**Title**: Hospital Management System

**Faculty:** Science

**Department:** Computer Science and Mathematics

**Team members**: Haifa Sidani (ID: 202400882)

Yosr Najjar (ID: 202402846)

**Introduction:**

Medical care is one of the most important aspects in our daily life. As years pass, the population is increasing, and with every human being born, the data of the medical information is increasing. Where hospitals are the largest institutions that hold medical records, it is important to maintain this data for it being sensitive.

For that, we built a hospital database management system to manage departments, appointments, medical records and prescriptions, based on the treatments doctors were trained for and the state of patients.

**Objectives:**

* Nurses will be able to make appointments easily and professionally
* Schedules for physicians and nurses, based on appointments and work hours, will be more clear
* Managing departments easily
* Maintaining medical records safely
* Managing rooms for appointments and stays properly
* Managing medication and prescriptions information
* Looking up for available treatments
* Avoiding noisy data

**Database Design:**

**Entity-Relationship (ER) Diagram:**

* Physician:

A physician has an employee ID to be easily accessible with IDENTITY (11, 20), a full name (fname, midin, lname) that he/she will be addressed with, a position, work hours and a unique SSN.

* Nurse:

A nurse has an employee ID to be easily accessible with IDENTITY (14, 20), a full name (fname, midin, lname) that he/she will be addressed with, a position, work hours, a unique SSN and an attribute to see if the nurse I registered or not.

* Patient:   
  Every patient has a unique SSN, a full name (fname, midin, lname) to be addressed with, age, gender, a phone number and an address.
* Treatment:   
  every treatment has a unique code with IDENTITY (16, 20), a name and a cost.
* Department:   
  A department has a unique ID with IDENTITY (1, 1), and a name.
* Room:   
  Every room has a unique number and a type.
* Appointment:   
  Every appointment has a partial Id, and a start and end date time that concludes the duration.
* Medication:   
  Each medication has a unique code, a name, and a brand.
* Prescription:  
  Each prescription has a partial ID, the date prescribed and a dosage.

Relations:

* Every physician can have an assisting nurse or more, Every nurse can assist a physician or more
* Every physician belongs to one and only one department, and a department has at least one physician.
* Only one physician manages a department.
* Every treatment available in the hospital has at least one physician trained in it, and a physician is trained in at least one treatment.
* Many physicians can check up a patient, a physician can have up to many patients.
* An appointment never exists unless a physician appoints it and a patient makes it, and every patient can make up to many appointments and a physician can appoint up to many appointments as well.
* A prescription never exists unless a physician prescribes it and a medication is included in it after an appointment is done, a medication can exist in many prescriptions, and a physician can prescribe up to many prescriptions, and an appointment has one and only one prescription.
* A patient can take from one to many prescriptions, and one and only one patient takes a prescription.
* A nurse can attend at least one appointment, and an appointment has at least one nurse to attend.
* An appointment is located strictly in one room; a room can have an appointment located in it.
* A patient can have no stays or many stays, a room can have no stays or many stays.

**Relational model description**

* Prescription:

ID: (INT) partial key

mcode: (INT) partial key, foreign key referencing code from medication

PhyID: (INT) partial key, foreign key referencing ID from Physician

AppID: (INT) partial key, foreign key referencing ID from appointment

Pid: (INT) foreign key referencing SSN from patient

dose: (VARCHAR length 10) represents the dosage of the medication

date: (CURRENT TIMESTAMP) represents the date and time the prescription was prescribed

* medication:

code: (INT) PRIMARY KEY

mname: (VARCHAR length 100) name of medication

brand: (VARCHAR length 50) brand of medication

* assist: many-to-many relationship between nurse and physician

nID: (INT) partial key, foreign key referencing ID from nurse

phyID: (INT) partial key, foreign key referencing ID from physician

* Room:

num: (INT) Primary key

type: (VARCHAR of length 50) type of room

* stay:

rnum: (INT) partial key, foreign key referencing num from Room

pSSN: (INT) partial key, foreign key referencing SSN from patient

ststart: (DATETIME NOT NULL) day and time of stay start

stend: (DATETIME) day and time of stay at the end (discharge)

\*duration: not a physical quantity in the database, but rather a derived one in the external view.\*

