

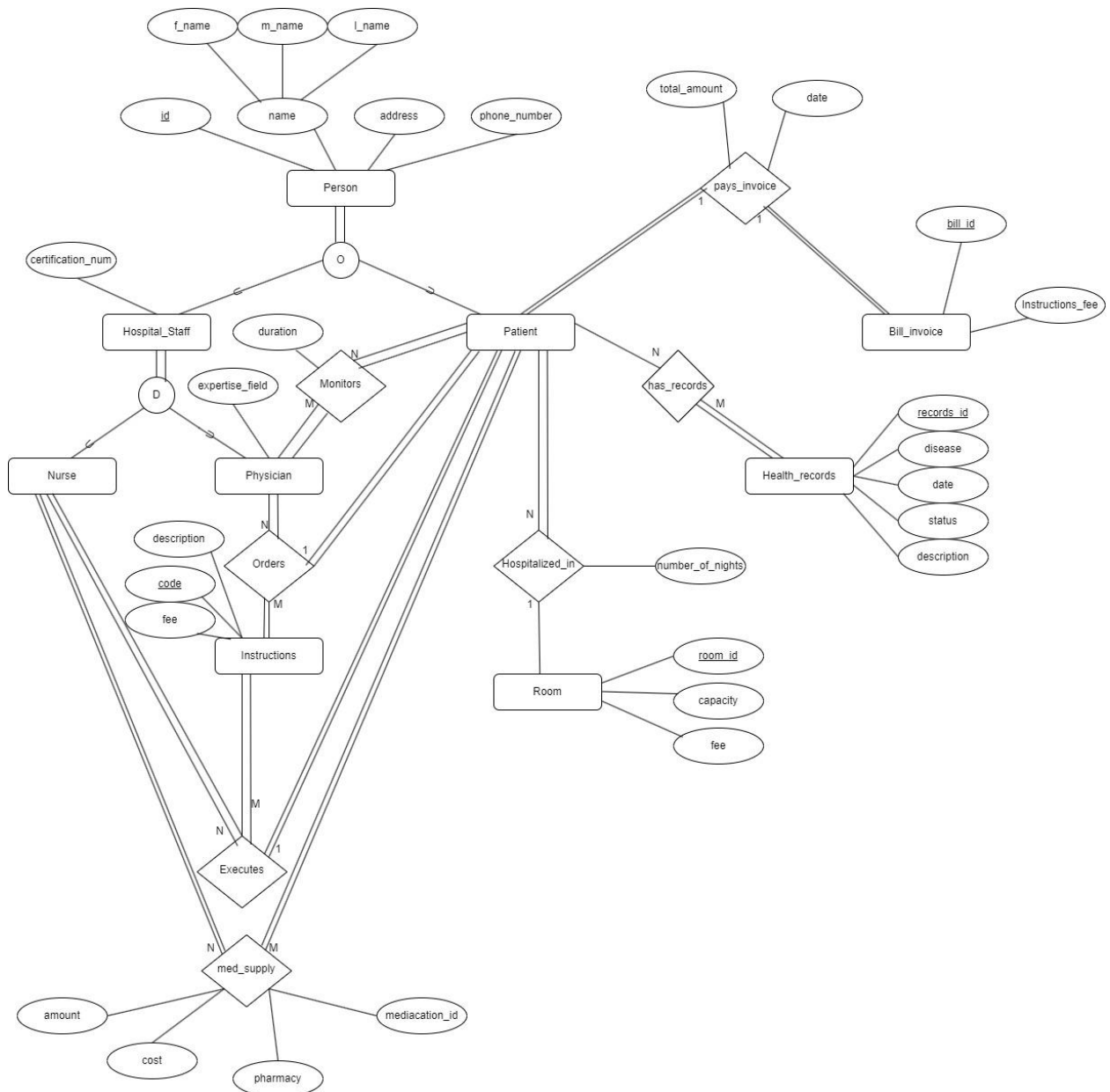
Report

Group Members: Dimitar Gjorgievski, Wajahat Ansari

Assumptions

- Each person occupying the hospital has a unique ID, regardless of whether they are a patient or staff member. The way the persons are distinctive is by the certification number. Patients by default have a NULL value, while physicians have an even certification number, and nurses have an odd certification number. Each new physician that enters the hospital database by default will be a General Physician, unless his/her expertise field is specifically stated or later updated. Multiple Physicians have the responsibility to monitor multiple patients. Also, multiple physicians have the authority to give out multiple instructions that are specific to one patient, and no physician can give the same instruction to the same patient twice. Likewise, multiple nurses have the duty to execute multiple instructions to a single patient, and no nurse can execute the same instruction to the same patient twice. Each instruction of each patient is accumulated into a single bill, and the total amount of the invoice the patient is required to pay is the total sum of all instruction fees plus the fee of spending a number of nights in a given room. Also, the hospital has a database of multiple health diagnoses and some of those are assigned to patients. If the hospital encounters a patient with a new unique diagnosis not encountered in the hospital before, that diagnosis will be stored into the health_records database of the hospital.

(E)ERD



Relations and keys

- person (id, f_name, m_name, l_name, address, phone_number, certification_num)
primary key(id)
- physician (id, expertise_field)
primary key(id)
foreign key(id) references person (id)

- nurse (id)
primary key(id)
foreign key(id) references person (id)
- monitors (physician_id, patient_id, duration)
primary key(physician_id, patient_id)
foreign key(physician_id) references person(id)
foreign key(patient_id) references person(id)
- room (room_id, capacity, fee)
primary key(room_id)
- health_records (records_id, disease, date, status, description)
primary key(records_id)
- has_records (patient_id, records_id)
