

Usha Mittal Institute of Technology

SNDT Women's University , Mumbai

REPORT ON :



HOSPITAL MANAGEMENT SYSTEM

Submitted by :

47- Madhuri G. Patil

49 – Durva P. Patkar

57 – Vaishnavi Rawate

Guided by :

Mrs. Kumud Wasnik

TABLE OF CONTENTS

Sr.No.	Topic	Signature
1.	Introduction	
2.	Objective	
3.	Problem Statement	
4.	ER Diagram	
5.	Schema Diagram	
6.	Database Table	
7.	Queries Performed	
8.	Conclusion	
9.	References	

Introduction:

A Hospital Management System (HMS) is an integrated software designed to streamline the administrative, medical, legal, and financial aspects of a healthcare institution. It facilitates the efficient handling of patient records, hospital workflows, and data management across multiple departments. The system ensures improved coordination, better healthcare delivery, and enhanced patient care by digitizing birth records and OPD records management. This digital approach not only minimizes paperwork but also provides secure and easily accessible patient data.

Application:

- The Hospital Management System can be applied to various areas within healthcare institutions, such as:
- Birth Record Management: Facilitating the recording of birth details, storing and retrieving birth certificates, and managing related legal documents.
- OPD Record Management: Handling outpatient visits, recording patient details, diagnoses, treatments, prescriptions, and follow-up appointments.
- Billing and Finance: Managing billing, invoicing, and insurance claims for both OPD and in-patient services.
- Inventory Management: Keeping track of medical supplies, drugs, and equipment.
- Laboratory and Diagnostics: Managing lab tests and diagnostic results for easy access and review by doctors.
- Staff and Doctor Scheduling: Organizing work shifts, doctor appointments, and staff allocation efficiently.

Advantages:

- Improved Efficiency: The system automates administrative tasks, reducing manual errors and saving time.
- Centralized Data Management: All patient data, from birth records to OPD visits, is stored centrally, making it easier to retrieve and manage.
- Enhanced Patient Care: Doctors and medical staff can access accurate patient history, improving the quality of treatment and decision-making.
- Cost Savings: Automation reduces paperwork, physical storage, and administrative overheads.
- Data Security: Implementing strong data encryption and access control ensures the confidentiality and security of patient records.

- **Faster Processing:** Efficient handling of records and scheduling reduces patient wait times and increases overall hospital throughput.

Disadvantages :

- **Initial Cost:** Implementing an HMS requires substantial investment in software, hardware, and training.
- **Technical Issues:** Systems might face downtimes or bugs that can disrupt hospital operations.
- **Data Breach Risks:** Without proper security measures, there's always a risk of sensitive patient information being compromised.
- **Complexity:** The system can be complex, and staff may require time to adjust to using it effectively.
- **Maintenance Costs:** Ongoing costs for system maintenance, updates, and support can be high.

Objectives :

To streamline birth record management by automating the process of registering births, issuing certificates, and ensuring compliance with legal standards.

To manage OPD records efficiently, allowing easy tracking of patient visits, diagnoses, and treatments.

To enhance the overall patient experience by reducing wait times and providing faster access to medical records.

To secure sensitive patient data by implementing role-based access controls and encryption mechanisms.

To provide analytical insights for hospital management regarding patient trends, diagnoses, and operational efficiency.

Problem Statement :

