BUAN 6320

DATABASE FOUNDATIONS
FOR
BUSINESS ANALYTICS

PROJECT REPORT
HOSPITAL MANAGEMENT SYSTEM

GROUP 5
MEMBERS:

ANAGHA SHEVADE
VISHVESH PANDEY
AAYUSH SHUKLA
TAPASYA CHITALDURG
PRAGATI DIVEKAR

TABLE OF CONTENTS

1. Introduction

- We have chosen the topic of hospital management systems because healthcare is a field that is expanding at a swift pace.
- During the global pandemic around this world, we all realised how important healthcare is to everyone, and so is the healthcare management system.
- To make the healthcare sector more efficient and better in terms of technology, we have proposed creating our version of the hospital management database system.
- Hospital Management systems help to keep track of patients, staff, and doctors and even with the help of advanced querying logic that we have come up with, it makes it so much fun along with providing some interesting knowledge from the raw data.

2. Contribution

Name	Contribution
Vishvesh Pandey	Table-creation,DataInsertion,Stored Procedures,DataCompilation,Trigger, Presentation,Report
Aayush Shukla	Table creation, Data Insertion , Stored Procedures , Presentation , Report .
Tapasya Chitaldurg	Table-creation,DataInsertion,Stored function, Presentation
Pragati Divekar	Table creation,Data Insertion,Joins , ,Trigger,Views, Stored functions,Presentation,Report
Anagha Shevade	Table creation,Data Insertion,Join Queries , , Trigger, Stored functions, Stored Procedure, Data Compilation,

3. Objective

- Our present modern information system makes use of computers for the execution, each of them connected through an optimised network.
- Healthcare is the most critical aspect of our society, and many healthcare providers face challenges in offering practical and active services to patients.
- Considering a multi-speciality hospital, many people enter and exit the hospital in a day and maintaining their records safely can result in a tedious process.
- To reduce this type of burden and to manage the financial, hospital administration and clinical aspects, the Hospital management system came into existence.
- This project aims to build a system to keep track of all data related to patients, employees (Ex: Nurses, Ward boys, Cleaners, Clerks etc.), medicines, and so much more.
- The database system will help the hospital to monitor all transactions (physical and/or digital) and help them in making informed decisions that are aligned with the company's vision.

4. Scope Definition

- This database management system will be used to manage and maintain the records in a hospital.
- We aim at delivering a fully working Database Management where the data can be stored efficiently and also evaluate insightful decisions with the help of triggers, stored procedures and functions.
- We have aimed at dividing our project into 3 modules-
- Patient, Employee and Stock Management systems, respectively.
- Each module consists of corresponding tables accordingly.
- We aim to make the lives of patients as well as the hospital staff easier with the help of our database management system.

5. Business Overview

A hospital management system is a computer system that helps manage the information related to health care and aids in the job completion of healthcare providers effectively.

HMS came into the picture of hospital management as early as 1960 and had ever since been evolving and synchronising with the technologies while modernising healthcare facilities. In today's world, the management of healthcare starts from the hands of the patients through their mobile phones and facilitates the patient's needs.

