mediForms

Hospital Management System

Group 13

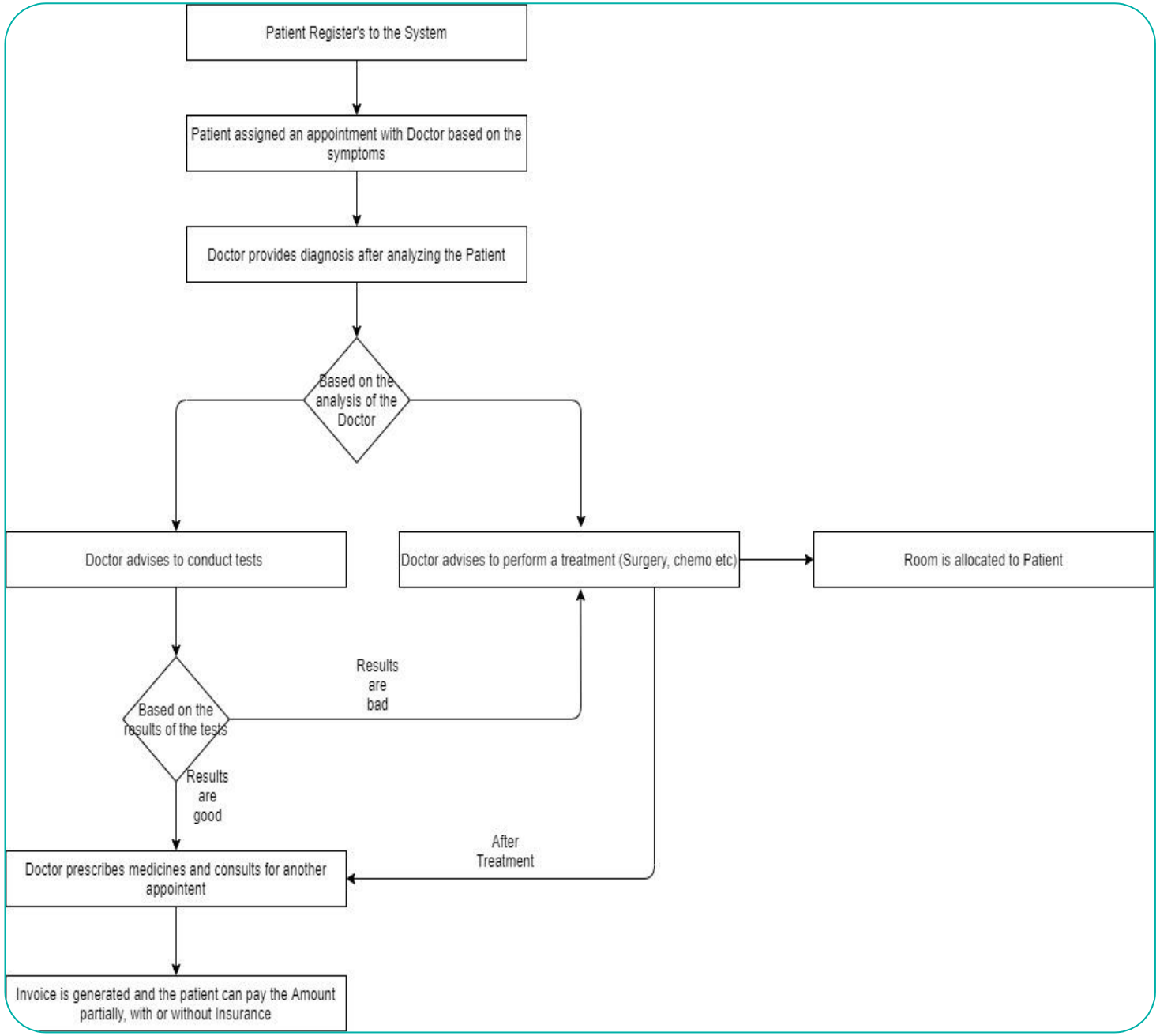
Anusha Peddigari	001023769
Chetan Dhowlaghar	001023435
Parul Vig	001356967
Samhitha Vinjamoori	001086599

Problem Statement

- Medical officers and staff have a great scope for being an erroneous employee
- The resources are not always efficiently used
- Physical storage of bulky records makes it difficult to retrieve data and also poses a security threat
- The services become expensive and sub-optimal