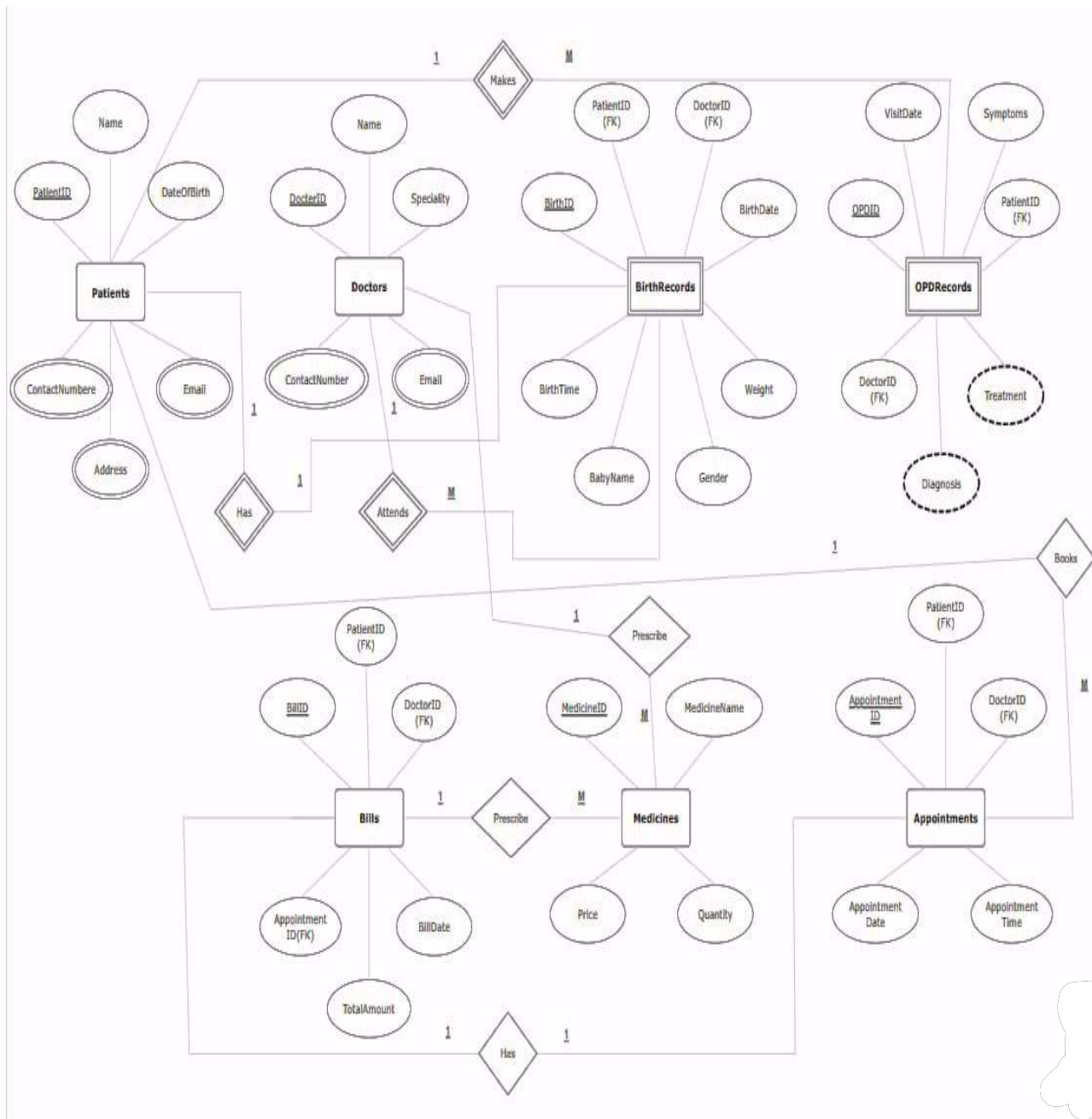
The current manual or outdated system for managing birth and OPD records in hospitals leads to data inaccuracies, inefficiency, and delays in patient care. Manual processes are prone to errors, making it difficult to ensure compliance with legal requirements like birth certificates.

Additionally, physical records are at risk of loss or damage, compromising data integrity.