primary key(patient_id, records_id)
foreign key(patient_id) references person (id)
foreign key(records_id) references health_records(records_id)
- instructions (code, description, fee)
primary key(code)
- orders (physician_id, patient_id, inst_code)
primary key (physician_id, patient_id, inst_code)
foreign key (physician_id) references person (id)
foreign key(patient_id) references person (id)
foreign key(inst_code) references instructions (code)

- executes (nurse_id, patient_id, inst_code)

 primary key (nurse_id, patient_id, inst_code)

 foreign key (nurse_id) references person (id)

 foreign key(patient_id) references person (id)

 foreign key(inst_code) references instructions (code)

- med_supply (medication_id, nurse_id, patient_id, amount, cost, pharmacy)

 primary key (nurse_id, patient_id)

 foreign key (nurse_id) references person (id)

 foreign key(patient_id) references person (id)

- patient (patient_id, room_id, nights_hospitalized)

 primary key(patient_id)

 foreign key (patient_id) references person (id)

 foreign key(room_id) references room (room_id)

- bill (bill_id, instructions_fee)

 primary key(bill_id)

- pays_invoice (patient_id, bill_id, room_id, nights_hospitalized, instructions_fee, total_amount, date)

 primary key(patient_id, bill_id)

 foreign key (patient_id) references person (id)

 foreign key (bill_id) references bill (bill_id)

 foreign key(room_id) references room (room_id)

Views

Description: This view outputs all the people in the person category that are patients

```
CREATE VIEW full_patients_info AS

SELECT p.id AS 'Patient ID', CONCAT(p.f_name, ' ', p.m_name, ' ', p.l_name) AS 'Name', p.address
AS 'Address', p.phone_number AS 'Phone Number', pa.room_id AS 'Room'

FROM person p, patient pa

WHERE p.id=pa.patient_id;
```

Description: This view outputs all the people in the person category that are physicians

```
CREATE VIEW full_physician_info AS

SELECT p.id AS 'Physician ID', CONCAT(p.f_name, ' ', p.m_name, ' ', p.l_name) AS 'Name',
p.address AS 'Address', p.phone_number AS 'Phone Number', p.certification_num AS
'Certification Number', ph.expertise_field AS 'Expertise'

FROM person p, physician ph

WHERE p.id=ph.id;
```