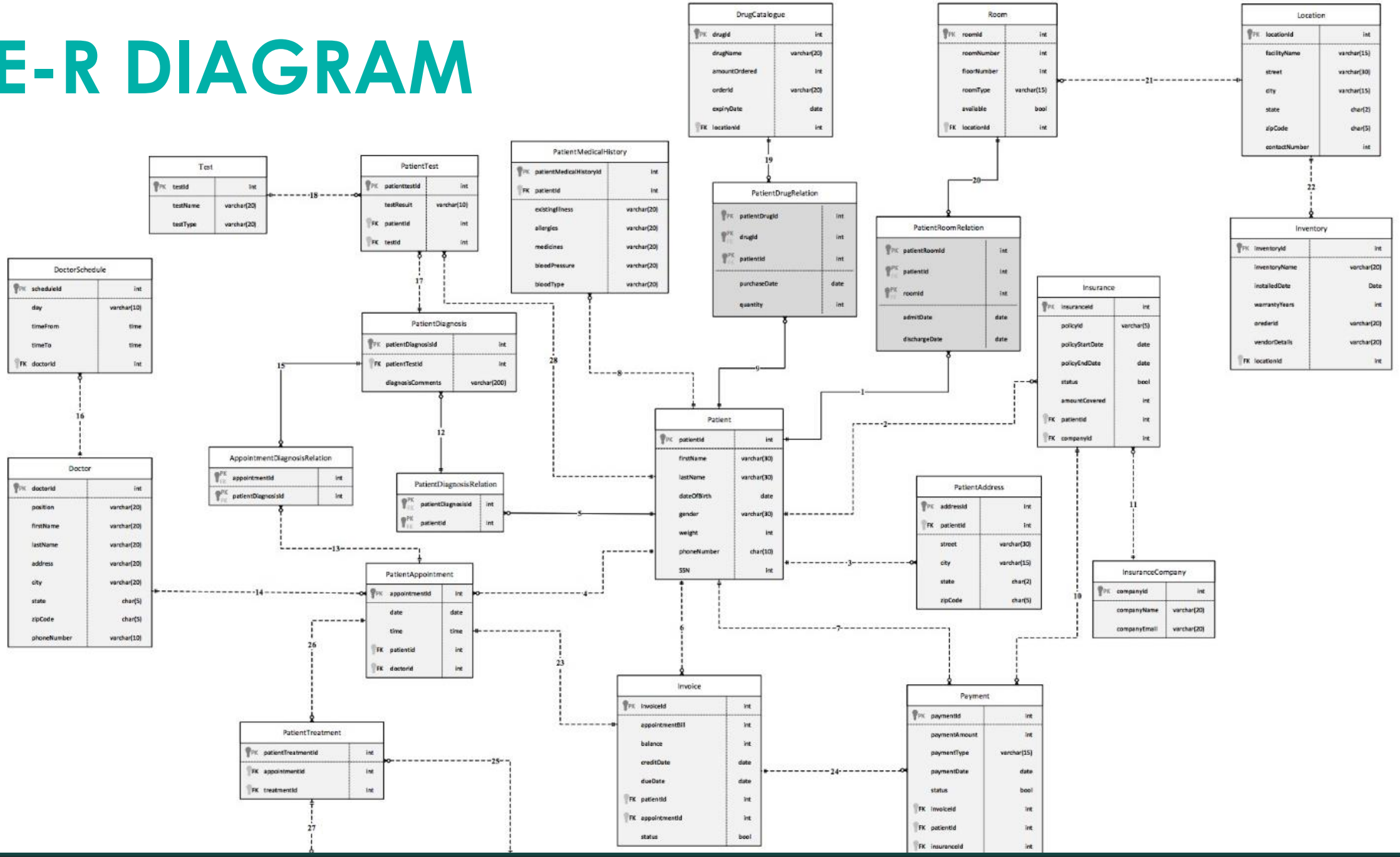
Objective Highlights

- To retrieve information faster than traditional systems
- Easing the appointment making process
- Reduce process costs
- Draw analytics to understand the data secured
- Efficient use of resources
- Reduced human errors