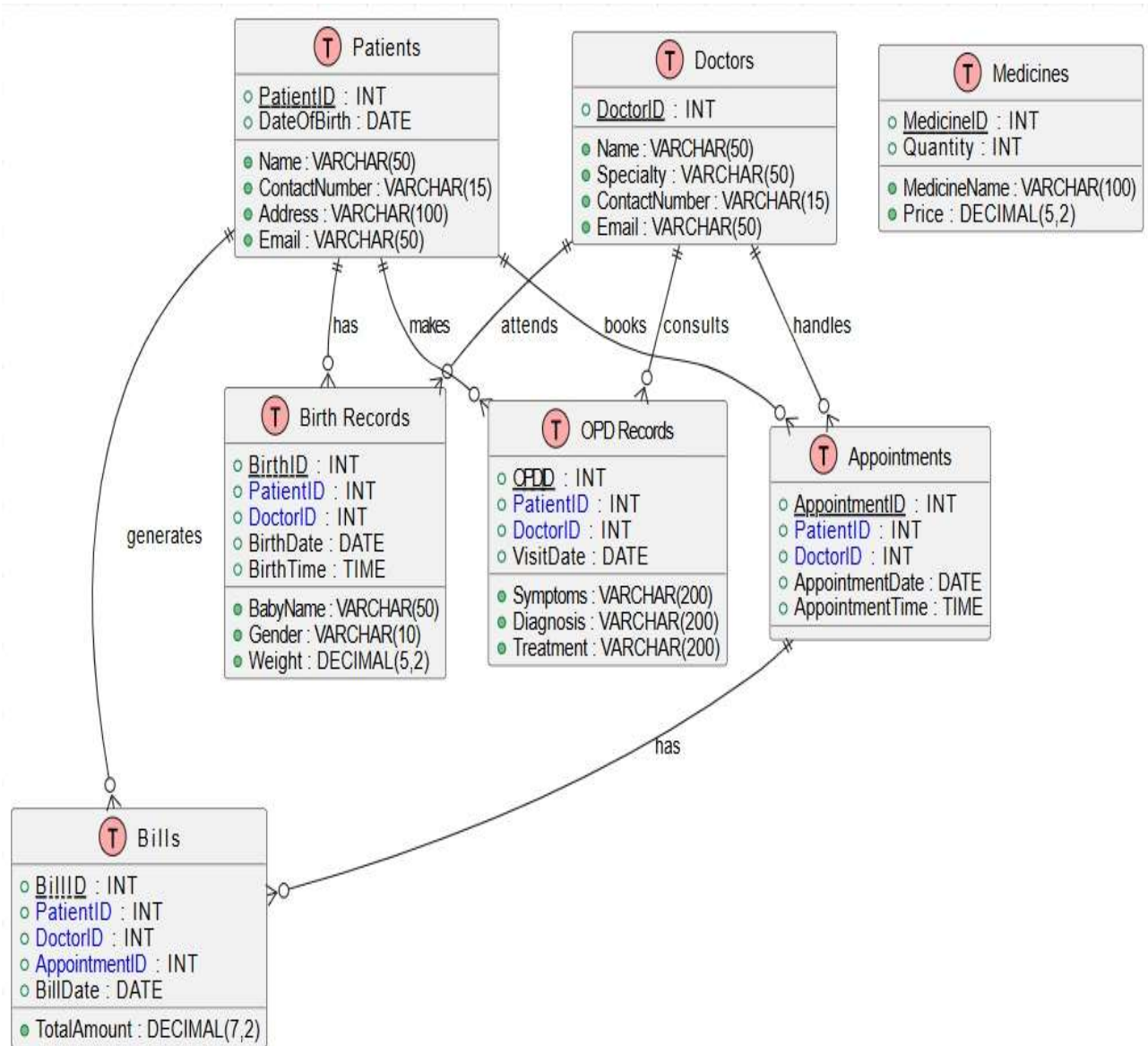
The lack of a centralized, digital system results in slow information retrieval and operational

inefficiencies. Ensuring the security and privacy of sensitive patient data is also a challenge in these traditional systems. The Hospital Management System aims to address these issues by providing a digital solution that offers accurate, efficient, and secure management of hospital records and workflows.

ER DIAGRAM :



SCHEMA DIAGRAM:



DATABASE TABLES :

1. Patient Table

```
hospital=# select * from patients;
 patientid |  name  | dateofbirth | contactnumber | address | email
-----+-----+-----+-----+-----+-----
          1 | John Doe | 1990-01-01 | 1234567890 | 123 Main St | john.doe@example.com
          2 | Jane Doe | 1995-06-01 | 9876543210 | 456 Elm St | jane.doe@example.com
          3 | Smita | 1996-08-10 | 5279317857 | 432 Main St | smita.23@example.com
          4 | Sam | 1996-09-12 | 7824262474 | 879 Elm St | sam2348@example.com
(4 rows)
```