Description: This view outputs all the people in the person category that are nurses

```
CREATE VIEW full_nurse_info AS

SELECT p.id AS 'Nurse ID', CONCAT(p.f_name, ' ', p.m_name, ' ', p.l_name) AS 'Name', p.address
AS 'Address', p.phone_number AS 'Phone Number', p.certification_num AS 'Certification
Number'

FROM person p, nurse n

WHERE p.id=n.id;
```

Description: This view outputs rooms, their capacity, and the current occupancy. This allows the hospital to see how they can arrange their patients and in which available room.

```
CREATE VIEW room_occupancy AS

SELECT r.room_id AS 'Room ID', r.capacity AS 'Capacity', COUNT(p.patient_id) AS 'Current
Occupancy'

FROM room r

LEFT JOIN patient p ON r.room_id = p.room_id

GROUP BY r.room_id, r.capacity;
```

Description: This view outputs each patient's total payment due. This helps the hospital to check on the current amount each patient owes and also see if additional fees are being added to the total amount.

```
CREATE VIEW patient_payment AS

SELECT p.id AS 'Patient ID', CONCAT(p.f_name, ' ', p.m_name, ' ', p.l_name) AS 'Name', pp.bill_id
AS 'BILL ID', pp.total_amount AS 'Payment Due'

FROM person p, pays_invoice pp

WHERE p.id = pp.patient_id;
```

Triggers

Description: This trigger is activated each time a person with an odd certification number is added. This helps the hospital to automatize the nurse database, so they don't have to manually insert each nurse

```
DELIMITER //

CREATE TRIGGER insert_nurse

AFTER INSERT ON person

FOR EACH ROW

BEGIN

    DECLARE certification_num INT;
```

```
SET certification_num = NEW.certification_num % 2;

IF certification_num = 1 THEN
    INSERT INTO nurse (id) VALUES (NEW.id);
END IF;
END;
//
DELIMITER ;
```

Description: This trigger is activated each time a person with an even certification number is added. This helps the hospital to automatize the physician database, so they don't have to manually insert each physician.

```
DELIMITER //

CREATE TRIGGER insert_physician
AFTER INSERT ON person
FOR EACH ROW
BEGIN
    DECLARE certification_num INT;
    SET certification_num = NEW.certification_num % 2;

    IF certification_num = 0 THEN
        INSERT INTO physician (id, expertise_field) VALUES (NEW.id, 'General Physician');
    END IF;
END;
//
DELIMITER ;
```

Description: This trigger is activated each time a person with no certification number is added. This helps the hospital to automatize the patient database, so they don't have to manually insert each patient.

```
DELIMITER //

CREATE TRIGGER insert_patient

AFTER INSERT ON person

FOR EACH ROW

BEGIN

    IF NEW.certification_num IS NULL THEN

        INSERT INTO patient (patient_id) VALUES (NEW.id);

    END IF;

END;

//

DELIMITER ;
```

Description: This trigger is activated each time a new instruction a physician assigns to a patient. The fee of the new instruction is added to the total amount the patient currently owes. This helps to update the patient's bill and invoice.

```
DELIMITER //

CREATE TRIGGER update_invoice

AFTER INSERT ON orders

FOR EACH ROW

BEGIN

    DECLARE instr_fee INT;

    DECLARE b_id INT;

    SELECT fee INTO instr_fee
```



```

FROM instructions

WHERE code = NEW.inst_code;

UPDATE pays_invoice

SET instructions_fee = instructions_fee + instr_fee,

    total_amount = total_amount + instr_fee

WHERE patient_id = NEW.patient_id;

    SELECT bill_id INTO b_id

FROM pays_invoice

    WHERE patient_id = NEW.patient_id;

UPDATE bill

SET instructions_fee = instructions_fee + instr_fee

WHERE b_id = bill_id;

END;

//

DELIMITER ;

```

QUERIES AND DESCRIPTION:

Query 1:

```

1  -- JOIN QUERIES START HERE--
2
3  -- print which patient requires which instructions or medicine
4  SELECT CONCAT(pe.f_name, ' ', pe.m_name, ' ', pe.l_name) patient_name, i.description AS instructions,m.medication_id,m.cost AS medication_cost FROM patient p
5  JOIN executes e ON p.patient_id = e.patient_id JOIN instructions i ON e.inst_code = i.code JOIN med_supply m ON e.nurse_id = m.nurse_id AND p.patient_id = m.patient_id
6  JOIN person pe ON p.patient_id = pe.id;
7
8
9

