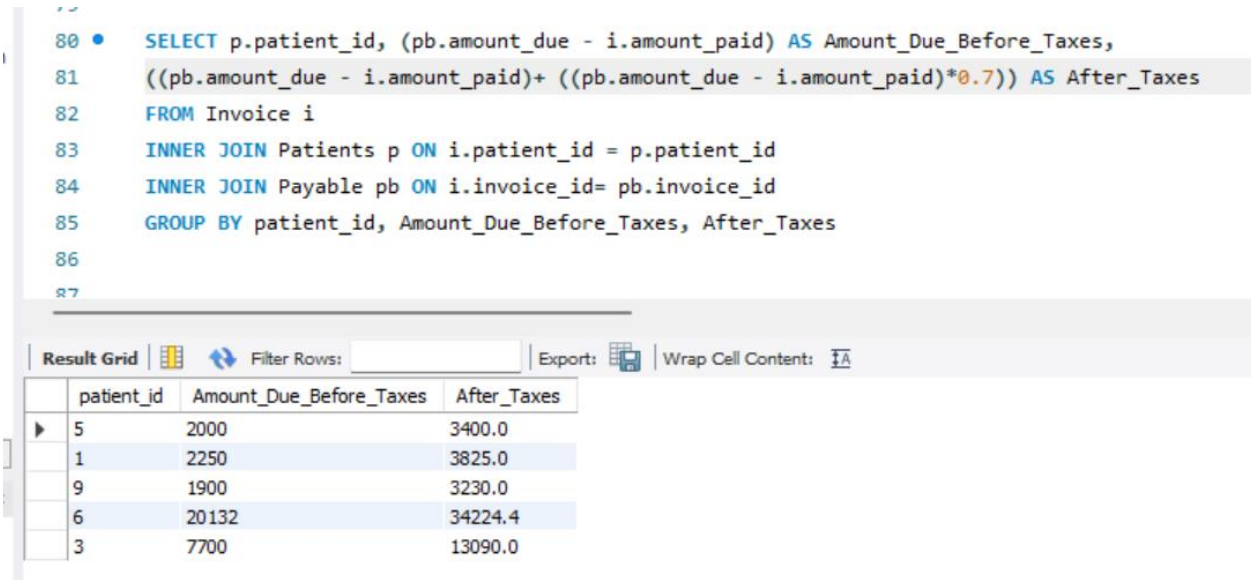
3.



```sql
SELECT CONCAT(p.pName, ' has \'', rd.disease, '\' on their file' ) AS What_do_patients_have_on_file
FROM Patients pt
INNER JOIN Person p ON pt.patient_id=p.IdNum
INNER JOIN Record_Diseases rd ON pt.patient_id= rd.patient_id
```

What_do_patients_have_on_file
Jessica Cruz has 'Appendix Removal' on their file
Bryan Ramos has 'Flu' on their file
Maribeth Grey has 'Infection' on their file
Jade Victory has 'Car Accident' on their file
Melissa Rivera has 'Sinus Infection' on their file

4.