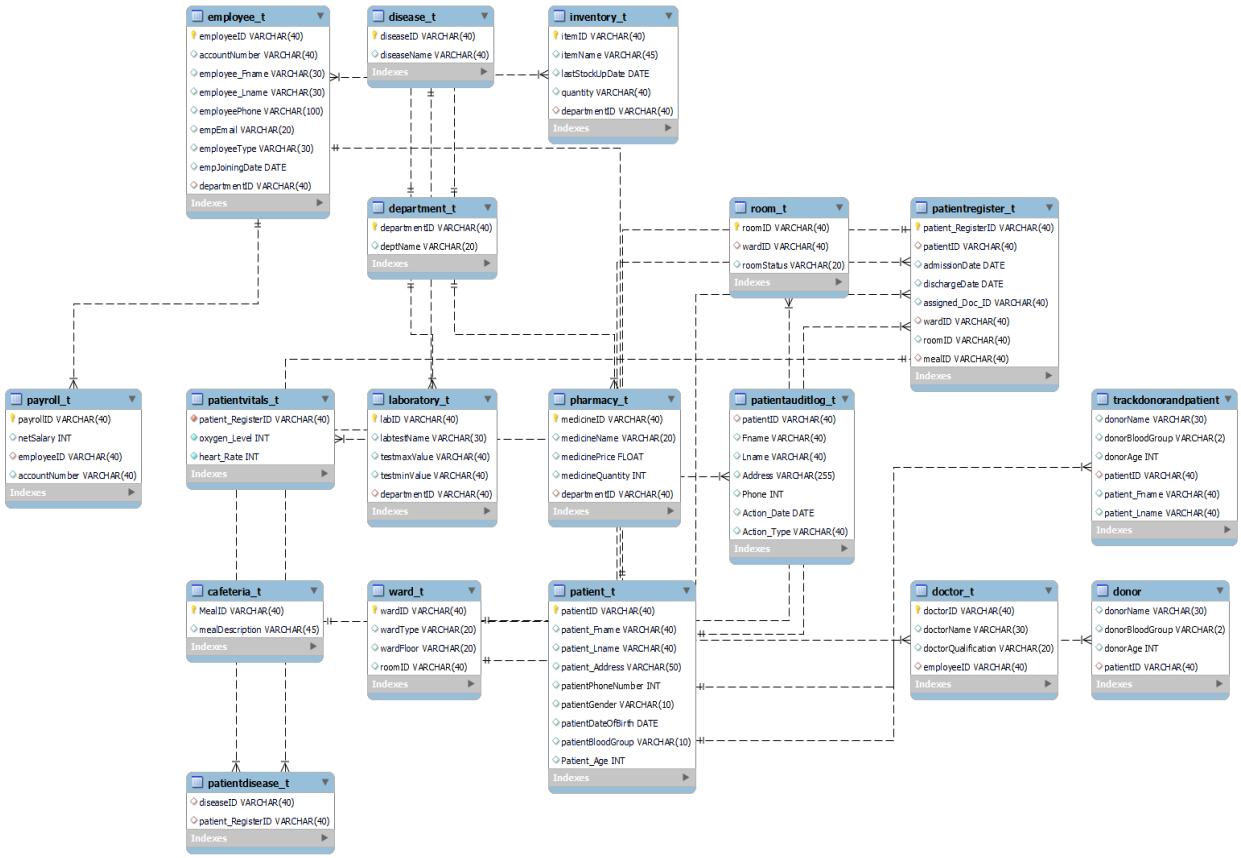
HMS was introduced to solve the complications coming from managing all the paperwork of every patient associated with the various departments of hospitalisation with confidentiality. HMS provides the ability to manage all the paperwork in one place, reducing the work of staff in arranging and analysing the patients' paperwork. HMS can help in many areas like

- ➔ Maintain the medical records of the patient
- ➔ Maintain the contact details of the patient
- ➔ Keeping track of the appointment dates
- ➔ Save the insurance information for later reference
- ➔ Tracking the bill payments.
- ➔ Maintain a proper flow of all the details involved with an HMS.

6. Implementation & Evaluation

Week 1	Team Meeting and Ideation
Week 2	Topic Finalisation
Week 3	Implementation and dividing tasks
Week 4	Creating Tables
Week 5	Insertion of Data
Week 6	Finalising Stored Procedures , Triggers , Views and Functions .
Week 7	Queries
Week 8	Implementation of Stored procedures and Funtions
Week 9	Implementation of Triggers and Views
Week 10	Code Testing
Week 11	Final Presentation and Report