```

patient_name	instructions	medication...	medication_c...
Johnson S John	Chemotherapy	4	80
Johnson S John	Bed rest	4	80
Doe M Jane	Care IV	1	50
Doe M Jane	Bed rest	1	50
Ahmad B Kazim	Physical therapy	2	30
Ahmad B Kazim	Bed rest	2	30

```

7
8
9 -- printing name of physician with its name of patient
10 SELECT CONCAT(phy.f_name, ' ', phy.m_name, ' ', phy.l_name) AS physician_name,
11        CONCAT(pat.f_name, ' ', pat.m_name, ' ', pat.l_name) AS patient_name
12 FROM monitors m JOIN person phy ON m.physician_id = phy.id JOIN person pat ON m.patient_id = pat.id;
13
14 -- printing all the bill with patient name

```

100% 1/8

Result Grid Filter Rows: Search Export

physician_name	patient_name
James M Alex	Doe M Jane
Phillips C Bruce	Johnson S John
Phillips C Bruce	Ahmad B Kazim

Result Grid Form Editor

```

13
14
15 -- printing all the bill with patient name
16 SELECT CONCAT(p.f_name, ' ', p.m_name, ' ', p.l_name) AS patient_name, b.bill_id, b.instructions_fee, (b.instructions_fee +
17        (SELECT SUM(i.fee) FROM executes e JOIN instructions i ON e.inst_code = i.code WHERE e.patient_id = p.id)) AS total_amount,
18        p.date FROM pays_invoice p JOIN person p ON p.patient_id = p.id JOIN bill b ON p.bill_id = b.bill_id;
19
20

```

100% 1/14

Result Grid Filter Rows: Search Export

patient_name	bill_id	instructions_f...	total_amount	date
Doe M Jane	1	50	180	2023-09-22
Johnson S John	2	140	280	2024-01-31
Ahmad B Kazim	3	110	220	2023-11-08

Result Grid Form Editor

```

19
20
21 -- printing the pharmacy the patient will get the medicine from including the description of instruction
22 SELECT CONCAT(p.f_name, ' ', p.m_name, ' ', p.l_name) AS patient_name, i.description AS instruction_description, ms.pharmacy FROM patient pat
23 JOIN person p ON pat.patient_id = p.id JOIN pays_invoice b ON pat.patient_id = b.patient_id JOIN instructions i ON b.bill_id = i.code
24 JOIN room r ON b.room_id = r.room_id JOIN med_supply ms ON pat.patient_id = ms.patient_id;
25
26

```

100% 1/25

Result Grid Filter Rows: Search Export

patient_name	instruction_description	pharmacy
Johnson S John	Physical therapy	OVS Pharmacy
Johnson S John	Physical therapy	OVS Pharmacy
Doe M Jane	Give IV	Walgreens Pharmacy
Doe M Jane	Give IV	Walgreens Pharmacy
Ahmad B Kazim	Chemotherapy	OVS Pharmacy
Ahmad B Kazim	Chemotherapy	OVS Pharmacy

Result Grid Form Editor

```

25
26
27
28 -- printing all patient's health records
29 SELECT CONCAT(p.f_name, ' ', p.m_name, ' ', p.l_name) AS patient_name, hr.disease, hr.date, hr.status, hr.description
30 FROM health_records hr JOIN has_records hrp ON hr.records_id = hrp.records_id JOIN person p ON hrp.patient_id = p.id;
31
32

```

100% 1/26

Result Grid Filter Rows: Search Export

patient_name	disease	date	status	description
Doe M Jane	Fever	2023-07-10	Positive	High fever
Ahmad B Kazim	Fracture	2023-07-11	Recovered	fractured arm
Ahmad B Kazim	Headache	2023-07-12	Recovered	severe headache
Johnson S John	Cancer	2023-07-12	Positive	Second Stage