High Level Approach

E-R DIAGRAM


DrugCatalogue
drugId int
drugName varchar(20)
amountOrdered int
orderId varchar(20)
expiryDate date
locationId int
Room
roomId int
roomNumber int
floorNumber int
roomType varchar(15)
available bool
locationId int
Location
locationId int
facilityName varchar(15)
street varchar(30)
city varchar(15)
state char(2)
zipCode char(5)
contactNumber int
Test
testId int
testName varchar(20)
testType varchar(20)
PatientTest
patienttestId int
testResult varchar(10)
patientId int
testId int
PatientMedicalHistory
patientMedicalHistoryId int
patientId int
existingIllness varchar(20)
allergies varchar(20)
medicines varchar(20)
bloodPressure varchar(20)
bloodType varchar(20)
PatientDrugRelation
patientDrugId int
drugId int
patientId int
purchaseDate date
quantity int
PatientRoomRelation
patientRoomId int
patientId int
roomId int
admitDate date
dischargeDate date
Inventory
inventoryId int
inventoryName varchar(20)
installedDate Date
warrantyYears int
orederId varchar(20)
vendorDetails varchar(20)
locationId int
DoctorSchedule
scheduleId int
day varchar(10)
timeFrom time
timeTo time
doctorId int
PatientDiagnosis
patientDiagnosisId int
patientTestId int
diagnosisComments varchar(200)
Insurance
insuranceId int
policyId varchar(5)
policyStartDate date
policyEndDate date
status bool
amountCovered int
patientId int
companyId int
Patient
patientId int
firstName varchar(30)
lastName varchar(30)
dateOfBirth date
gender varchar(30)
weight int
phoneNumber char(10)
SSN int
AppointmentDiagnosisRelation
appointmentId int
patientDiagnosisId int
PatientDiagnosisRelation
patientDiagnosisId int
patientId int
Doctor
doctorId int
position varchar(20)
firstName varchar(20)
lastName varchar(20)
address varchar(20)
city varchar(20)
state char(5)
zipCode char(5)
phoneNumber varchar(10)
PatientAddress
addressId int
patientId int
street varchar(30)
city varchar(15)
state char(2)
zipCode char(5)
InsuranceCompany
companyId int
companyName varchar(20)
companyEmail varchar(20)
PatientAppointment
appointmentId int
date date
time time
patientId int
doctorId int
Invoice
invoiceId int
appointmentBill int
balance int
creditDate date
dueDate date
patientId int
appointmentId int
status bool
Payment
paymentId int
paymentAmount int
paymentType varchar(15)
paymentDate date
status bool
invoiceId int
patientId int
insuranceId int
PatientTreatment
patientTreatmentId int
appointmentId int
treatmentId int

```sql
-- Table PatientDiagnosisRelation

CREATE TABLE PatientDiagnosisRelation
(
PatientDiagnosisID INT REFERENCES  PatientDiagnosis(PatientDiagnosisID),
PatientID INT REFERENCES Patient(PatientID),
Constraint PKPatientDiagnosisRelation PRIMARY KEY CLUSTERED(PatientDiagnosisID, PatientID)
);

-- Table Doctor

CREATE TABLE Doctor
(
DoctorID INT NOT NULL PRIMARY KEY,
Position VARCHAR(20) NOT NULL,
FirstName VARCHAR(15) NOT NULL,
LastName VARCHAR(15)NOT NULL,
Address VARCHAR(20) NOT NULL,
City VARCHAR(15) NOT NULL ,
State CHAR(2) NOT NULL,
ZipCode CHAR(5) NOT NULL CHECK (ZipCode like '[0-9][0-9][0-9][0-9][0-9]'),
PhoneNumber CHAR(10)NOT NULL CHECK (PhoneNumber like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][
);
```

2001	101	4	2
2002	102	15	12
2003	103	1	2
2004	104	0	0
2005	105	2	2
2006	106	0	2
2007	107	8	5
2008	108	13	10
2009	109	8	12

```sql
CREATE TABLE Patient
(
PatientID INT NOT NULL PRIMARY KEY ,
FirstName VARCHAR(15),
LastName VARCHAR(15),
DOB DATE NOT NULL CHECK(DOB < getdate()),
Gender VARCHAR(10),
PatientWeight INT,
PhoneNumber CHAR(10)NOT NULL CHECK (PhoneNumber like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
SSN INT
);

-- Table PatientAddress

CREATE TABLE PatientAddress
(
AddressID INT  NOT NULL PRIMARY KEY,
PatientID INT NOT NULL  REFERENCES Patient(PatientID),
Street VARCHAR(20) NOT NULL,
City VARCHAR(15) NOT NULL ,
State CHAR(2) NOT NULL,
ZipCode CHAR(5) NOT NULL CHECK (ZipCode like '[0-9][0-9][0-9][0-9][0-9]')
```

