Hospital Database Management System

Frida Ekner

2024-02-20

1 Introduction

Well-designed hospital databases are important for the collection of patient information and achieving efficiency due to the processing of data, thus giving the healthcare provider relevant information. The data will require a structured ordering and a level of manipulation to make it easier to use. Healthcare database management systems are a best way to monitor and improve the value of the healthcare services and the well-being of the patients, which can be achieved by making information available on access to the appropriateness, effectiveness and quality of healthcare services and providers. The database system created in this project should be able to query basic information about the hospital employees such as doctor ID number, name, address, and phone number, query information about departments and patient assignments for each employee and store patient information for use by employees, such as patient ID, name, location, assigned doctor, prescriptions, and medical records to generate reports.

1.1 User requirements

User requirements encompass the essential needs of individuals and the tasks they aim to accomplish upon the completion of a database. Essentially, it involves outlining the functionalities users can execute within the database management system. In the context of this hospital database, the users are the staff, and the subsequent list outlines specific tasks to be carried out by the hospital's staff.

- Patient registration
- Patient check out
- Generation of patient information reports
- Storage of mandatory patient information
- Updating patient information

1.2 Non-functional requirements

Non-functional requirements help in understanding how the database system should behave. Below are some of the most important non-functional requirements for the hospital database.

- **Security:** Ensuring security is paramount for a hospital database, preventing unauthorized alterations to the information. This involves setting appropriate permissions or administrative rights, limiting data modification access to hospital staff exclusively.
- **Performance:** The system's response time for creating/changing records should be swift, and the database system must operate seamlessly without causing any slowdown.
- **Maintainability:** In a hospital database, having reliable backup mechanisms is imperative due to the constant influx of patients and the busy nature of the environment.
- **Reliability:** The hospital database system should maintain constant availability, ensuring uninterrupted access at all times.

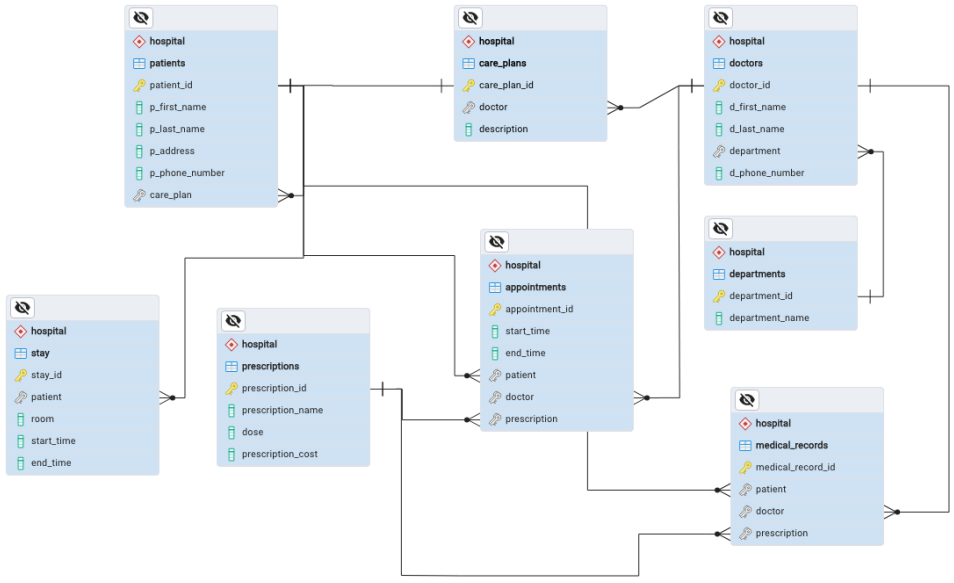
1.3 Operational regulations

Operational regulations outline the organizational guidelines that govern the management of information within a company's database. They articulate the perspective of the business on the utilization of its data. The designated business rules for the hospital database include, but are not limited to the following.