7. ER Diagram


employee_t
employeeID VARCHAR(40)
accountNumber VARCHAR(40)
employee_Fname VARCHAR(30)
employee_Lname VARCHAR(30)
employeePhone VARCHAR(100)
empEmail VARCHAR(20)
employeeType VARCHAR(30)
empJoiningDate DATE
departmentID VARCHAR(40)
Indexes
disease_t
diseaseID VARCHAR(40)
diseaseName VARCHAR(40)
Indexes
inventory_t
itemID VARCHAR(40)
itemName VARCHAR(45)
lastStockUpDate DATE
quantity VARCHAR(40)
departmentID VARCHAR(40)
Indexes
department_t
departmentID VARCHAR(40)
deptName VARCHAR(20)
Indexes
room_t
roomID VARCHAR(40)
wardID VARCHAR(40)
roomStatus VARCHAR(20)
Indexes
patientregister_t
patient_RegisterID VARCHAR(40)
patientID VARCHAR(40)
admissionDate DATE
dischargeDate DATE
assigned_Doc_ID VARCHAR(40)
wardID VARCHAR(40)
roomID VARCHAR(40)
mealID VARCHAR(40)
Indexes
payroll_t
payrollID VARCHAR(40)
netSalary INT
employeeID VARCHAR(40)
accountNumber VARCHAR(40)
Indexes
patientvitals_t
patient_RegisterID VARCHAR(40)
oxygen_Level INT
heart_Rate INT
Indexes
laboratory_t
labID VARCHAR(40)
labtestName VARCHAR(30)
testmaxValue VARCHAR(40)
testminValue VARCHAR(40)
departmentID VARCHAR(40)
Indexes
pharmacy_t
medicineID VARCHAR(40)
medicineName VARCHAR(20)
medicinePrice FLOAT
medicineQuantity INT
departmentID VARCHAR(40)
Indexes
patientauditlog_t
patientID VARCHAR(40)
Fname VARCHAR(40)
Lname VARCHAR(40)
Address VARCHAR(255)
Phone INT
Action_Date DATE
Action_Type VARCHAR(40)
Indexes
trackdonorandpatient
donorName VARCHAR(30)
donorBloodGroup VARCHAR(2)
donorAge INT
patientID VARCHAR(40)
patient_Fname VARCHAR(40)
patient_Lname VARCHAR(40)
Indexes
cafeteria_t
MealID VARCHAR(40)
mealDescription VARCHAR(45)
Indexes
ward_t
wardID VARCHAR(40)
wardType VARCHAR(20)
wardFloor VARCHAR(20)
roomID VARCHAR(40)
Indexes
patient_t
patientID VARCHAR(40)
patient_Fname VARCHAR(40)
patient_Lname VARCHAR(40)
patient_Address VARCHAR(50)
patientPhoneNumber INT
patientGender VARCHAR(10)
patientDateOfBirth DATE
patientBloodGroup VARCHAR(10)
Patient_Age INT
Indexes
doctor_t
doctorID VARCHAR(40)
doctorName VARCHAR(30)
doctorQualification VARCHAR(20)
employeeID VARCHAR(40)
Indexes
donor
donorName VARCHAR(30)
donorBloodGroup VARCHAR(2)
donorAge INT
patientID VARCHAR(40)
Indexes
patientdisease_t
diseaseID VARCHAR(40)
patient_RegisterID VARCHAR(40)
Indexes