Data Entry

```sql
INSERT INTO Patient VALUES(111, 'Liam', 'Iran', '1996/02/16', 'Male', 65, 4133211531, 8776598)
```

00 %

Messages

```
Msg 547, Level 16, State 0, Line 364
The INSERT statement conflicted with the CHECK constraint "checkPatientRegistered". The conflict occurred in database "CancerTreatmentSystem", table "dbo.Patient".
The statement has been terminated.

Completion time: 2020-08-07T20:11:11.0090172-04:00
```

Error while adding same patient again

Run Cancel Disconnect Change Connection MediForms Explain Enable SQLCMD Export as Notebook

```sql
create trigger trigger_patient_record_log
On Patient
AFTER INSERT

AS
begin
    insert into dbo.patient_record_log
    SELECT PatientID, FirstName, LastName FROM inserted
end;

INSERT INTO Patient VALUES(191, 'Liam', 'Iran', '1996/02/16', 'Male', 65, 4133211531, 8776598);
SELECT * FROM patient_record_log;
```

New Patient Entry

PatientID	FirstName	LastName	DOB	Gender	PatientWeight	PhoneNumber	SSN
101	Liam	Iran	1/20/1960	Male	70	4133211531	8776598
102	Cassian	Smith	9/28/1960	Male	89	4133211542	6552187
103	Jim	Kelly	5/19/1960	Male	50	4133211553	8741987
104	Dan	Rock	11/22/1999	Male	66	4133215144	5813981
105	Kelly	Smith	2/21/1995	Male	88	4133211568	4719832

Triggers

Views

Balance calculated from Paid Amount

	PatientID	FirstName	LastName	PhoneNumber	invoiceID	balance	status
1	101	Liam	Iran	4133211531	7001	1000	1
2	102	Cassian	Smith	4133211542	7002	2300	0
3	103	Jim	Kelly	4133211553	7003	2000	1
4	104	Dan	Rock	4133215144	7004	6002	0
5	105	Kelly	Smith	4133211568	7005	7700	1
6	106	Pharell	Iran	4133211519	7006	400	0
7	107	Joel	Dough	4133115011	7007	2388	1
8	108	Louis	Smith	4133215300	7008	7100	0
9	109	karen	Lee	4133211506	7009	1200	0
1…	110	Larry	Dennis	4133315807	7010	700	1

Ward Details of the Patient

	PatientID	FirstName	LastName	PhoneNumber	roomNumber	FloorNumber	RoomType
1	101	Liam	Iran	4133211531	105	1	Personal Room
2	103	Jim	Kelly	4133211553	204	2	Personal Room
3	101	Liam	Iran	4133211531	201	2	Personal Room
4	105	Kelly	Smith	4133211568	301	3	Personal Room
5	104	Dan	Rock	4133215144	301	3	Personal Room
6	110	Larry	Dennis	4133315807	105	1	Personal Room
7	109	karen	Lee	4133211506	201	2	Personal Room
8	106	Pharell	Iran	4133211519	301	3	Personal Room
9	105	Kelly	Smith	4133211568	404	4	Personal Room

Functions

Calculating age from DOB

```sql
--COMPUTE COLUMN BASED ON A FUNCTION

--Calculate Age Column from PatientID and DOB

CREATE FUNCTION calculateAgeFromDOB(@PatientID INT)
RETURNS INT
AS
    BEGIN

    DECLARE @Age AS INT

    SELECT @Age = DATEDIFF(hour,P.DOB,GETDATE())/8766 FROM Patient P
    WHERE P.PatientID = @PatientID

    RETURN @Age
    END

ALTER TABLE dbo.Patient
ADD Age AS (dbo.calculateAgeFromDOB(PatientID));
```

Integrity constraint for Patient records

Run Cancel Disconnect Change Connection MediForms Explain Enable SQLCMD Export as Notebook

```sql

--TABLE LEVEL CHECK CONSTRAINT BASED ON FUNCTION

--Add a Constraint to not allow a patient with the same Name and phone Number to register
CREATE OR ALTER FUNCTION isPatientRegistered(@firstName VARCHAR(30), @lastName VARCHAR(30), @phoneNumber CHAR(10))
RETURNS INT
AS
BEGIN
    DECLARE @COUNT AS INT

    SELECT @COUNT = COUNT(PatientID) FROM Patient
    WHERE FirstName = @firstName AND LastName = @lastName AND PhoneNumber = @phoneNumber

    RETURN @COUNT
END
```