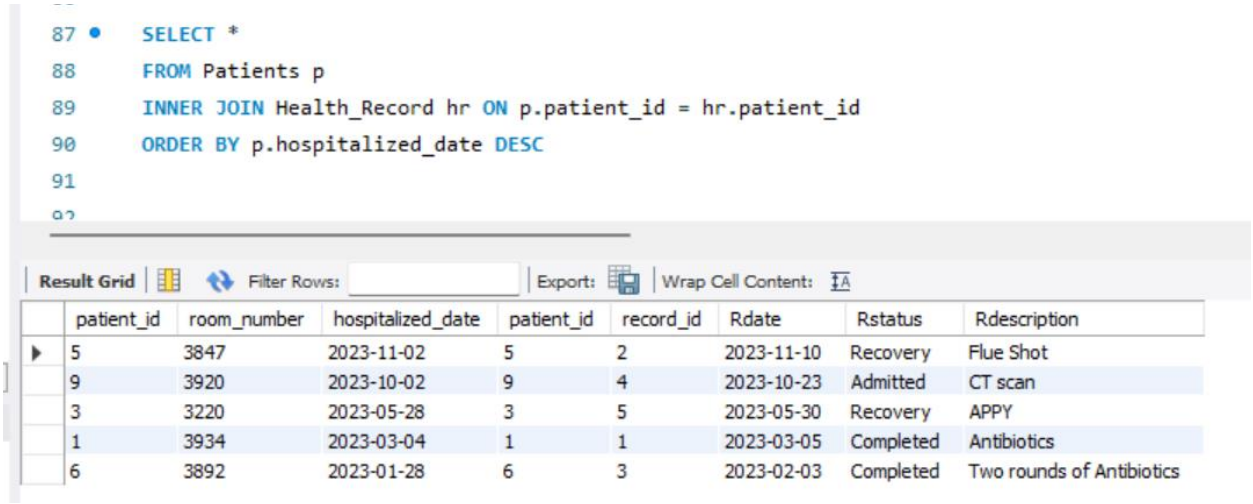


```sql
SELECT p.patient_id, (pb.amount_due - i.amount_paid) AS Amount_Due_Before_Taxes,
((pb.amount_due - i.amount_paid)+ ((pb.amount_due - i.amount_paid)*0.7)) AS After_Taxes
FROM Invoice i
INNER JOIN Patients p ON i.patient_id = p.patient_id
INNER JOIN Payable pb ON i.invoice_id= pb.invoice_id
GROUP BY patient_id, Amount_Due_Before_Taxes, After_Taxes
```

patient_id	Amount_Due_Before_Taxes	After_Taxes
5	2000	3400.0
1	2250	3825.0
9	1900	3230.0
6	20132	34224.4
3	7700	13090.0

5.

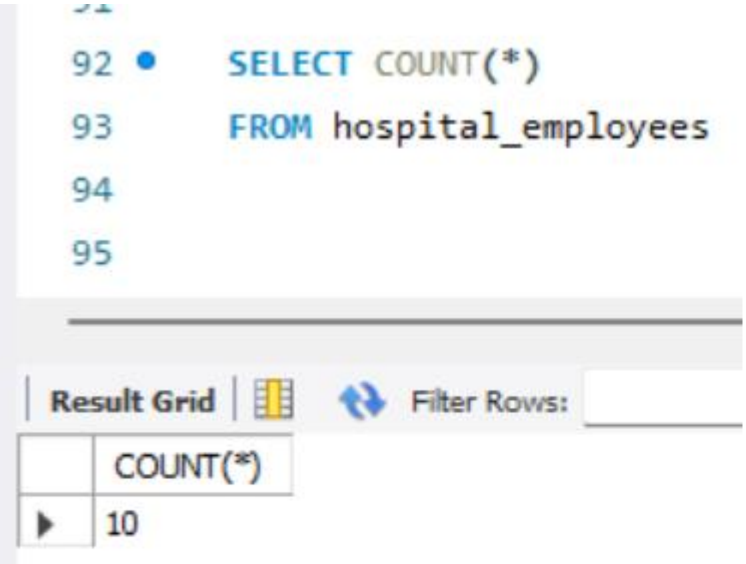
6.



```sql
SELECT *
FROM Patients p
INNER JOIN Health_Record hr ON p.patient_id = hr.patient_id
ORDER BY p.hospitalized_date DESC
```

patient_id	room_number	hospitalized_date	patient_id	record_id	Rdate	Rstatus	Rdescription
5	3847	2023-11-02	5	2	2023-11-10	Recovery	Flue Shot
9	3920	2023-10-02	9	4	2023-10-23	Admitted	CT scan
3	3220	2023-05-28	3	5	2023-05-30	Recovery	APPY
1	3934	2023-03-04	1	1	2023-03-05	Completed	Antibiotics
6	3892	2023-01-28	6	3	2023-02-03	Completed	Two rounds of Antibiotics