8. Code and output Screenshots

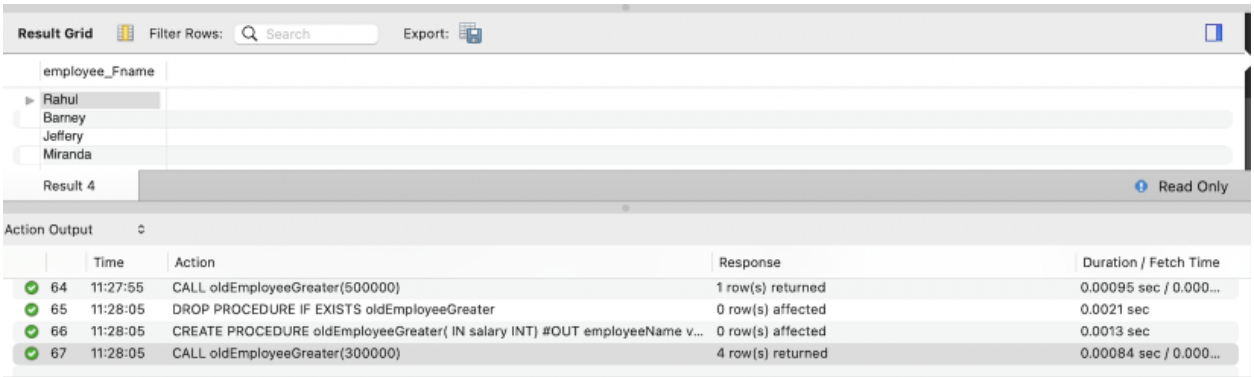
- **Stored Procedures**

#1 To find the diseases whose occurrence rate is high.

```
/*SP1. To find the disease whos occurance rate is high */
DROP PROCEDURE IF EXISTS maxDiseaseOcccurence;
DELIMITER $$
CREATE PROCEDURE maxDiseaseOcccurence(
OUT result VARCHAR(40)
)
BEGIN
DECLARE cumulativeResult DEC(10,2) DEFAULT 0;
SELECT patient_RegisterID INTO cumulativeResult from patientdisease_t where diseaseID IN (select max(count(diseaseID)) from patientdisease_t);
END$$
DELIMITER ;
CALL maxDiseaseOcccurence(@cumulativeResult);
select @cumulativeResult;
```

#2 Finding the name of the employees whose salary is greater than a specified range.

```
/*SP2.Find the name of the employees whos salary is greater than specified range*/
DROP PROCEDURE IF EXISTS oldEmployeeGreater;
DELIMITER $$
CREATE PROCEDURE oldEmployeeGreater(
IN salary INT)
#OUT employeeName varchar(30))
BEGIN
DECLARE ename varchar(30);
SELECT employee_Fname from employee_t where employeeID IN(select employeeID from payroll_t where netSalary > salary);
END$$
DELIMITER ;
CALL oldEmployeeGreater(300000);
```



#3 To give bonus to employees whose tenure is greater than 10 years

```
721  /*SP3.Write procedure to give bonus to employees whos tenure is greater than 10 years */
722  DROP PROCEDURE IF EXISTS bonusup;
723  DELIMITER $$
724  CREATE PROCEDURE bonusup(
725  IN bonusPercent float(20))
726  BEGIN
727  SET bonusPercent=bonusPercent+1;
728  SELECT employee_Fname,employee_Lname,DATE_FORMAT(FROM_DAYS(DATEDIFF(NOW(), empJoiningDate)), '%Y') + 0 AS tenure,netSalary*bonusPercent as bonus,netSalary as previousSalary
729  FROM employee_t E JOIN payroll_t P on  E.employeeID=P.employeeID
730  GROUP BY E.employeeID
731  HAVING tenure > '10';
732  END$$
733  DELIMITER ;
734  CALL bonusup(0.25);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

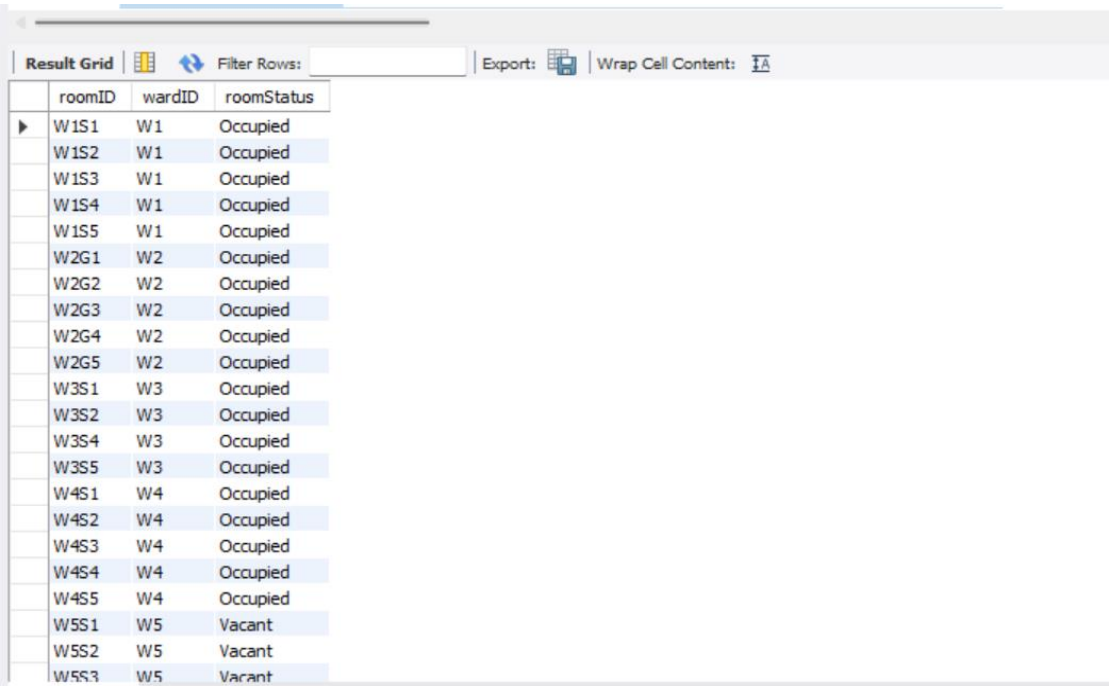
employee_Fname	employee_Lname	tenure	bonus	previousSalary
Anagha	Shevade	22	312500	250000
Monica	Geller	22	343750	275000
Joey	Tribbiani	11	62500	50000
Michael	Scott	17	343750	275000
Andy	Bernard	15	75000	60000

Result 39

#4 To check the current room status.