Procedures

```sql
-- Stored Procedure To add New Patient Appointment
CREATE PROCEDURE usp_AddNewPatientAppointment
        @ScheduleDay VARCHAR(10)
       ,@TimeFrom Time(7)
       ,@TimeTo Time(7)
       ,@DoctorFirstName VARCHAR(15)
       ,@DoctorLastName VARCHAR(15)
       ,@PatientFirstName VARCHAR(15)
       ,@PatientLastName VARCHAR(15)
       ,@AppointmentDate Date
       ,@OutputResult VARCHAR(1000) OUTPUT
AS
BEGIN
           Declare @PatientId INT =0 ;
           Declare @DoctorId INT=0 ;
           Declare @ScheduleId INT=0 ;
           Declare @AppointmentID INT=0 ;
           Select @PatientId = PatientId from Patient where LastName=@PatientLastName and FirstName = @PatientFirstName
            Select @DoctorId = DoctorId from Doctor where LastName=@DoctorLastName and FirstName = @DoctorFirstName
            Select @ScheduleId = ScheduleId from DoctorSchedule
            Select @AppointmentID = AppointmentId from PatientAppointment
           INSERT INTO dbo.DoctorSchedule
           (ScheduleID,ScheduleDay,TimeFrom,TimeTo,DoctorID)
           VALUES
           (@ScheduleId+1,@ScheduleDay,@TimeFrom,@TimeTo,@DoctorId)

              INSERT INTO dbo.PatientAppointment
           (AppointmentID,AppointmentDate,AppointmentTime,PatientID,DoctorID)
           VALUES
           (@AppointmentID+1,@AppointmentDate,@TimeFrom,@PatientId,@DoctorId)
       SET @OutputResult = 'SUCCESS'

END
GO
```

To add into the Patient and Doctor Schedule Relation simultaneously

```sql

-- Stored Procedure To add New Patient
CREATE PROCEDURE usp_AddNewPatient
        @FirstName VARCHAR(15)
       ,@LastName VARCHAR(15)
       ,@DOB DATE
       ,@Gender VARCHAR(10)
       ,@PatientWeight INT
       ,@PhoneNumber VARCHAR(10)
       ,@SSN INT
       ,@OutputResult VARCHAR(1000) OUTPUT
AS
BEGIN
           Declare @PatientId INT = 0;
           Select @PatientId = PatientId from Patient
           INSERT INTO dbo.Patient
           (PatientId,FirstName,LastName,DOB,Gender,PatientWeight,PhoneNumber,SSN)
           VALUES
           (@PatientId+1,@FirstName,@LastName,@DOB,@Gender,@PatientWeight,@PhoneNumber,@SSN)

       SET @OutputResult = 'SUCCESS'

END

GO
DECLARE @OutputResult VARCHAR(1000) = ''
EXEC usp_AddNewPatient
        @FirstName = 'MARTIN',@LastName ='DONNA',@DOB = '1987-04-19',@Gender = 'M',
        @PatientWeight = 156
       ,@PhoneNumber = '4089870098',@SSN = 1236666
       ,@OutputResult  = @OutputResult OUTPUT
SELECT @OutputResult
```

To make faster entry into the Patient Relation

Procedures

Run Cancel Disconnect Change Connection MediForms Explain Enable SQLCMD Export as Notebo

```sql

--STORED PROCEDURE TO CALCULATE REVENUE

CREATE PROCEDURE CalculateRevenue @YEAR INT, @MONTH INT = NULL,@Revenue AS VARCHAR(20) OUTPUT
AS
BEGIN

    IF @MONTH IS NOT NULL
    BEGIN
        SELECT @Revenue = SUM(Balance) FROM INVOICE
        WHERE YEAR(CreditDate) = @YEAR AND MONTH(CreditDate) = @MONTH AND Status = 1
    END

    ELSE
    BEGIN
        SELECT @Revenue = SUM(Balance) FROM INVOICE
        WHERE YEAR(CreditDate) = @YEAR AND Status = 1
    END

END

--To Calculate Yearly Revenue
DECLARE @YearlyRevenue AS VARCHAR(20)
EXEC CalculateRevenue @YEAR=2008, @MONTH = Null,@Revenue = @YearlyRevenue OUTPUT
Select @YearlyRevenue

--To Calculate Revenue for a particular month
DECLARE @MonthlyRevenue AS VARCHAR(20)
EXEC CalculateRevenue @YEAR=2000, @MONTH = 12,@Revenue = @MonthlyRevenue OUTPUT
Select @MonthlyRevenue

```

To calculate Revenue from an aggregate of the Invoices generated in a particular year.