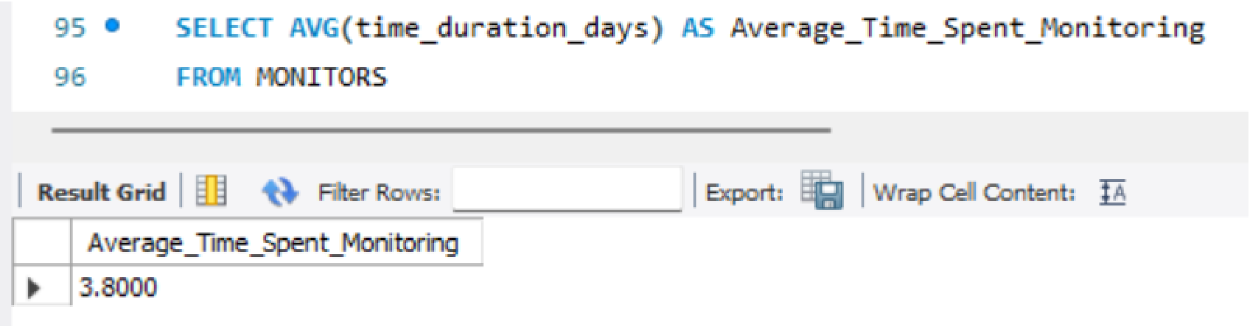
7.



```sql
SELECT COUNT(*)
FROM hospital_employees
```

COUNT(*)
10

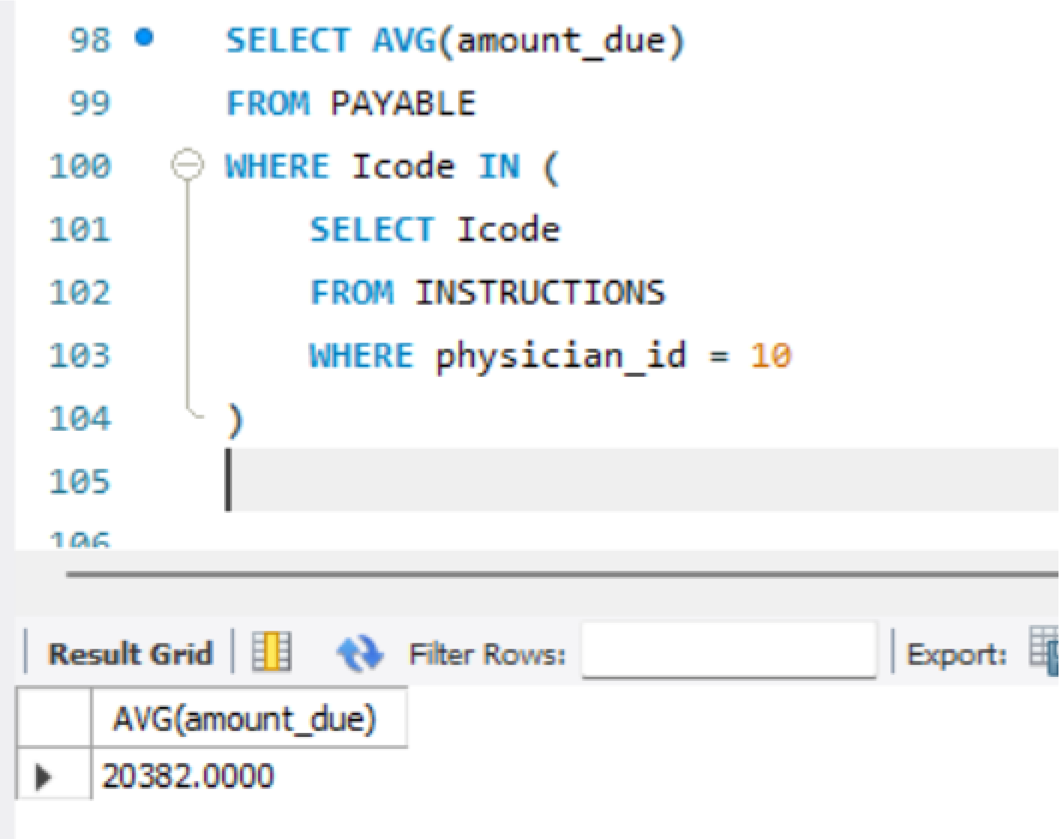
8.



```sql
SELECT AVG(time_duration_days) AS Average_Time_Spent_Monitoring
FROM MONITORS
```