Result Grid Form Editor

```

34 -- -- Aggregation QUERIES START HERE--
35
36 -- -- Finding the biggest room
37 SELECT room_id, MAX(capacity) AS max_room_capacity FROM room group by room_id order by max_room_capacity DESC limit 1;
38
39

```

130% 1/33

Result Grid Filter Rows: Search Export Fetch rows:

room_id	max_room_capacity
3	3

Result Grid Form Editor

```

38
39
40 -- -- finding Total number of patients
41 SELECT COUNT(*) AS total_patients FROM patient;
42
43
44 -- -- finding the patient with highest bill

```

130% 1/43

Result Grid Filter Rows: Search Export

total_patients
3

Result Grid Form Editor

```

43
44 -- -- finding the patient with highest bill
45 * SELECT CONCAT(p.f_name, ' ', p.m_name, ' ', p.l_name) AS patient_name, MAX(b.instructions_fee) AS highest_bill FROM pays_invoice pi
46 JOIN person p ON pi.patient_id = p.id JOIN bill b ON pi.bill_id = b.bill_id GROUP BY p.id, p.f_name, p.m_name, p.l_name ORDER BY
47 highest_bill DESC LIMIT 1;
48
49 -- -- printing how many orders for a patient
50
130% 0 27:47

```

Result Grid

patient_name	highest_bill
Doe M Jane	90

```

48
49 -- -- printing how many orders for a patient
50 * SELECT CONCAT(p.f_name, ' ', p.m_name, ' ', p.l_name) AS patient_name, COUNT(o.patient_id) AS num_orders
51 FROM patient pat JOIN person p ON pat.patient_id = p.id JOIN orders o ON pat.patient_id = o.patient_id
52 GROUP BY patient_name;
53
54
130% 0 1:48

```

Result Grid

patient_name	num_orders
Johnson S John	2
Doe M Jane	2
Ahmad B Kazm	2

```

54 Execute the selected portion of the script or
55 everything, if there is no selection
56 -- -- printing the count of executed instructions by each nurse for each patient
57 * SELECT CONCAT(n.f_name, ' ', n.m_name, ' ', n.l_name) AS nurse_name,
58 CONCAT(p.f_name, ' ', p.m_name, ' ', p.l_name) AS patient_name, COUNT(*) AS num_of_executed_instructions FROM executes e
59 JOIN person n ON e.nurse_id = n.id JOIN person p ON e.patient_id = p.id GROUP BY n.f_name, n.m_name, n.l_name, p.f_name, p.m_name, p.l_name;
60
130% 0 1:54

```

Result Grid

nurse_name	patient_name	num_of_executed_instructions
Andrew A Ben	Doe M Jane	2
Andrew A Ben	Johnson S John	2
All R Borhan	Ahmad B Kazm	2

```

61 -- -- Nested QUERIES START HERE--
62
63 -- -- Printing all physician name , nurse name and patient name
64 * SELECT DISTINCT (SELECT CONCAT(f_name, ' ', m_name, ' ', l_name) FROM person WHERE id = monitors.physician_id) AS physician_name,
65 (SELECT CONCAT(f_name, ' ', m_name, ' ', l_name) FROM person WHERE id = executes.nurse_id) AS nurse_name,
66 (SELECT CONCAT(f_name, ' ', m_name, ' ', l_name) FROM person WHERE id = executes.patient_id) AS patient_name
67 FROM monitors JOIN executes ON monitors.patient_id = executes.patient_id;
68
130% 0 1:58

```

Result Grid

physician_name	nurse_name	patient_name
James M Alex	Andrew A Ben	Doe M Jane
Phillip C Bruce	Andrew A Ben	Johnson S John
Phillip C Bruce	All R Borhan	Ahmad B Kazm

```

68
69 -- -- Finding physician expertise and printing with its name
70 * SELECT (SELECT CONCAT(f_name, ' ', m_name, ' ', l_name) FROM person WHERE id = physician.id) AS physician_name,
71 expertise_field FROM physician;
72
130% 0 1:58