- Null values are not allowed in the date columns.
- Each doctor will be allocated a distinct identification number.
- The departments are categorized based on medical requirements.
- Each doctor is exclusively assigned to one department.
- A doctor is permitted to be assigned to multiple patients.
- Patients have the option to schedule appointments with doctors.
- When required, patients can be admitted to the hospital.
- Patients may receive prescribed medication when deemed necessary.

2 Entity-Relationship Diagram (ERD)

An entity-relationship diagram serves as a visual representation of the entities and their relationships within the database structure. Its primary objective is to offer a simplified conceptual overview of the intricate database design. Each entity in the diagram collects and stores data, while the relationships illustrate the connections between the data. In the Hospital network's ERD, the foundational table is Patients, followed by Doctors and Departments. This then branches into Appointments, ultimately connecting to Medical records through Prescriptions. The Patient table further extends its connections, branching into Stay and Care plans.



3 Data entry

3.1 Departments

```
CREATE TABLE hospital.departments (
    department_ID char(4) PRIMARY KEY,
    department_name varchar(30)
)
```

```
INSERT INTO hospital.departments (department_ID, department_name) VALUES
(3001, 'Cardiology'),
(3002, 'Emergency'),
(3003, 'Oncology'),
(3004, 'Psychiatric'),
(3005, 'Urology'),
(3006, 'Surgery'),
(3007, 'Radiology'),
(3008, 'Accident'),
(3009, 'Anesthetic'),
(3010, 'Neurology');
```

3.2 Doctors

```
CREATE TABLE hospital.doctors (
    doctor_ID char(4) PRIMARY KEY,
    d_first_name varchar(20),
    d_last_name varchar(20),
    department char(4) REFERENCES hospital.departments(department_ID),
    d_phone_number char(10)
)
```

```
INSERT INTO hospital.doctors (doctor_ID, d_first_name, d_last_name, department, d_phone_number) VALUES
(2001, 'Tobias', 'Moberg', 3003, 0736659878),
(2002, 'Matilda', 'Oscarsson', 3007, 07023764333),
(2003, 'Måns', 'Lindgren', 3004, 0733774980),
(2004, 'Josefin', 'Ohlin', 3008, 0702213535),
(2005, 'Gustav', 'Gustafsson', 3009, 0700198984),
(2006, 'Fatima', 'Juhlin', 3009, 0702354547),
(2007, 'Emilia', 'Nordgren', 3001, 07044498673),
(2008, 'Johan', 'Krok', 3010, 0730967930),
(2009, 'Stella', 'Nilsson', 3002, 0732255343),
(2010, 'Joar', 'Andersson', 3006, 0707745458);
```

3.3 Care plans

```
CREATE TABLE hospital.care_plans (
    care_plan_ID char(4) PRIMARY KEY,
    doctor char(4) REFERENCES hospital.doctors(doctor_ID),
    description varchar(100)
)
```

```
INSERT INTO hospital.care_plans (care_plan_ID, doctor, description) VALUES
(5001, 2007, 'Radical prostatectomy every forth week'),
(5002, 2008, 'Daily exercise and if no improvements surgery'),
```

```
(5003, 2004, 'Therapy every thursday'),
(5004, 2006, 'Cholecystectomy'),
(5005, 2001, 'Upper GI Endoscopy '),
(5006, 2003, 'Chemotherapy'),
(5007, 2010, 'Weekly check ups'),
(5008, 2006, 'Strict diet before colostomy'),
(5009, 2008, 'Tests, x-rays and scans once a year for check up'),
(5010, 2002, 'Dialysis three times a week');
```