Average_Time_Spent_Monitoring
3.8000

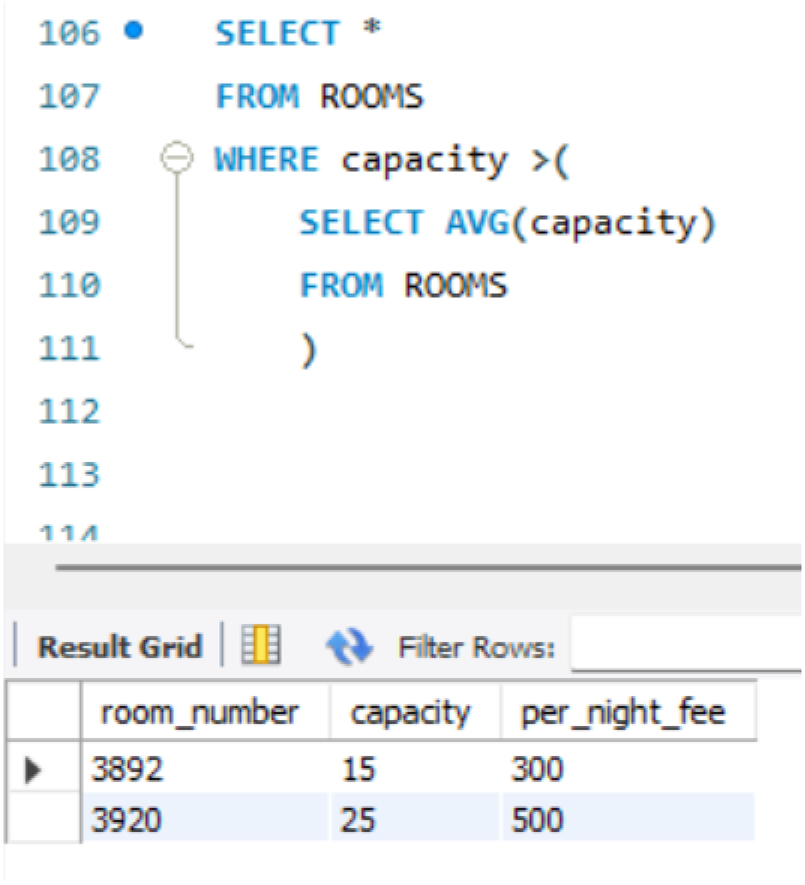
9.



```sql
SELECT AVG(amount_due)
FROM PAYABLE
WHERE Icode IN (
    SELECT Icode
    FROM INSTRUCTIONS
    WHERE physician_id = 10
)
```

	AVG(amount_due)
▶	20382.0000

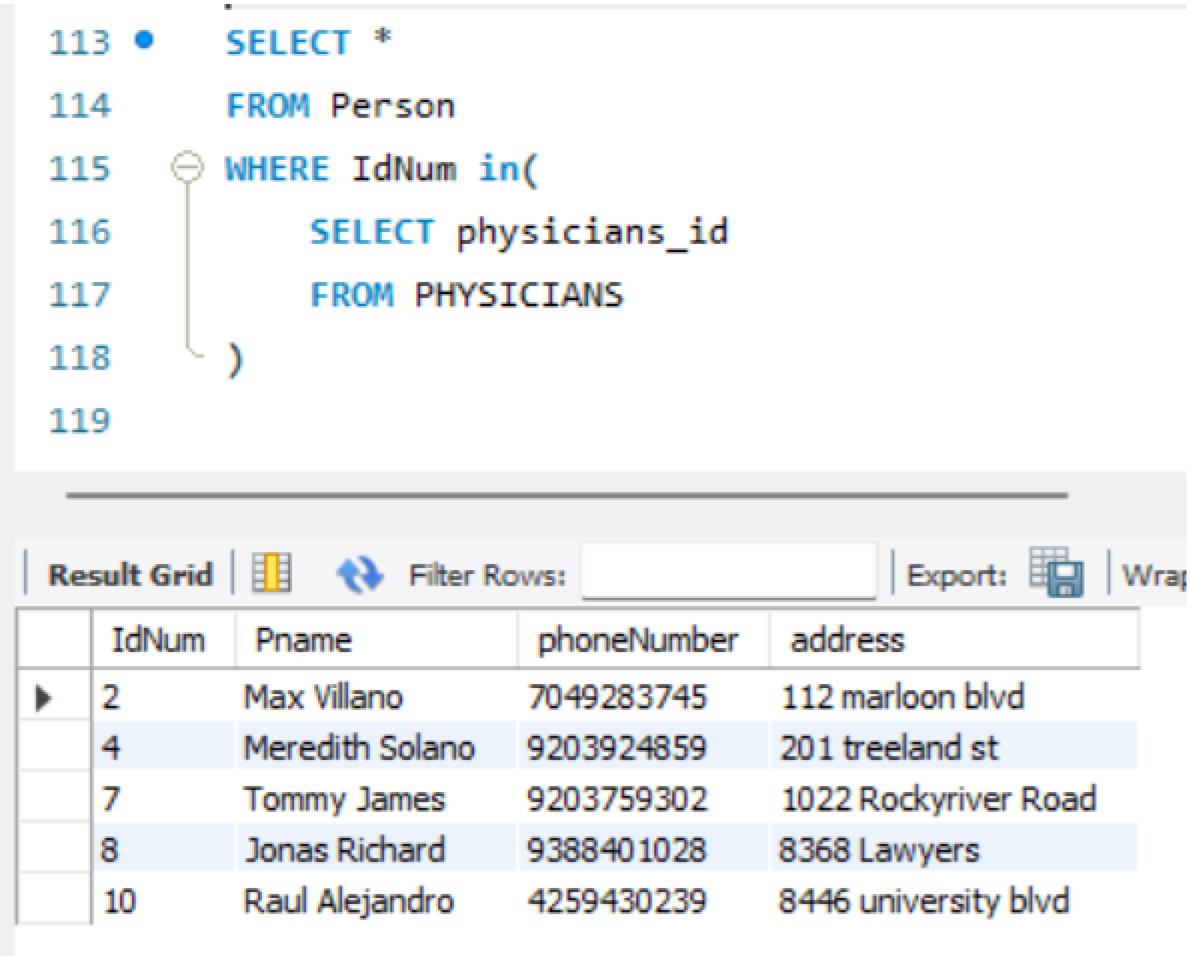
10.



```sql
SELECT *
FROM ROOMS
WHERE capacity >(
    SELECT AVG(capacity)
    FROM ROOMS
    )
```