Encryption

```sql
--Encryption on Payment

--CREATE MASTER KEY

CREATE MASTER KEY ENCRYPTION
BY PASSWORD = 'Payment2020$';

 -- CREATE CERTIFICATE

CREATE CERTIFICATE paymentcert

WITH SUBJECT = 'User Payment';


-- CREATE SYMMETRIC KEY

CREATE SYMMETRIC KEY payment_Key_1

WITH ALGORITHM = AES_256  -- it can be AES_128,AES_192,DES etc

ENCRYPTION BY CERTIFICATE paymentcert;

--Encryption

ALTER TABLE Payment ADD paymentamount_encrypt varbinary(MAX),paymentType_encrypt varbinary(MAX),paymentDate_encrypt va
```

AES_256 Encryption

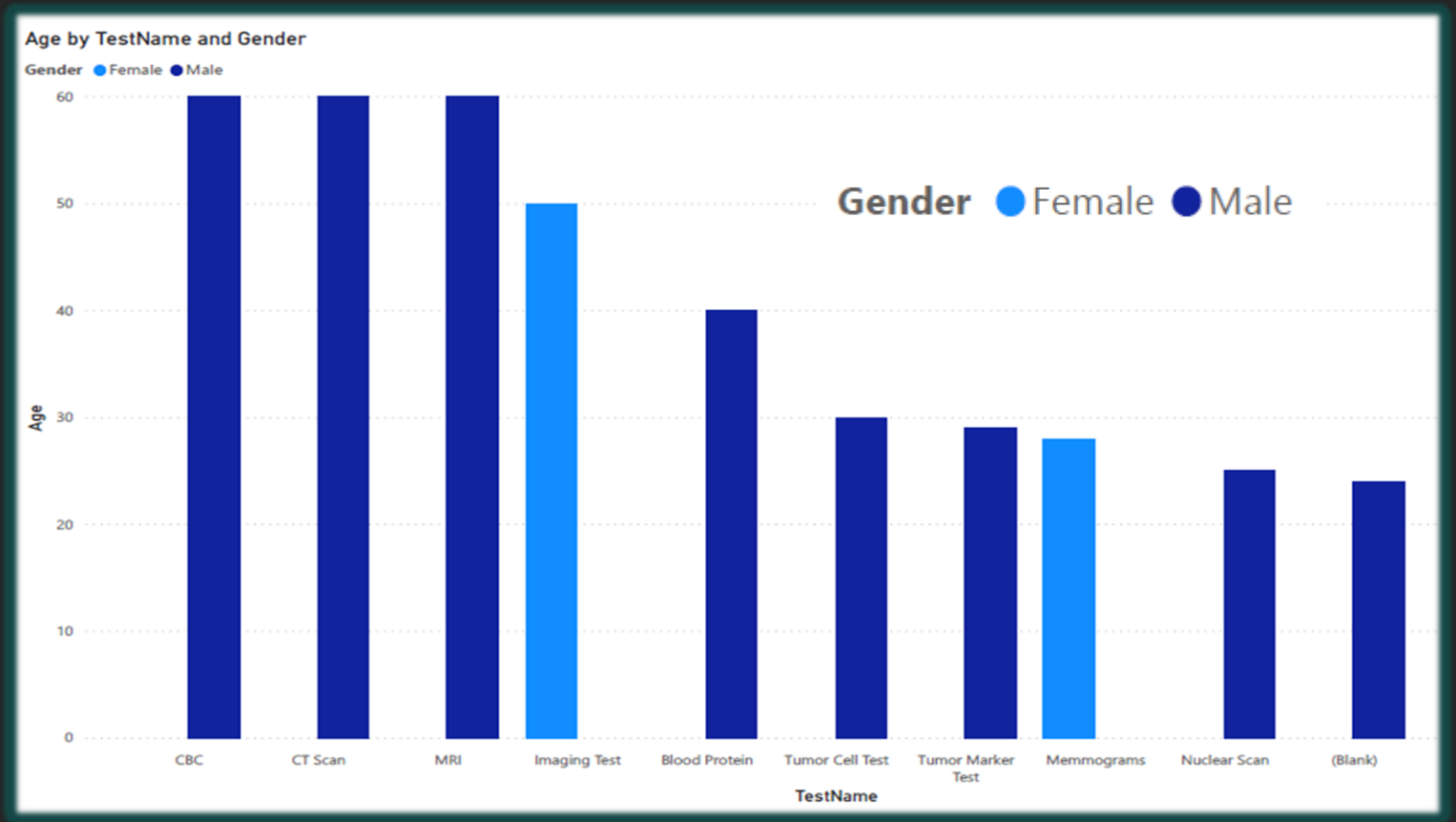
After Encryption

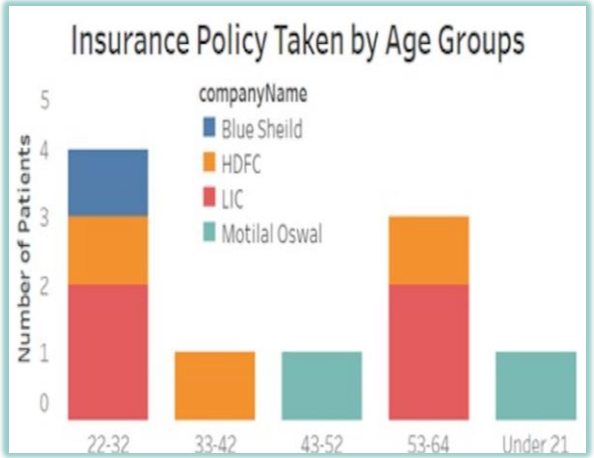
	paymentamount_encrypt
02000000B7FD7BE...	0x002FE1600CD0344180F14A55DFD75CB7020000006
02000000F35487E...	0x002FE1600CD0344180F14A55DFD75CB7020000001
02000000EAE4EF9...	0x002FE1600CD0344180F14A55DFD75CB7020000008