```
660  /*SP4.*/
661  DROP PROCEDURE IF EXISTS ROOM_STATUS;
662  DELIMITER $$
663  CREATE PROCEDURE ROOM_STATUS()
664  BEGIN
665  DECLARE id varchar(40);
666  SELECT roomID into id from patientregister_t where roomID IN(select RoomID from room_t where roomStatus='Vacant');
667  UPDATE room_t set roomStatus='Occupied' where roomID=id;
668  END$$
669  DELIMITER $$
670  call ROOM_STATUS();
671  INSERT INTO patientregister_t(patient_RegisterID,patientID, admissionDate, dischargeDate, assigned_Doc_ID, wardID ,roomID,mealID)
672  VALUES('P89','88','2022-11-20','2022-11-24','Doc1', 'W4', 'W5S1', 'Meal1');#Test
673  select* from room_t;
```

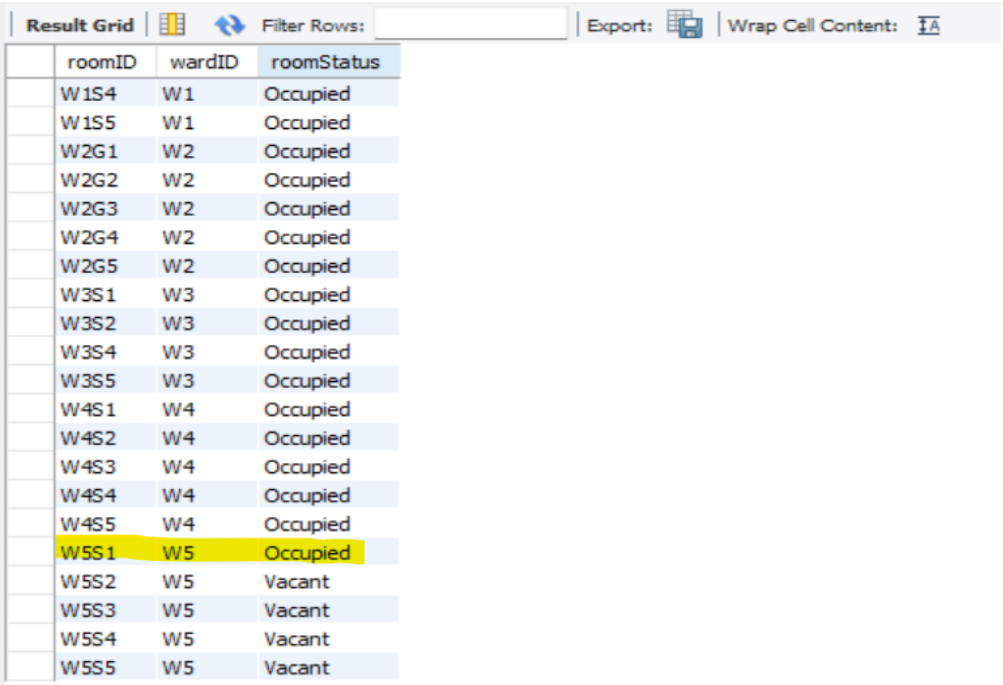
Before procedure



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

roomID	wardID	roomStatus
W1S1	W1	Occupied
W1S2	W1	Occupied
W1S3	W1	Occupied
W1S4	W1	Occupied
W1S5	W1	Occupied
W2G1	W2	Occupied
W2G2	W2	Occupied
W2G3	W2	Occupied
W2G4	W2	Occupied
W2G5	W2	Occupied
W3S1	W3	Occupied
W3S2	W3	Occupied
W3S4	W3	Occupied
W3S5	W3	Occupied
W4S1	W4	Occupied
W4S2	W4	Occupied
W4S3	W4	Occupied
W4S4	W4	Occupied
W4S5	W4	Occupied
W5S1	W5	Vacant
W5S2	W5	Vacant
W5S3	W5	Vacant

After Procedure



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

roomID	wardID	roomStatus
W1S4	W1	Occupied
W1S5	W1	Occupied
W2G1	W2	Occupied
W2G2	W2	Occupied
W2G3	W2	Occupied
W2G4	W2	Occupied
W2G5	W2	Occupied
W3S1	W3	Occupied
W3S2	W3	Occupied
W3S4	W3	Occupied
W3S5	W3	Occupied
W4S1	W4	Occupied
W4S2	W4	Occupied
W4S3	W4	Occupied
W4S4	W4	Occupied
W4S5	W4	Occupied
W5S1	W5	Occupied
W5S2	W5	Vacant
W5S3	W5	Vacant
W5S4	W5	Vacant
W5S5	W5	Vacant

- **Triggers**

#1 Trigger to Check if Medicine quantity is less than 1000.