* Nurse:

ID: (INT with IDENTITY seed = 14, auto increment by 20) PRIMARY KEY

SSN: (INT) UNIQUE attribute

fname: (VARCHAR of length 50 NOT NULL) first name of nurse

midin: (VARCHAR of length 1) middle initial of the nurse

lname: (VARCHAR of length 50 NOT NULL) last name of nurse

position: (VARCHAR of length 50) position of the nurse

registered: (BIT NOT NULL) 0 if the nurse is not registered, 1 otherwise

Work\_Hours: (VARCHAR of length 100 NOT NULL) structure is like (starting hour - ending hour)

* Physician:

ID: (INT with IDENTITY seed = 11, auto increment by 20) PRIMARY KEY

SSN: (INT) UNIQUE attribute

fname: (VARCHAR of length 50 NOT NULL) first name of physician

midin: (VARCHAR of length 1) middle initial of the physician

lname: (VARCHAR of length 50 NOT NULL) last name of physician

position: (VARCHAR of length 50) position of the physician

DiD: (INT) foreign key referencing ID from department (affiliation)

Work\_Hours: (VARCHAR of length 100 NOT NULL) structure is like (starting hour - ending hour)

* patient:

SSN: (INT) PRIMARY KEY

fname: (VARCHAR of length 50 NOT NULL) first name of patient

midin: (VARCHAR of length 1) middle initial of the patient

lname: (VARCHAR of length 50 NOT NULL) last name of patient

address: (VARCHAR of length 100) address of the patient

phone: (VARCHAR of length 15) phone number of the patient

age: (DATETIME) Date of birth of patient, actual age will be derived and displayed

gender: (VARCHAR of length 15) gender of patient

* have: many-to-many relationship between patient and physician

phyID: (INT) partial key, foreign key referencing ID from physician

pSSN: (INT) partial key, foreign key referencing SSN from patient

* Department:

ID: (INT with IDENTITY seed = 1, auto increment by 1) PRIMARY KEY

dname: (VARCHAR of length 50 NOT NULL) name of the department

manager: (INT) UNIQUE attribute, foreign key referencing ID from physician

* treatment:

code: (INT with IDENTITY seed = 16, and auto increment = 20) PRIMARY KEY

name: (VARCHAR of length 50) name of treatment

cost: (INT) cost/price of the treatment

trained\_in: many-to-many relationship between treatment and physician

tcode: (INT) partial key, foreign key referencing code from treatment

phyID: (INT) partial key, foreign key referencing ID from physician

trstart: (DATETIME NOT NULL) day and time of starting treatment training

trend: (DATETIME) day and time of end of treatment training

\*duration: not a physical quantity in the database, but rather a derived one in the external view.\*

* appointment:

ID: (INT) partial key

PhyID: (INT) partial key, foreign key referencing ID from Physician

Pid: (INT) partial key, foreign key referencing SSN from patient

nID: (INT) foreign key referencing ID from nurse

rnum: (INT) foreign key referencing num from Room

apstart: (DATETIME NOT NULL) day and time of starting appointment

apend: (DATETIME) day and time of end of appointment

\*duration: not a physical quantity in the database, but rather a derived one in the external view.\*

**Query Execution:**

**List of Questions:**

1. When is the physician available?
2. When is the nurse available?
3. When can the patient make an appointment?
4. What is the patient’s medical record?
5. What are the medications available?
6. What are the available treatments in the hospital and by who?
7. Where can the patient have his/her stay?
8. How much did the patient stay?
9. What are the existing departments and who is the manager?
10. What is the position of each physician?
11. What is the position of each nurse, and is the nurse registered?
12. What is the patient’s last prescription?
13. Who is going to participate in the appointment and what is the treatment?
14. What is the treatment and its cost?
15. How many physicians are there in each department?
16. How many nurse is registered?
17. What are the appointments records?
18. How to retrieve data about a specific patient (from name or SSN)?
19. How to find a specific treatment (from name)?
20. How to retrieve data about a specific medication (from name or brand)?