3.4 Patients

```
CREATE TABLE hospital.patients (
    patient_ID char(4) PRIMARY KEY,
    p_first_name varchar(20),
    p_last_name varchar(20),
    p_address varchar(50),
    p_phone_number char(10),
    care_plan char(4) REFERENCES hospital.care_plans(care_plan_ID)
)
```

```
INSERT INTO hospital.patients (patient_ID, p_first_name, p_last_name, p_address, p_phone_number, care_pl
(1001, 'Linnea', 'Johansson', 'Ringvägen 10B, 17070 Solna', 0736659878, 5004),
(1002, 'Karl', 'Lindberg','Skogsstigen 2, 18400 Åkersberga', 0704775925, 5005),
(1003, 'Anna', 'Nordin','Bergsgatan 72, 19124 Sollentuna', 0702264988, 5008),
(1004, 'Erik', 'Jonsson','Björkvägen 13, 12333 Farsta', 0731181901, 5001),
(1005, 'Sofi', 'Wallin','Odengatan 20, 10044 Stockholm', 0705301744, 5002),
(1006, 'Mikael', 'Holm','Humlegatan 43, 10215 Stockholm', 0702730032, 5007),
(1007, 'Anna', 'Söderström','Stationsgatan 11, 18037 Täby', 0733989881, 5009),
(1008, 'Johan', 'Andersson','Granstigen 7, 14731 Tumba', 0735535010, 5003),
(1009, 'Clara', 'Viklund','Ängägen 9B, 12524 Älvsjö', 0702320649, 5010),
(1010, 'Jan', 'Lindgren','Hornsgatan 27, 10041 Stockholm', 0703203348, 5006);
```

3.5 Prescriptions

```
CREATE TABLE hospital.prescriptions (
    prescription_ID char(4) PRIMARY KEY,
    prescription_name varchar(30),
    dose char(6),
    prescription_cost char(6)
)
```

```
INSERT INTO hospital.prescriptions (prescription_ID, prescription_name, dose, prescription_cost) VALUES
(6001, 'Acetaminophen', 160, 20),
(6002, 'Xanax', 0.5, 50.9),
(6003, 'Lisinopril', 30, 9.9),
(6004, 'Augmentin', 250, 69),
(6005, 'Ibuprofen', 800, 49.7),
(6006, 'Humira', 40, 32),
(6007, 'Methotrexate', 2.5, 55),
(6008, 'Tramadol', 100, 9.8),
(6009, 'Amoxicillin', 600, 39.9),
(6010, 'Farxiga', 10, 22);
```

3.6 Appointments

```
CREATE TABLE hospital.appointments (
    appointment_ID char(4) PRIMARY KEY,
    start_time timestamp,
    end_time timestamp,
    patient char(4) REFERENCES hospital.patients(patient_ID),
    doctor char(4) REFERENCES hospital.doctors(doctor_ID),
    prescription char(4) REFERENCES hospital.prescriptions(prescription_ID)
)
```

```
INSERT INTO hospital.appointments (appointment_ID, start_time, end_time, patient, doctor, prescription)
(4001, '1953-01-08 11:00:00', '1953-01-08 11:50:00', 1001, 2007, 6007),
(4002, '1952-12-30 08:00:00', '1952-12-30 10:00:00', 1010, 2001, 6010),
(4003, '1953-01-08 15:30:00', '1953-01-08 16:00:00', 1005, 2004, 6003),
(4004, '1953-01-08 14:00:00', '1953-01-08 15:10:00', 1002, 2008, 6006),
(4005, '1953-01-08 10:00:00', '1953-01-08 11:00:00', 1003, 2007, 6002),
(4006, '1953-02-02 08:00:00', '1953-02-02 08:40:00', 1001, 2009, 6009),
(4007, '1953-02-05 09:20:00', '1953-02-05 10:00:00', 1004, 2004, 6004),
(4008, '1953-01-08 12:00:00', '1953-01-08 13:00:00', 1008, 2006, 6001),
(4009, '1953-01-11 14:30:00', '1953-01-11 15:10:00', 1003, 2007, 6005),
(4010, '1953-03-08 13:00:00', '1953-03-08 13:30:00', 1009, 2003, 6008);
```