```
800  /* Trigger1 to check if medicine quantity is less than 1000 */
801  Drop Trigger IF EXISTS Check_quantity;
802  Delimiter $$
803  CREATE TRIGGER  Check_quantity  AFTER INSERT ON Pharmacy_T
804  FOR EACH ROW
805  BEGIN
806  IF NEW.medicineQuantity < 1000 THEN
807  SIGNAL SQLSTATE '45000'
808  SET MESSAGE_TEXT = 'ERROR: refill Stock!';
809  END IF;
810  END; $$
811  select * from payroll_t;
812  INSERT INTO Pharmacy_T(
813      medicineID ,
814      medicineName ,
815      medicinePrice,
816      medicineQuantity) VALUES
817      ('M45','Paracetamol','40','900');
```

```
Error Code: 1644 ERROR: refill Stock!
```

#2 Trigger to check after insert on donor data patient record is updated in trackdonorpatient table.

```
     /*trigger 2.To check after insert on donor data patient record is updated in trackdonorpatient table*/
     DROP TRIGGER IF EXISTS archiveDonorPatient;
     DELIMITER $$
     CREATE TRIGGER archiveDonorPatient
822  AFTER INSERT ON donor
823  for each row
824  BEGIN
825
826  INSERT INTO trackDonorAndPatient(donorName,donorBloodGroup,donorAge,patientID,patient_Fname,patient_Lname)
827  select d.donorName,d.donorBloodGroup, d.donorAge,d.patientID,t.patient_Fname,t.patient_Lname from donor d join patient_t t ON t.patientID=d.patientID;
828  END$$
829  DELIMITER ;
830  Insert into donor(donorName,donorBloodGroup,donorAge,patientID)values
831  ("minini","A-",26,1);
832
833  select * from trackDonorAndPatient;
```

donorName	donorBloodGroup	donorAge	patientID	patient_Fname	patient_Lname
Sherlyn	A-	21	8	Anushka	Gupta
minini	A-	26	1	Jane	Anderson
Priyam	A+	20	10	Chirag	Rupani
ching yang	O+	25	1	Jane	Anderson
Pranay	A+	31	15	Amy	Santiago

trackDonorAndPatient 38 ×

#3 Trigger to Monitor Patient's Oxygen level. If the oxygen level goes below 80, it should send a trigger and call the doctor.