	room_number	capacity	per_night_fee
▶	3892	15	300
	3920	25	500

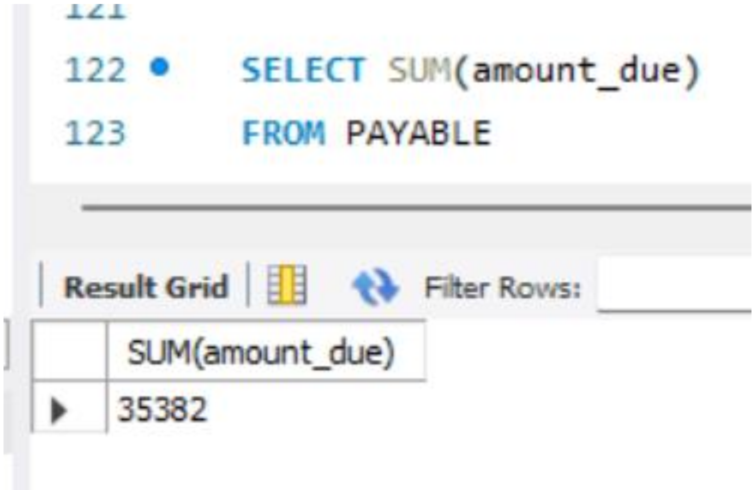
11.



```
SELECT *
FROM Person
WHERE IdNum in(
    SELECT physicians_id
    FROM PHYSICIANS
)
```

IdNum	Pname	phoneNumber	address
2	Max Villano	7049283745	112 marloon blvd
4	Meredith Solano	9203924859	201 treeland st
7	Tommy James	9203759302	1022 Rockyriver Road
8	Jonas Richard	9388401028	8368 Lawyers
10	Raul Alejandro	4259430239	8446 university blvd