```

Result Grid

physician_name	expertise_field
James M Alex	Cardiology
Phillip C Bruce	Neurology

```

72
73 -- -- Printing which physician prescribed the instructions for which patient
74 * SELECT (SELECT CONCAT(f_name, ' ', m_name, ' ', l_name) FROM person WHERE id = o.physician_id) AS physician_name,
75 (SELECT CONCAT(f_name, ' ', m_name, ' ', l_name) FROM person WHERE id = o.patient_id) AS patient_name,
76 (SELECT description FROM instructions WHERE code = o.inst_code) AS instruction_description
77 FROM orders o;
78
130% 0 1:72

```

Result Grid

physician_name	patient_name	instruction_description
James M Alex	Doe M Jane	Give IV
James M Alex	Doe M Jane	Red rest
Phillip C Bruce	Johnson S John	Chemotherapy
Phillip C Bruce	Johnson S John	Red rest
Phillip C Bruce	Ahmad B Kazm	Physical therapy
Phillip C Bruce	Ahmad B Kazm	Red rest

```

80 -- -- Printing which room assigned to which patient with its room capacity
81 Execute the selected portion of the script or
82 everything, if there is no selection (f_name, ' ', m_name, ' ', l_name) FROM person WHERE id = p.patient_id) AS patient_name,
83 (SELECT room_id FROM room WHERE room_id IN (SELECT room_id FROM patient WHERE patient_id = p.patient_id)) AS assigned_room_id,
84 (SELECT capacity FROM room WHERE room_id IN (SELECT room_id FROM patient WHERE patient_id = p.patient_id)) AS capacity,
85 (SELECT fee FROM room WHERE room_id IN (SELECT room_id FROM patient WHERE patient_id = p.patient_id)) AS room_fee FROM patient p;
86
130% 1:85
Result Grid  Filter Rows: Search Export:


| patient_name   | assigned_room_id | capacity | room_fee |
|----------------|------------------|----------|----------|
| Johnson B John | 1                | 2        | 150      |
| Doe M Jane     | 2                | 1        | 200      |
| Ahmad B Kazm   | 2                | 1        | 200      |


87
88 -- -- printing phone of physician and nurse
89 SELECT CONCAT(f_name, ' ', m_name, ' ', l_name) AS name, phone_number FROM person WHERE id IN (SELECT id FROM nurse) OR id IN (SELECT id FROM physician);
90
130% 1:86
Result Grid  Filter Rows: Search Export:


| name             | phone_number |
|------------------|--------------|
| James M Alex     | 123456       |
| Andrew A Ben     | 456789       |
| Prakash C Dancer | 789432       |
| Ali R Borhan     | 789027       |


```

Transactions

Description: This transaction creates a new patient record, a new health record for the patient, and orders an instruction for the patient. The instruction is then executed by a nurse.

START TRANSACTION;

INSERT INTO person (id, f_name, m_name, l_name, address, phone_number, certification_num)VALUES

(45, 'Debra', 'M', 'Collins', '234 Burr Ridge', '123456', NULL);

INSERT INTO patient (patient_id, room_id, nights_hospitalized)

VALUES (45, 3, 3);

INSERT INTO health_records (records_id, disease, date, status, description)

VALUES (123, 'COVID-19', '2023-07-28', 'Active', 'Patient was diagnosed with COVID-19.');

INSERT INTO instructions (code, description, fee)VALUES

(1001, 'Put on ventilator', '50');

```
INSERT INTO orders (physician_id, patient_id, inst_code)
VALUES (1, 45, 1001);
```

```
INSERT INTO executes (nurse_id, patient_id, inst_code)
VALUES (2, 45, 1001);
```

```
COMMIT;
```

Description: This transaction retrieves the patient's bill and then updates the total amount of the bill by adding a tax worth 100.

```
START TRANSACTION;
```

```
SELECT bill_id, instructions_fee, total_amount
FROM pays_invoice
WHERE patient_id = 4;
```

```
UPDATE pays_invoice
SET total_amount = total_amount + 100
WHERE patient_id = 4;
```

```
COMMIT;
```