3.7 Medical records

```
CREATE TABLE hospital.medical_records (
    medical_record_ID char(4) PRIMARY KEY,
    patient char(4) REFERENCES hospital.patients(patient_ID),
    doctor char(4) REFERENCES hospital.doctors(doctor_ID),
    prescription char(4) REFERENCES hospital.prescriptions(prescription_ID)
)
```

```
INSERT INTO hospital.medical_records (medical_record_ID, patient, doctor, prescription) VALUES
(7001, 1001, 2007, 6007), (7002, 1003, 2002, 6004), (7003, 1007, 2002, 6005),
(7004, 1010, 2001, 6010), (7005, 1005, 2008, 6009), (7006, 1005, 2001, 6003),
(7007, 1005, 2004, 6003), (7008, 1001, 2005, 6010), (7009, 1009, 2008, 6006),
(7010, 1002, 2008, 6005), (7011, 1007, 2006, 6002), (7012, 1005, 2010, 6002),
(7013, 1003, 2007, 6002), (7014, 1004, 2007, 6003), (7015, 1002, 2003, 6001),
(7016, 1001, 2009, 6009), (7017, 1004, 2006, 6004), (7018, 1003, 2002, 6010),
(7019, 1004, 2004, 6004), (7020, 1010, 2010, 6008), (7021, 1010, 2007, 6008),
(7022, 1008, 2006, 6001), (7023, 1008, 2009, 6007), (7024, 1007, 2005, 6005),
(7025, 1003, 2007, 6005), (7026, 1002, 2002, 6007), (7027, 1010, 2007, 6007),
(7028, 1009, 2003, 6008), (7029, 1008, 2004, 6005), (7030, 1006, 2005, 6004);
```

3.8 Stay

```
CREATE TABLE hospital.stay (
    stay_ID char(4) PRIMARY KEY,
    patient char(4) REFERENCES hospital.patients(patient_ID),
    room char(3),
    start_time timestamp,
    end_time timestamp
)
```

```
INSERT INTO hospital.stay (stay_ID, patient, room, start_time, end_time) VALUES
(8001, 1003, 912, '1952-10-28 10:50:00', '1952-11-04 12:00:00'),
(8002, 1004, 922, '1952-12-15 11:00:00', '1952-12-17 08:00:00'),
(8003, 1004, 912, '1952-11-21 08:00:00', '1952-11-23 10:30:00'),
(8004, 1001, 902, '1952-11-09 12:30:00', '1952-11-09 19:00:00'),
(8005, 1009, 928, '1952-12-04 17:00:00', '1952-12-08 10:00:00'),
(8006, 1002, 900, '1953-01-13 08:00:00', '1953-01-13 16:20:00'),
(8007, 1004, 920, '1952-12-25 07:30:00', '1953-01-02 09:00:00'),
(8008, 1001, 904, '1952-10-11 08:00:00', '1952-10-16 14:00:00'),
(8009, 1004, 912, '1953-01-07 22:10:00', '1953-01-09 09:30:00'),
(8010, 1010, 904, '1952-09-30 16:00:00', '1952-10-30 10:00:00');
```

4 Data retrieval

4.1 Busiest day of the week

```
SELECT CASE date_part('isodow', start_time)
    WHEN 1 THEN 'Sunday'
    WHEN 2 THEN 'Monday'
    WHEN 3 THEN 'Tuesday'
    WHEN 4 THEN 'Wednesday'
    WHEN 5 THEN 'Thursday'
    WHEN 6 THEN 'Friday'
    WHEN 7 THEN 'Saturday' END AS busiest_day
FROM hospital.appointments
GROUP BY busiest_day
ORDER BY count(*) DESC LIMIT 1;
```

Table 1: Busiest day

busiest_day
Wednesday

4.2 Prescription most commonly prescribed

```
SELECT prescription_name AS most_common_prescription FROM hospital.prescriptions
WHERE prescription_id =
    (SELECT prescription FROM hospital.medical_records
     GROUP BY prescription
     ORDER BY count(prescription) DESC LIMIT 1);
```