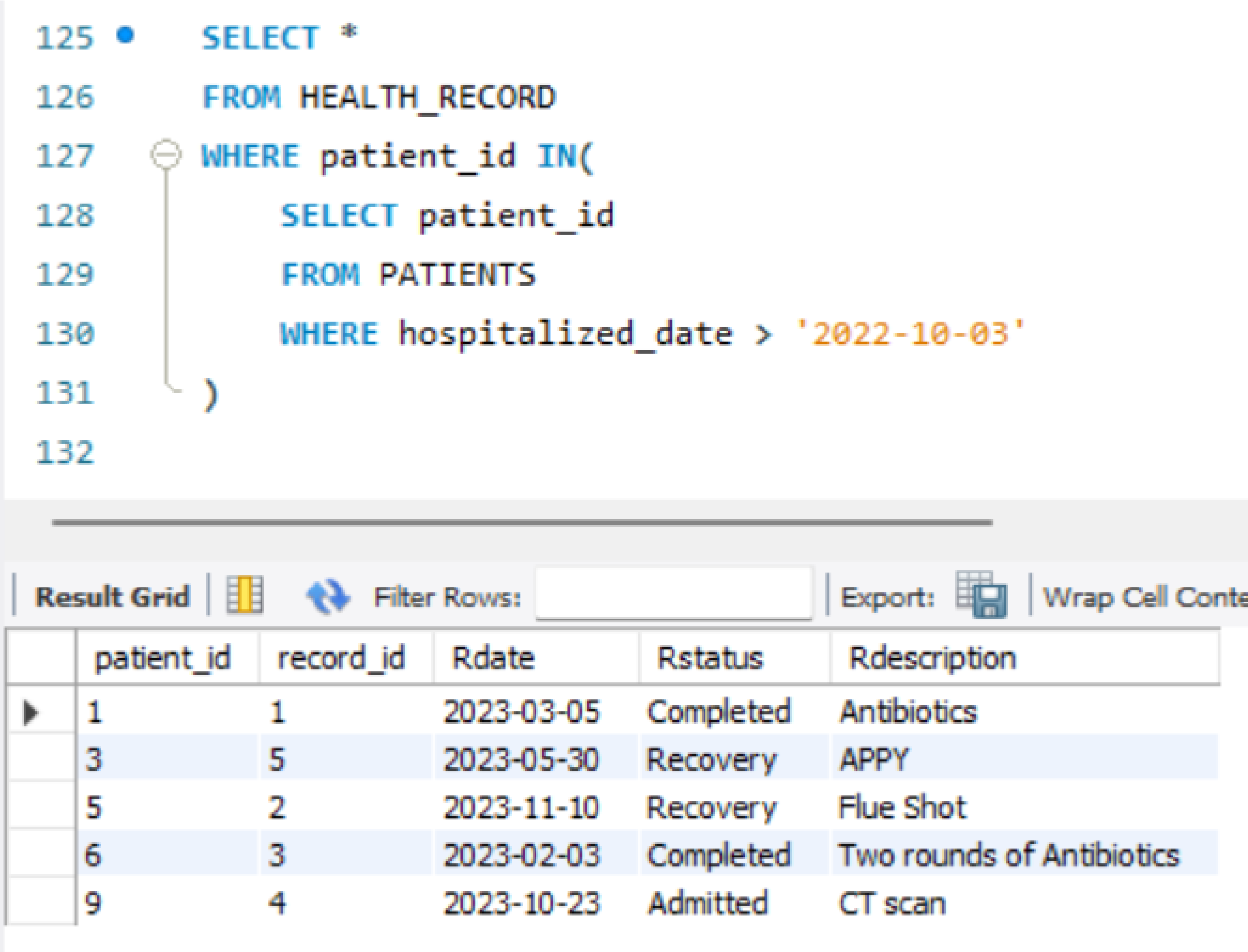
12.



```
SELECT SUM(amount_due)
FROM PAYABLE
```