2. Doctor Table

```
hospital=# select * from doctors;
 doctorid |  name  | specialty | contactnumber | email
-----+-----+-----+-----+-----
          1 | Dr. Smith | Obstetrics | 1234567890 | dr.smith@example.com
          2 | Dr. Johnson | Pediatrics | 9876543210 | dr.johnson@example.com
          3 | Dr. Sham | Gynecologist | 4278658942 | dr.sham@example.com
          4 | Dr. Rishi | Gynecologist | 4278428932 | dr.rishi@example.com
(4 rows)
```

3. Birth Record Table

```
hospital=# select * from birthrecords;
 birthid | patientid | doctorid | birthdate | birthtime | babyname | gender | weight
-----+-----+-----+-----+-----+-----+-----+-----
          1 |          1 |          1 | 2022-01-01 | 10:00:00 | Baby Doe | Male | 3.50
          2 |          2 |          2 | 2022-06-01 | 11:00:00 | Baby Jane | Female | 3.20
          3 |          3 |          3 | 2023-11-10 | 10:00:00 | Krish | Male | 3.30
          4 |          4 |          4 | 2024-01-01 | 12:00:00 | Sanika | Female | 3.50
(4 rows)
```

4. OPD Records Table

```
hospital=# select * from opdrecords;
 opdid | patientid | doctorid | visitdate | symptoms | diagnosis | treatment
-----+-----+-----+-----+-----+-----+-----
          1 |          1 |          1 | 2022-01-15 | Fever | Influenza | Antibiotics
          2 |          2 |          2 | 2022-06-15 | Cough | Bronchitis | Cough Syrup
          3 |          3 |          3 | 2023-04-12 | Fever | Influenza | Antibiotics
          4 |          4 |          4 | 2024-09-15 | Cough | Bronchitis | Cough Syrup
(4 rows)
```


5. Medicines Table

```
hospital=# select * from medicines;
 medicineid | medicinename | quantity | price
-----+-----+-----+-----
          1 | Aspirin      |        100 |    5.00
          2 | Ibuprofen    |         50 |    3.00
          3 | Paracetamol  |        200 |    2.50
          4 | Amoxicillin  |        150 |    8.00
          5 | Cough Syrup  |         80 |    6.50
(5 rows)
```

6. Appointment Table

```
hospital=# select * from appointments;
 appointmentid | patientid | doctorid | appointmentdate | appointmenttime
-----+-----+-----+-----+-----
          1 |      1 |      1 | 2023-03-01 | 10:00:00
          2 |      2 |      2 | 2023-03-02 | 11:00:00
          3 |      3 |      3 | 2023-03-03 | 12:00:00
          4 |      4 |      4 | 2023-03-04 | 09:00:00
          5 |      5 |      5 | 2023-03-05 | 14:00:00
(5 rows)
```

7. Bills Table

```
hospital=# select * from bills;
 billid | patientid | doctorid | appointmentid | billdate | totalamount
-----+-----+-----+-----+-----+-----
      1 |      1 |      1 |          1 | 2023-03-01 |    100.00
      2 |      2 |      2 |          2 | 2023-03-02 |    150.00
      3 |      3 |      3 |          3 | 2023-03-03 |    120.00
      4 |      4 |      4 |          4 | 2023-03-04 |    130.00
(4 rows)
```

QUERIES PERFORMED :

Basic Commands (CREATE, INSERT, SELECT , DROP):

1. Create table Birth Records.

```
hospital=# CREATE TABLE BirthRecords (  
hospital(# BirthID INT PRIMARY KEY,  
hospital(# PatientID INT,  
hospital(# DoctorID INT,  
hospital(# BirthDate DATE,  
hospital(# BirthTime TIME,  
hospital(# BabyName VARCHAR(50),  
hospital(# Gender VARCHAR(10),  
hospital(# Weight DECIMAL(5,2),  
hospital(# FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),  
hospital(# FOREIGN KEY (DoctorID) REFERENCES Doctors(DoctorID)  
hospital(# );  
CREATE TABLE
```