**Queries Selected:**

1)

SELECT a.ID, (n.Fname+ ' ' +n.Minit + ' ' + n.Lname) AS nurse\_name,(p.Fname+ ' ' + p.Minit+ ' ' + p.Lname) AS patient\_name, a.rnum, a.apstart

FROM appointment a

JOIN physician as phy ON phy.ID=a.phyID JOIN nurse as n ON n.ID = a.nID JOIN patient as p ON p.SSN = a.pID

WHERE a.apend IS NULL AND phy.ID = 10;

Select Id of appointment, names of assisting nurse and patient, the room number and the date and time of the appointment to be made.

Join the tables with appointment on primary keys and foreign keys and display the required data.

2)SELECT a.ID, (phy.Fname+ ' ' +phy.Minit + ' ' + phy.Lname) AS physician\_name,(p.Fname+ ' ' + p.Minit+ ' ' + p.Lname) AS patient\_name, a.rnum, a.apstart

FROM appointment a

JOIN physician as phy ON phy.ID=a.phyID JOIN nurse as n ON n.ID=a.nID JOIN patient as p ON p.SSN = a.pID

WHERE a.apend IS NULL AND phy.ID = 20;

Select Id of appointment, names of physician and patient, the room number and the date and time of the appointment to be made.

Join the tables with appointment on primary keys and foreign keys and display the required data.

3)SELECT a.ID, (n.Fname+ ' ' +n.Minit + ' ' + n.Lname) AS nurse\_name,(p.Fname+ ' ' + p.Minit+ ' ' + p.Lname) AS patient\_name, a.rnum, a.apstart

FROM appointment a

JOIN physician as phy ON phy.ID=a.phyID JOIN nurse as n ON n.ID=a.nID JOIN patient as p ON p.SSN = a.pID

WHERE a.apend IS NULL AND phy.ID = 30;

Select Id of appointment, names of assisting nurse and patient, the room number and the date and time of the appointment to be made.

Join the tables with appointment on primary keys and foreign keys and display the required data.

Then add an appointment using INSERT INTO.

4)SELECT a.ID, (phy.Fname+ ' ' +phy.Minit + ' ' + phy.Lname) AS physician\_name, (n.Fname+ ' ' +n.Minit + ' ' + n.Lname) AS nurse\_name,(p.Fname+ ' ' + p.Minit+ ' ' + p.Lname) AS patient\_name, a.rnum, a.apstart , a.apend

FROM appointment a

JOIN physician as phy ON phy.ID=a.phyID JOIN nurse as n ON n.ID=a.nID JOIN patient as p ON p.SSN = a.pID

WHERE a.apend IS NOT NULL AND p.SSN = 10;

Select Id of appointment, names of assisting physician, nurse and patient, the room number and the date and time of the appointment to be made (start and end).

Join the tables with appointment on primary keys and foreign keys and display the required data.

Set a condition to ensure that the appointment is done.

5)SELECT code, mname, brand

FROM MEDICATION;

Select all attributes from MEDICATION to see types of medications available

6) SELECT T.code, T.tname, T.cost, P.Fname AS Physician\_FirstName, P.Lname AS Physician\_LastName

FROM TRAINED\_IN Tr

JOIN PHYSICIAN P ON Tr.phyID = P.ID

JOIN TREATMENT T ON Tr.tcode = T.code;

Select code, name and cost from treatments and the physicians name to see available treatments and details

7)SELECT num, rtype

FROM room

WHERE num IN (

SELECT rnum

FROM appointment

WHERE apend IS NOT NULL

GROUP BY rnum

)

OR num IN(

SELECT rnum

FROM stay

WHERE stend IS NOT NULL

GROUP BY rnum

);

Select room attributes from room where there is no appointment in it or no stay in it

8)SELECT pSSN, DATEDIFF(DAY, ststart, stend) AS Stay\_Duration

FROM STAY

WHERE stend IS NOT NULL AND pSSN = 12345;

Select stay duration and ssn based on ssn.

9)SELECT D.ID, D.Dname, P.Fname AS Manager\_FirstName, P.Lname AS Manager\_LastName

FROM DEPARTMENT D

JOIN PHYSICIAN P ON D.manager = P.ID;

Select department number and name, and manager name based on a join on the manager id.

10)SELECT ID, Fname, Lname, position

FROM PHYSICIAN;

Show the list of physicians with their respective names, ids and positions.