020000009D03FBD...	0x002FE1600CD0344180F14A55DFD75CB7020000007
02000000563057 8...	0x002FE1600CD0344180F14A55DFD75CB7020000009
02000000073BED8...	0x002FE1600CD0344180F14A55DFD75CB702000000E
02000000EFAC936...	0x002FE1600CD0344180F14A55DFD75CB7020000004
02000000103E689...	0x002FE1600CD0344180F14A55DFD75CB7020000007
0200000034B2FE3...	0x002FE1600CD0344180F14A55DFD75CB7020000005
020000003DFEA2F...	0x002FE1600CD0344180F14A55DFD75CB702000000E

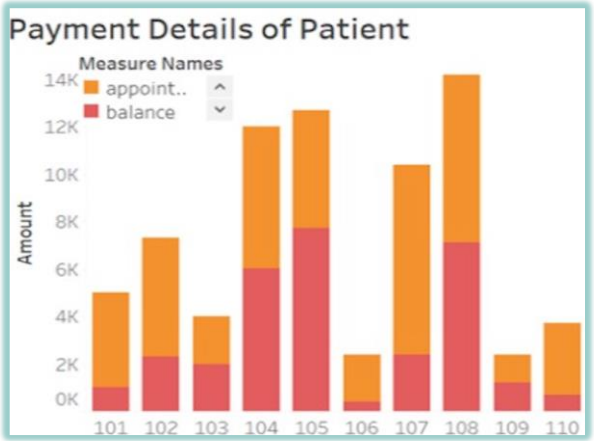
Before Encryption

Decrypted Payment amount	Decrypted Payment Type	Decrypted Payment Date
7000	Cash	2001-09-23
1200	Credit	2010-11-09
5699	Debit	2000-11-05
12997	Cash	2008-12-20
3999	Credit	2020-05-08
5677	Cheque	2010-04-03
12000	Echeque	2020-05-23
51299	Debit	2020-09-09
1700	Debit	2000-08-08
2100	Echeque	2020-05-08