2. Insert into Birth record table.

```
hospital=# INSERT INTO BirthRecords (BirthID, PatientID, DoctorID, BirthDate, BirthTime, BabyName, Gender, Weight)  
hospital-# VALUES  
hospital-# (1, 1, 1, '2022-01-01', '10:00:00', 'Baby Doe', 'Male', 3.5),  
hospital-# (2, 2, 2, '2022-06-01', '11:00:00', 'Baby Jane', 'Female', 3.2),  
hospital-# (3, 3, 3, '2023-11-10', '10:00:00', 'Krish', 'Male', 3.3),  
hospital-# (4, 4, 4, '2024-01-01', '12:00:00', 'Sanika', 'Female', 3.5);  
INSERT 0 4  
hospital=# select * from birthrecords;  
 birthid | patientid | doctorid | birthdate | birthtime | babyname | gender | weight  
-----+-----+-----+-----+-----+-----+-----+-----  
1 | 1 | 1 | 2022-01-01 | 10:00:00 | Baby Doe | Male | 3.50  
2 | 2 | 2 | 2022-06-01 | 11:00:00 | Baby Jane | Female | 3.20  
3 | 3 | 3 | 2023-11-10 | 10:00:00 | Krish | Male | 3.30  
4 | 4 | 4 | 2024-01-01 | 12:00:00 | Sanika | Female | 3.50  
(4 rows)
```

3. Select from Appointment table where Doctor id is 2.

```
hospital=# SELECT * FROM Appointments WHERE DoctorID = 2;  
 appointmentid | patientid | doctorid | appointmentdate | appointmenttime  
-----+-----+-----+-----+-----  
2 | 2 | 2 | 2023-03-02 | 11:00:00  
(1 row)
```

4. Drop the column baby name from Birth records table.

```
hospital=# ALTER TABLE BirthRecords  
hospital-# DROP COLUMN BabyName;  
ALTER TABLE  
hospital=# select * from birthrecords;  
 birthid | patientid | doctorid | birthdate | birthtime | gender | weight  
-----+-----+-----+-----+-----+-----+-----  
1 | 1 | 1 | 2022-01-01 | 10:00:00 | Male | 3.50  
2 | 2 | 2 | 2022-06-01 | 11:00:00 | Female | 3.20  
3 | 3 | 3 | 2023-11-10 | 10:00:00 | Male | 3.30  
4 | 4 | 4 | 2024-01-01 | 12:00:00 | Female | 3.50  
(4 rows)
```

Modifying Commands (DELETE , UPDATE, ALTER , RENAME)

1. Delete from OPD records table where the patient has been diagnosed with Influenza.

```
hospital=# DELETE FROM OPDRecords
hospital=# WHERE Diagnosis = 'Influenza';
DELETE 2
hospital=# select * from opdrecords;
```

opdid	patientid	doctorid	visitdate	symptoms	diagnosis	treatment
2	2	2	2022-06-15	Cough	Bronchitis	Cough Syrup
4	4	4	2024-09-15	Cough	Bronchitis	Cough Syrup

(2 rows)

2. Update the Patients Contact number whose Patient id is 1.

```
hospital=# UPDATE Patients
hospital=# SET ContactNumber = '1122334455'
hospital=# WHERE PatientID = 1;
UPDATE 1
hospital=# select * from patients;
```

patientid	name	dateofbirth	contactnumber	address	email
2	Jane Doe	1995-06-01	9876543210	456 Elm St	jane.doe@example.com
3	Smita	1996-08-10	5279317857	432 Main St	smita.23@example.com
4	Sam	1996-09-12	7824262474	879 Elm St	sam2348@example.com
1	John Doe	1990-01-01	1122334455	123 Main St	john.doe@example.com

(4 rows)

3. Alter Patient table to add a column of bloodgroup.

```
hospital=# ALTER TABLE Patients
hospital=# ADD BloodGroup VARCHAR(5);
ALTER TABLE
hospital=# select * from patients;
```

patientid	name	dateofbirth	contactnumber	address	email	bloodgroup
2	Jane Doe	1995-06-01	9876543210	456 Elm St	jane.doe@example.com	
3	Smita	1996-08-10	5279317857	432 Main St	smita.23@example.com	
4	Sam	1996-09-12	7824262474	879 Elm St	sam2348@example.com	
1	John Doe	1990-01-01	1122334455	123 Main St	john.doe@example.com	