```
/*Trigger3.Monitor Patient's Oxygen level. If oxygen level goes below 80,it should send trigger and call the doctor*/
DROP TRIGGER IF EXISTS Monitor_Oxygen_Level;
DELIMITER $$
CREATE TRIGGER Monitor_Oxygen_Level
AFTER UPDATE
ON PatientVitals_T
FOR EACH ROW
BEGIN
DECLARE msg VARCHAR (128);
IF NEW.oxygen_Level < 80
THEN
SET msg = concat('Patient Oxygen Level Critical Call Doctor', ' ',
(SELECT assigned_Doc_ID FROM PatientRegister_T INNER JOIN PatientVitals_T
ON PatientRegister_T.patient_RegisterID = PatientVitals_T.patient_RegisterID
WHERE PatientVitals_T.oxygen_Level = NEW.oxygen_Level)) ;
SIGNAL SQLSTATE '45000' set message_text = msg;
END IF;
end
$$
SET
SQL_SAFE_UPDATES=0;
UPDATE PatientVitals_T
SET oxygen_Level = 40 WHERE
patient_RegisterID = "P4";
```

Error Code: 1644 Patient Oxygen Level Critical Call Doctor Doc3

#4 Trigger to check whether the patient's heart rate goes above 110 or below 50 then trigger will run and send the msg check on the patient.

```
/*Trigger4.If paient's heart rate goes above 110 or below 50 then trigger will run and send the msg check on the patient.*/
DROP TRIGGER IF EXISTS HeartRate_Critical;
    DELIMITER $$
    CREATE TRIGGER HeartRate_Critical
    AFTER UPDATE
    ON PatientVitals_T
    FOR EACH ROW
    BEGIN
    DECLARE msg2 VARCHAR (128);
IF NEW.heart_Rate > 110 OR NEW.heart_Rate < 50 THEN
    SET msg2 = CONCAT((SELECT Patient_T.patient_Fname FROM Patient_T
    INNER JOIN (SELECT PatientRegister_T.patient_RegisterID,PatientRegister_T.patientID FROM PatientRegister_T INNER JOIN
    PatientVitals_T
    ON PatientRegister_T.patient_RegisterID = PatientVitals_T.patient_RegisterID
    WHERE PatientVitals_T.heart_Rate = NEW.heart_Rate) AS temp ON temp.patientID = Patient_T.patientID),"'s"," ", "Heartbeat is Abnormal, Check Patient");
    SIGNAL SQLSTATE '45000' set message_text = msg2;
    END IF;
    END
    $$

SET SQL_SAFE_UPDATES=0;
UPDATE PatientVitals_T
SET heart_Rate = 120 WHERE
patient_RegisterID = "P1";
```

Error Code: 1644 Jane's Heartbeat is Abnormal, Check Patient

- **Views**

#1 Patient Ordinary Data

```
/*view1.Patient ordinary data*/
Create View Patient_Ordinary_Data
as
Select p.patientID,p.patient_Fname,p.patient_Lname,pr.admissionDate,pr.dischargeDate,pr.roomID from Patient_T as p join
PatientRegister_T as pr on p.patientID=pr.patientID;
select * from Patient_Ordinary_Data;
```

patientID	patient_Fname	patient_Lname	admissionDate	dischargeDate	roomID
1	Jane	Anderson	2022-11-20	2022-11-24	W4G5
10	Chirag	Rupani	2022-11-19	2022-11-13	W2G4
11	Shruthi	Pathak	2021-07-12	2021-08-12	W2G5
12	Raj	Arora	2022-11-23	2022-11-24	W3S1
13	Parth	Roy	2021-09-23	2021-09-29	W3S2
14	Nancy	Stellar	2000-11-30	2000-12-30	W3S3
15	Amy	Santiago	2022-10-08	2022-10-09	W3S4
16	Michelle	Parker	2018-09-21	2019-01-21	W3S5
17	Chritsitian	Pulisic	2021-11-24	2021-11-28	W4G1
18	David	Villa	2022-09-23	2022-10-23	W4G2

Patient_Ordinary_Data 34

#2 Patient Sensitive Data