11)SELECT ID, Fname, Lname, position, registered

FROM NURSE;

Show the list of nurses with their respective names, ids, positions and registration status.

12)SELECT TOP 1 P.ID, P.predate, M.mname, M.brand, P.dose

FROM PRESCRIPTION P

JOIN MEDICATION M ON P.mcode = M.code

WHERE P.pID = 12345

ORDER BY P.predate DESC;

Select the patient prescriptions by descending by the date it is prescribed.

13)SELECT A.ID AS Appointment\_ID,

PAT.Fname AS Patient\_FirstName, PAT.Lname AS Patient\_LastName,

PHY.Fname AS Physician\_FirstName, PHY.Lname AS Physician\_LastName,

N.Fname AS Nurse\_FirstName, N.Lname AS Nurse\_LastName,

T.tname AS Treatment\_Name

FROM APPOINTMENT A

JOIN PATIENT PAT ON A.pID = PAT.SSN

JOIN PHYSICIAN PHY ON A.phyID = PHY.ID

LEFT JOIN NURSE N ON A.nID = N.ID

LEFT JOIN TREATMENT T ON A.phyID = T.phyID

WHERE A.ID = 23;

Select names of parties participating in the appointment based on a certain id.

14)SELECT tname, cost

FROM TREATMENT;

Select treatments name and cost

15)SELECT D.Dname, COUNT(P.ID) AS Physician\_Count

FROM DEPARTMENT D

LEFT JOIN PHYSICIAN P ON D.ID = P.DID

GROUP BY D.Dname;

Select department names and number of physicians for each department.

16)SELECT COUNT(ID) AS Registered\_Nurses

FROM NURSE

WHERE registered = 1;

Count nurses registered based on registration attribute

17)SELECT a.ID, (phy.Fname+ ' ' +phy.Minit + ' ' + phy.Lname) AS physician\_name, (n.Fname+ ' ' +n.Minit + ' ' + n.Lname) AS nurse\_name,(p.Fname+ ' ' + p.Minit+ ' ' + p.Lname) AS patient\_name, a.rnum, a.apstart , a.apend,(p.Fname+ ' ' + p.Minit+ ' ' + p.Lname) AS patient\_name, a.rnum

FROM appointment a

JOIN physician as phy ON phy.ID=a.phyID JOIN nurse as n ON n.ID=a.nID JOIN patient as p ON p.SSN = a.pID

WHERE a.apend IS NOT NULL;

Select appointments information that are done

18)SELECT \*

FROM PATIENT

WHERE (Fname = ‘x’ AND Lname = 's')

OR SSN = 123456789;

Search for a certain patient based on what the user types.

19)SELECT \*

FROM TREATMENT

WHERE tname = '<Treatment\_Name>';

Search for a certain treatment based on what the user types.

20)SELECT \*

FROM MEDICATION

WHERE mname = '<Medication\_Name>' OR brand = '<Brand\_Name>';

Search for a certain medication based on what the user types.

**Website Prototype:**

**Overview:**

The website has a simple interface. User (nurse or physician) will be able to see his/her schedule and status. User will be able to see others schedule and status as well. User will be able to appoint appointments for patients based on available treatment(s) and physician’s schedule. User will be able to see information about each room, medications available, and patients’ medical record. User will be able to view information about departments available and treatments available.

**Diagram Representation:**

* structure of the website:
* User navigation flow:

**Challenges and Solutions**

* Bad choice of entities that makes the database more complex 🡪 solution: redesigning the ER-Diagram and making things clearer to obtained the final model.
* Choosing the data to be displayed, and make it user-friendly 🡪 solution: pretending to be the user and see the requirements.
* Key word RESTRICT did not work 🡪 use key word NO ACTION
* Adding more than one key constraint by altering the table once 🡪 failed, so we altered the table at every constraint to add

**Conclusion**

* Constructing a database is easy when you put clear goals and pretend to be the user.
* Everything should be taken into consideration to avoid invoking guidelines when building a database.
* Writing the requirements makes drawing the ER diagram and Relational Schema easier and lowers error rate.
* Revising your draft (ER diagram and Relational Schema) is a crucial step before building the database on a DBMS.