(4 rows)

4. Rename the column name in doctor table to doctor name.

```
hospital=# ALTER TABLE Doctors
hospital=# RENAME COLUMN Name TO DoctorName;
ALTER TABLE
hospital=# select * from doctors;
```

doctorid	doctorname	specialty	contactnumber	email
1	Dr. Smith	Obstetrics	1234567890	dr.smith@example.com
2	Dr. Johnson	Pediatrics	9876543210	dr.johnson@example.com
3	Dr. Sham	Gynecologist	4278658942	dr.sham@example.com
4	Dr. Rishi	Gynecologist	4278428932	dr.rishi@example.com

(4 rows)

Compound Conditions (BETWEEN , AND , OR, LIKE, NOT LIKE)

1. Find all the patients whose Date of Birth is between 1990 and 1995.

```
hospital=# SELECT * FROM Patients
hospital=# WHERE DateOfBirth BETWEEN '1990-01-01' AND '1995-12-31';
 patientid | name   | dateofbirth | contactnumber | address   | email               | bloodgroup
-----+-----+-----+-----+-----+-----+-----
          2 | Jane Doe | 1995-06-01 | 9876543210 | 456 Elm St | jane.doe@example.com | 
          1 | John Doe | 1990-01-01 | 1122334455 | 123 Main St | john.doe@example.com | 
(2 rows)
```

2. Find the patient id whose weight is greater than 3.0 AND gender is male.

```
hospital=# SELECT * FROM BirthRecords
hospital=# WHERE Weight > 3.0 AND Gender = 'Male';
 birthid | patientid | doctorid | birthdate | birthtime | babyname | gender | weight
-----+-----+-----+-----+-----+-----+-----+-----
          1 |          1 |          1 | 2022-01-01 | 10:00:00 | Baby Doe | Male | 3.50
          3 |          3 |          3 | 2023-11-10 | 10:00:00 | Krish   | Male | 3.30
(2 rows)
```

3. Find all the patients who live either on Main St OR Elm St.

```
hospital=# SELECT * FROM Patients
hospital=# WHERE Address LIKE '%Main St%' OR Address LIKE '%Elm St%';
 patientid | name   | dateofbirth | contactnumber | address   | email               | bloodgroup
-----+-----+-----+-----+-----+-----+-----
          2 | Jane Doe | 1995-06-01 | 9876543210 | 456 Elm St | jane.doe@example.com | 
          3 | Smita   | 1996-08-10 | 5279317857 | 432 Main St | smita.23@example.com | 
          4 | Sam     | 1996-09-12 | 7824262474 | 879 Elm St | sam2348@example.com | 
          1 | John Doe | 1990-01-01 | 1122334455 | 123 Main St | john.doe@example.com | 
(4 rows)
```

4. Find all patients whose name starts with J.

```
hospital=# SELECT * FROM Patients
hospital=# WHERE Name LIKE 'J%';
 patientid | name   | dateofbirth | contactnumber | address   | email               | bloodgroup
-----+-----+-----+-----+-----+-----+-----
          2 | Jane Doe | 1995-06-01 | 9876543210 | 456 Elm St | jane.doe@example.com | 
          1 | John Doe | 1990-01-01 | 1122334455 | 123 Main St | john.doe@example.com | 
(2 rows)
```

5. Find all doctors whose speciality does not contain Pediatrics.

```
hospital=# SELECT * FROM Doctors
hospital=# WHERE Specialty NOT LIKE '%Pediatrics%';
 doctorid | name   | specialty   | contactnumber | email
-----+-----+-----+-----+-----
          1 | Dr. Smith | Obstetrics | 1234567890 | dr.smith@example.com
          3 | Dr. Sham  | Gynecologist | 4278658942 | dr.sham@example.com
          4 | Dr. Rishi | Gynecologist | 4278428932 | dr.rishi@example.com
(3 rows)
```

Aggregate Functions (SUM, AVERAGE, MAX ,MIN, COUNT)