```
/*View2.Patient sensitive data*/
Drop view if exists Patient_Sensitive_Data;
Create View Patient_Sensitive_Data
as
Select p.patientID,p.patient_Fname,p.patient_Lname,p.patient_Address,p.patientPhoneNumber,p.Patient_Age,dd.diseaseID,dd.diseaseName from Patient_T as p join
PatientRegister_T as pr on p.patientID=pr.patientID join PatientDisease_T as d on pr.patient_RegisterID=d.patient_RegisterID inner join Disease_T as dd on d.diseaseID=dd.diseaseI
select * from Patient_Sensitive_Data;
```

patientID	patient_Fname	patient_Lname	admissionDate	dischargeDate	roomID
1	Jane	Anderson	2022-11-20	2022-11-24	W4G5
10	Chirag	Rupani	2022-11-19	2022-11-13	W2G4
11	Shruthi	Pathak	2021-07-12	2021-08-12	W2G5
12	Raj	Arora	2022-11-23	2022-11-24	W3S1
13	Parth	Roy	2021-09-23	2021-09-29	W3S2
14	Nancy	Stellar	2000-11-30	2000-12-30	W3S3
15	Amy	Santiago	2022-10-08	2022-10-09	W3S4
16	Michelle	Parker	2018-09-21	2019-01-21	W3S5
17	Chritsitian	Pulisic	2021-11-24	2021-11-28	W4G1
18	David	Villa	2022-09-23	2022-10-23	W4G2

Patient_Ordinary_Data 34

#3 Employee Sensitive Data

```
768    /*view3.Employee sensitive data*/
769    Drop view if exists Employee_Sensitive_Data;
770    Create View Employee_Sensitive_Data
771    as
772    select e.employeeID,e.AccountNumber,concat(e.employee_Fname,e.employee_Lname) as Fullname,e.employeePhone,
773    e.empEmail,d.deptName,pay.netSalary from Employee_T as e join Department_T as d on e.departmentID=d.departmentID join Payroll_T as pay on e.employeeID=pay.employeeID;
774    select * from Employee_Sensitive_Data;
775
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

employeeID	AccountNumber	Fullname	employeePhone	empEmail	deptName
Emp1	Acc1	AnaghaShevade	9760981239	ana@gmail.com	Emergency
Emp10	Acc10	JohnDoe	4236780512	95john@gmail.com	Human Resc
Emp11	Acc11	MonicaGeller	1123409876	monica@gmail.com	Human Resc
Emp12	Acc12	JoeyTribbiani	8754093821	tribb@yahoo.com	Pathalogy
Emp13	Acc13	MichaelScott	3459867592	scott@rocketmail.com	Nursing
Emp14	Acc14	AndyBernard	3429945786	andy@gmail.com	Payroll
Emp15	Acc15	RahulRoy	2985579834	roy@gmail.com	Nursing
Emp16	Acc16	BarneyStinson	8964531889	barney540@yahoo.com	Ambulance
Emp17	Acc17	StephyMonroe	9238485912	stephy87@yahoo.com	Administrati

Employee_Sensitive_Data 36 ×

9. Future Scope

- Our project can be further enhanced to link the real-time ID such as SSN or Insurance details that can retrieve the patient's previous health records.
- We can use these records and store them in our archive so that we can write some meaningful Stored Procedures and Join operations to create some insightful decisions.
- An Ambulance management network can be linked with our project to track and improve the progress of-
- The ambulance reaching the patient + the ambulance bringing the patient to the hospital.
- Along with managing the drivers(staff), equipment in the ambulance (stocking) and also a basic doctor/nurse to provide emergency CPR to the patient.
- We can also create and integrate a feedback system for patients to lodge their complaints and listen to their suggested potential improvements.

10. Conclusion

★ Hence, here we have created database management for the effective management of the hospital.

★ Our Database contributes to the smooth workflow of operations by-

- ➢ Storing the patient's past and current data.
- ➢ Enrolling and assigning new patients.
- ➢ Maintaining and stocking the pharmacy as and when needed.
- ➢ Keeping a track of the patient's meals and room/ward.
- ➢ Ensuring the Doctor's data is stored separately and is linked to the patient assigned.
- ➢ Support the employee records via the department.
- ➢ Generate salaries and other finance using payroll definition.
- ➢ Maintain the pathology records with the help of patient vitals.
- ➢ Track dangerous and widespread diseases.
- ➢ Track the blood donor and patients.

11. References

https://ehealth4everyone.com/relevance-of-database-management-in-healthcare/

https://www.w3resource.com/sql-exercises/hospital-database-exercise/index.php