Reports

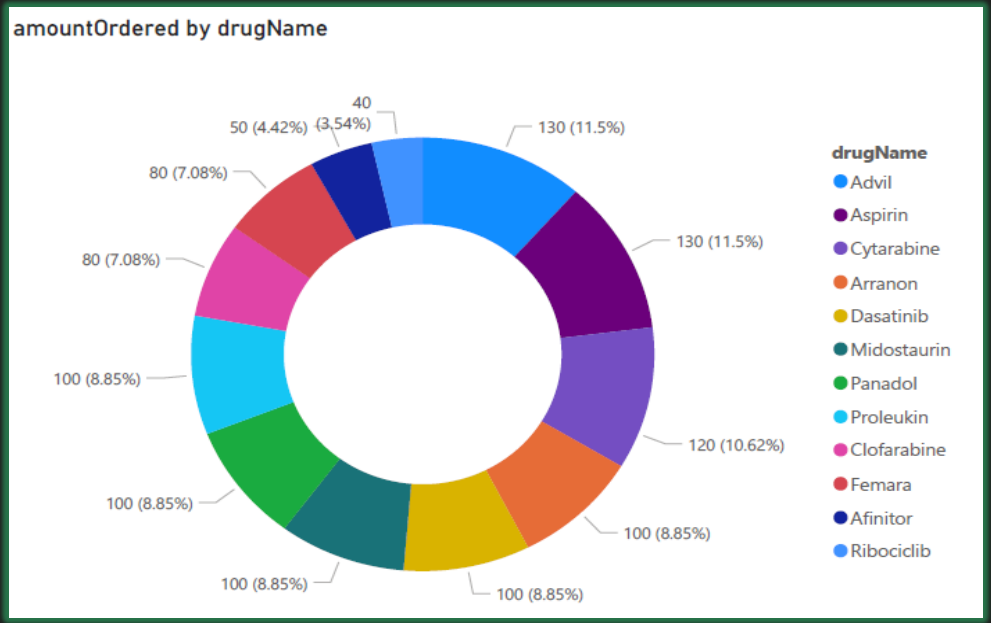

Age by TestName and Gender
Gender Female Male
Gender Female Male
Age
60
50
40
30
20
10
0
CBC
CT Scan
MRI
Imaging Test
Blood Protein
Tumor Cell Test
Tumor Marker Test
Memmograms
Nuclear Scan
(Blank)
TestName


Insurance Policy Taken by Age Groups
companyName
Blue Sheild
HDFC
LIC
Motilal Oswal
Number of Patients
5
4
3
2
1
0
22-32
33-42
43-52
53-64
Under 21

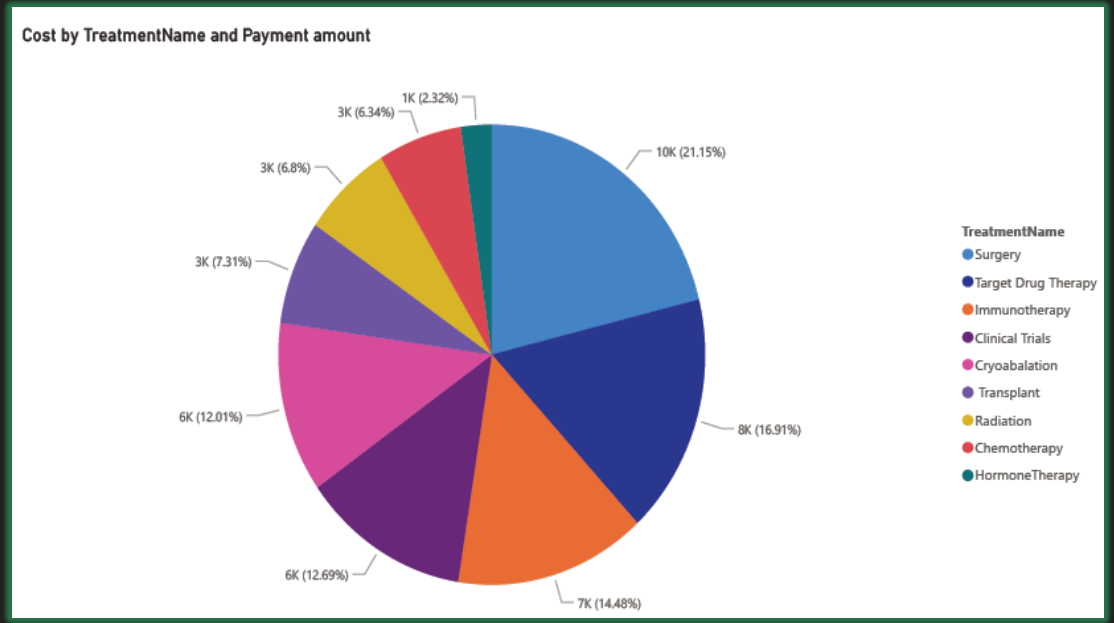

Payment Details of Patient
Measure Names
appoint..
balance
Amount
14K
12K
10K
8K
6K
4K
2K
0K
101
102
103
104
105
106
107
108
109
110

Visualization

Drug Name & Amount Ordered



Treatment and its Cost



Any Questions?

Thank You