1. Calculate the total billing amount for all patients.

```
hospital=# SELECT SUM(TotalAmount) AS TotalBillsAmount
hospital=# FROM Bills;
 totalbillsamount
-----
          500.00
(1 row)
```

2. Find the average of total amount from the bills amount table.

```
hospital=# SELECT AVG(TotalAmount) AS BillsAmount
hospital=# FROM Bills;
      billsamount
-----
 125.0000000000000000
(1 row)
```

3. Find the maximum weight in the Birth records table

```
hospital=# SELECT MAX(Weight) AS MaxBirthWeight
hospital=# FROM BirthRecords;
 maxbirthweight
-----
          3.50
(1 row)
```

4. Find the minimum billing amount in the billing table.

```
hospital=# SELECT MIN(TotalAmount) AS BillsAmount
hospital=# FROM Bills;
 billsamount
-----
      100.00
(1 row)
```

5. Count the total number of patients from the Patient table.

```
hospital=# SELECT COUNT(*) AS TotalPatients
hospital=# FROM Patients;
totalpatients
-----
4
(1 row)
```

GROUP BY & ORDER BY :

1. List all patients ordered by their name in ascending order from Patient table.

```
hospital=# SELECT * FROM Patients
hospital=# ORDER BY Name ASC;
 patientid | name   | dateofbirth | contactnumber | address | email | bloodgroup
-----+-----+-----+-----+-----+-----+-----
          2 | Jane Doe | 1995-06-01 | 9876543210 | 456 Elm St | jane.doe@example.com | 
          1 | John Doe | 1990-01-01 | 1122334455 | 123 Main St | john.doe@example.com | 
          4 | Sam     | 1996-09-12 | 7824262474 | 879 Elm St | sam2348@example.com | 
          3 | Smita   | 1996-08-10 | 5279317857 | 432 Main St | smita.23@example.com | 
(4 rows)
```

2. Sort the total amount from bill table in descending order.

```
hospital=# SELECT * FROM Bills
hospital=# ORDER BY TotalAmount DESC;
 billid | patientid | doctorid | appointmentid | billdate | totalamount
-----+-----+-----+-----+-----+-----
          2 |          2 |          2 |          2 | 2023-03-02 | 150.00
          4 |          4 |          4 |          4 | 2023-03-04 | 130.00
          3 |          3 |          3 |          3 | 2023-03-03 | 120.00
          1 |          1 |          1 |          1 | 2023-03-01 | 100.00
(4 rows)
```

3. Group the birth records by gender and calculate the average birth weight for each gender.

```
hospital=# SELECT Gender, AVG(Weight) AS AvgBirthWeight
hospital=# FROM BirthRecords
hospital=# GROUP BY Gender;
 gender | avgbirthweight
-----+-----
 Female | 3.3500000000000000
 Male   | 3.4000000000000000
(2 rows)
```

UNION , INTERSECT & MINUS:

1. Combine records from two tables, e.g , retrieve names of both patients and doctors.

```
hospital=# SELECT Name FROM Patients
hospital=# UNION
hospital=# SELECT Name FROM Doctors;
name
-----
Dr. Rishi
Smita
Sam
Dr. Smith
Dr. Sham
John Doe
Jane Doe
Dr. Johnson
(8 rows)
```

2. Find patients and doctors who have the same name.

```
hospital=# SELECT Name FROM Patients
hospital=# INTERSECT
hospital=# SELECT Name FROM Doctors;
name
-----
(0 rows)
```

3. Find patients whose names are not listed as doctors.

```
hospital=# SELECT Name FROM Patients
hospital=# EXCEPT
hospital=# SELECT doctorname FROM Doctors;
name
-----
John Doe
Sam
Smita
Jane Doe
(4 rows)
```

JOINS (INNER , LEFT , RIGHT , FULL OUTER) :

1. Retrieve patient names and birth dates for those who have a matching birth record .

```
hospital=# SELECT Patients.Name, BirthRecords.BirthDate FROM Patients
hospital=# INNER JOIN BirthRecords
hospital=# ON Patients.PatientID =BirthRecords.PatientID;
  name | birthdate
-----+-----
 John Doe | 2022-01-01
 Jane Doe | 2022-06-01
  Smita   | 2023-11-10
   Sam    | 2024-01-01
(4 rows)
```