Table 2: Prescription

most_common_prescription
Ibuprofen

4.3 Doctors, and their phone numbers, assigned to the Anesthitic department

```
SELECT (d_first_name ||' '|| d_last_name) AS doctor, d_phone_number AS phone_number
FROM hospital.doctors
WHERE department =
    (SELECT department_ID FROM hospital.departments
     WHERE department_name = 'Anesthetic')
```

Table 3: Doctors and their phone numbers

doctor	phone_number
Gustav Gustafsson	700198984
Fatima Juhlin	702354547

4.4 Patients who had an appointment on the 8th of January 1953, their doctor and their care plan

```
SELECT (p_first_name ||' '|| p_last_name) AS patient_name, (d_first_name ||' '|| d_last_name) AS doctor
FROM hospital.patients
    JOIN hospital.care_plans
    ON hospital.patients.care_plan = hospital.care_plans.care_plan_ID
    JOIN hospital.appointments
    ON hospital.care_plans.doctor = hospital.appointments.doctor
    JOIN hospital.doctors
    ON hospital.appointments.doctor = hospital.doctors.doctor_ID
WHERE DATE(start_time) = '1953-01-08'
GROUP BY patient_name, doctor_name, description
```

Table 4: Patients, their doctor and care plan

patient_name	doctor_name	description
Anna Nordin	Fatima Juhlin	Strict diet before colostomy
Anna Söderström	Johan Krok	Tests, x-rays and scans once a year for check up
Erik Jonsson	Emilia Nordgren	Radical prostatectomy every forth week
Johan Andersson	Josefin Ohlin	Therapy every thursday
Linnea Johansson	Fatima Juhlin	Cholecystectomy
Sofi Wallin	Johan Krok	Daily exercise and if no improvements surgery

4.5 Doctor with the longest queue times

```
SELECT (d_first_name ||' '|| d_last_name) AS busiest_doctor FROM hospital.doctors
WHERE doctor_id =
    (SELECT doctor FROM hospital.medical_records
    GROUP BY doctor
    ORDER BY count(*) DESC LIMIT 1);
```

Table 5: Busiest doctor

busiest_doctor
Emilia Nordgren

4.6 Patient that has spent most money on medications and the amount (displayed in dollars)

```
SELECT (p_first_name ||' '|| p_last_name) AS patient, total_cost FROM hospital.patients
JOIN(SELECT patient, SUM(medicin_cost) AS total_cost FROM
        (SELECT patient, (CAST(count(prescription) AS DOUBLE PRECISION)*
                          CAST(prescription_cost AS DOUBLE PRECISION))
                          AS medicin_cost FROM hospital.medical_records
        JOIN hospital.prescriptions
        ON hospital.prescriptions.prescription_ID = hospital.medical_records.prescription
        GROUP BY prescription_cost, patient)
    GROUP BY patient
    ORDER BY SUM(medicin_cost) DESC) AS costs
```

```
ON costs.patient = patients.patient_ID
ORDER BY total_cost DESC limit 1;
```

Table 6: Patient with highest costs

patient	total_cost
Anna Nordin	191.6

4.7 Patient that has stayed longest (including appointments) in the hospital and the time spent

```
SELECT (p_first_name ||' '|| p_last_name) AS patient, time_spent FROM hospital.patients
JOIN(SELECT COALESCE(patient_1, patient_2) AS patient, (COALESCE(total_stay,'0')
    + COALESCE(total_appointments,'0')) AS time_spent FROM
        (SELECT patient AS patient_1, SUM(end_time - start_time) AS total_stay
        FROM hospital.stay
        GROUP BY patient_1
        ORDER BY total_stay DESC) AS total_stay
    JOIN
        (SELECT patient as patient_2, SUM(end_time - start_time) AS total_appointments
        FROM hospital.appointments
        GROUP BY patient_2
        ORDER BY total_appointments DESC) AS total_time
    ON total_stay.patient_1 =  total_time.patient_2) AS time_spent
ON time_spent.patient = patients.patient_ID
ORDER BY time_spent DESC LIMIT 1;
```

Table 7: Patient with longest stay

patient	time_spent
Jan Lindgren	29 days 20:00:00