SUM(amount_due)
35382

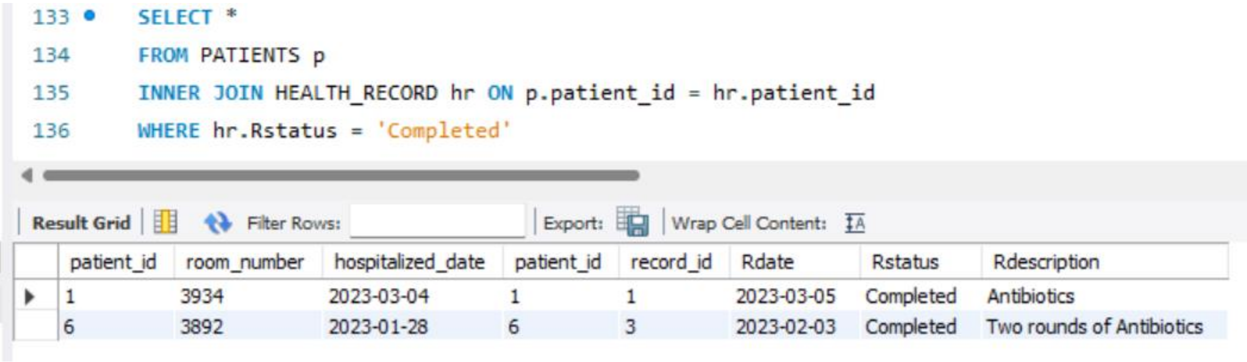
13.



```sql
SELECT *
FROM HEALTH_RECORD
WHERE patient_id IN(
    SELECT patient_id
    FROM PATIENTS
    WHERE hospitalized_date > '2022-10-03'
)
```

patient_id	record_id	Rdate	Rstatus	Rdescription
1	1	2023-03-05	Completed	Antibiotics
3	5	2023-05-30	Recovery	APPY
5	2	2023-11-10	Recovery	Flue Shot
6	3	2023-02-03	Completed	Two rounds of Antibiotics
9	4	2023-10-23	Admitted	CT scan

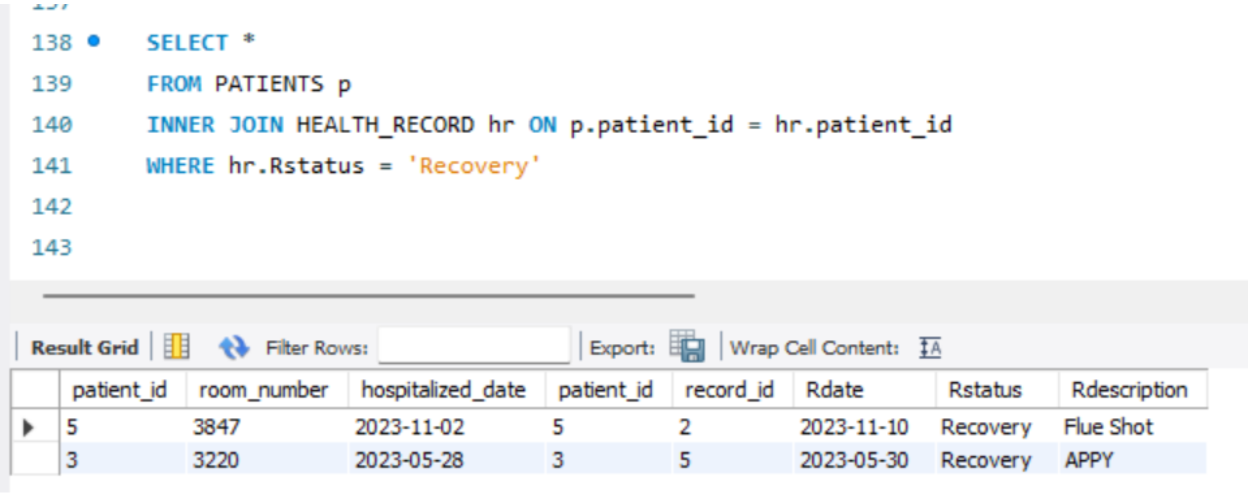
14.



```sql
SELECT *
FROM PATIENTS p
INNER JOIN HEALTH_RECORD hr ON p.patient_id = hr.patient_id
WHERE hr.Rstatus = 'Completed'
```

patient_id	room_number	hospitalized_date	patient_id	record_id	Rdate	Rstatus	Rdescription
1	3934	2023-03-04	1	1	2023-03-05	Completed	Antibiotics
6	3892	2023-01-28	6	3	2023-02-03	Completed	Two rounds of Antibiotics

15.



```
SELECT *
FROM PATIENTS p
INNER JOIN HEALTH_RECORD hr ON p.patient_id = hr.patient_id
WHERE hr.Rstatus = 'Recovery'
```

patient_id	room_number	hospitalized_date	patient_id	record_id	Rdate	Rstatus	Rdescription
5	3847	2023-11-02	5	2	2023-11-10	Recovery	Flue Shot
3	3220	2023-05-28	3	5	2023-05-30	Recovery	APPY