2. Retrieve all patient names and their birth dates, including patients without birth records.

```
hospital=# SELECT Patients.Name, BirthRecords.BirthDate FROM Patients
hospital=# LEFT JOIN BirthRecords
hospital=# ON Patients.PatientID = BirthRecords.PatientID;
  name | birthdate
-----+-----
 John Doe | 2022-01-01
 Jane Doe | 2022-06-01
  Smita   | 2023-11-10
   Sam    | 2024-01-01
(4 rows)
```

3. Retrieve all birth records and their corresponding patient names, including birth records without matching patients.

```
hospital=# SELECT Patients.Name, BirthRecords.BirthDate FROM Patients
hospital=# RIGHT JOIN BirthRecords
hospital=# ON Patients.PatientID = BirthRecords.PatientID;
  name | birthdate
-----+-----
 John Doe | 2022-01-01
 Jane Doe | 2022-06-01
  Smita   | 2023-11-10
   Sam    | 2024-01-01
(4 rows)
```

4. Retrieve all patient names and birth dates, including those without matches in either table.

```
hospital=# SELECT Patients.Name, BirthRecords.BirthDate FROM Patients
hospital=# FULL OUTER JOIN BirthRecords
hospital=# ON Patients.PatientID = BirthRecords.PatientID;
  name | birthdate
-----+-----
 John Doe | 2022-01-01
 Jane Doe | 2022-06-01
  Smita   | 2023-11-10
   Sam    | 2024-01-01
(4 rows)
```


VIEWS :

1. Create a view that shows patient names, their birth details .

```
hospital=# CREATE VIEW PatientBirthInfo AS
hospital=# SELECT Patients.Name, BirthRecords.BirthDate, BirthRecords.Gender, BirthRecords.Weight
hospital=# FROM Patients
hospital=# JOIN BirthRecords ON Patients.PatientID = BirthRecords.PatientID;
CREATE VIEW
hospital=# SELECT * FROM PatientBirthInfo;
  name | birthdate | gender | weight
-----+-----+-----+-----
John Doe | 2022-01-01 | Male | 3.50
Jane Doe | 2022-06-01 | Female | 3.20
Smita | 2023-11-10 | Male | 3.30
Sam | 2024-01-01 | Female | 3.50
```

INDEXEX :

1. Create an index on the Patients table for faster searching by name.

```
hospital=# CREATE INDEX idx_patient_name ON Patients(Name);
CREATE INDEX
hospital=# SELECT indexname, indexdef
hospital=# FROM pg_indexes
hospital=# WHERE tablename = 'patients';
  indexname | indexdef
-----+-----
patients_pkey | CREATE UNIQUE INDEX patients_pkey ON public.patients USING btree (patientid)
idx_patient_name | CREATE INDEX idx_patient_name ON public.patients USING btree (name)
(2 rows)
```

CONCLUSION:

In conclusion, implementing a Hospital Management System (HMS) addresses the significant challenges posed by outdated manual record-keeping methods. By digitizing birth and OPD records, the HMS enhances data accuracy, efficiency, and security, ultimately improving patient care. This integrated solution streamlines workflows across departments, ensuring better coordination and compliance with legal requirements. Additionally, it protects sensitive patient data while minimizing the risk of loss or damage. Overall, the HMS is essential for modernizing healthcare administration and enhancing the quality of care provided to patients.

REFERENCES:

<https://www.scribd.com/document/349571448/Hospital-Management-System>

<https://www.slideshare.net/slideshow/hospital-management-system-project/34616076>

https://www.researchgate.net/publication/380428854_HOSPITAL_MANAGEMENT_SYSTEM_PROJECT

https://mscw.ac.in/NAAC/Criteria1/Samples-of-ProjectWork_Fieldwork/Computer_Science/software_Engineering/Software%20Engineering/Hospital%20Managment%20Project%20SE-converted.pdf

<https://srmus.ac.in/view/governance/naac/1.3.4/Supporting%20Documents/MCA/ANISHA%20SHARMA%20%5B20IT103022%5D.pdf>