Design of a Hospital-Based Database System (A Case Study of BIRDEM)

Rosina Surovi Khan
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology (AUST)
Dhaka, Bangladesh
Email: surovi99@yahoo.com

Mohammad Saber
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology (AUST)
Dhaka, Bangladesh
Email: saber@aust.edu

Abstract— **As technology advances, information in different organizations of Bangladesh can no more be maintained manually. There is a growing need for the information to become computerized so that it can be suitably stored. This is where databases come into the picture. Databases are convenient storage systems which can store large amounts of data and together with application programs such as interfaces they can aid in faster retrieval of data. An initiative was taken to design a complete database system for a hospital management such as Bangladesh Institute of Research and rehabilitation in Diabetes, Endocrine and Metabolic disorders (BIRDEM) in Dhaka so that its information can be stored, maintained, updated and retrieved conveniently and efficiently. The existing information in BIRDEM is partly computerized via databases only in patients' admissions, doctors' appointments and medical tests and reports sections. A partly slow and tedious manual system still exists in BIRDEM for example, in record of ambulances in service, assigning ward boys and nurses to rooms, the billing process and record of doctors' prescriptions etc. However, this paper outlines one complete database design for the entire BIRDEM hospital in which data maintenance and retrieval are in perfect harmony and speedy. Sample SQL-based queries executed on the designed system are also demonstrated.**

Keywords- ***Database system, entity relationship diagram, relational model, normalization, SQL-based queries*.**

Introduction

Bangladesh Institute of Research and rehabilitation in Diabetes, Endocrine and Metabolic disorders, abbreviated as BIRDEM, is a Research Center for Diabetes and also a hospital. There are doctors, patients, and employees like nurses, ward boys, ambulance carriers which are considered as entities in the designed system. It becomes a tough, tedious and comparatively slow process to store the information partly manually and partly computerized. But having all the necessary information stored in one database, it not only helps in orderly maintenance but very speedy retrieval of data.

The work in this paper has been supervised by Rosina Surovi Khan for her student Syed Mahboob Nur and his group as a thesis work in Year 4, Semester 1, CSE, AUST in Fall 2009.

I. Extraction of Information from Hospital Management (BIRDEM)

BIRDEM was visited and information was gathered; demos were seen and its websites visited from the internet. Now Ibrahim Medical College is founded in BIRDEM for the MBBS degree. A lot of students pass MBBS from this medical college. [3,4]

A database system was designed based on a case study of BIRDEM Hospital via Entity Relationship Diagram (ERD), Relational Model , Normalization of tables and Implementation in SQL server. [1,2,5] The ERD is outlined below.

II. Database Design

A. Entity Relationship Model (ERM)

1) ER Diagram

In the ER diagram, we can view the entities- Patient, Doctor, Receptionist, Department, Medicine, Test, OT (Operation Theater), Room, Nurse, Ward_boy, Driver, Ambulance, Carriers, Accountant and Bill. Among these entities, relationships exist which connect all the entities in the diagram. For example, Patient, Doctor and Receptionist are connected via the relationship Appointment. In other words, a receptionist will set up a doctor's appointment for a patient. Similarly, Doctor, Patient and Medicine are connected via the relationship Prescription. Here, a doctor may prescribe one or more medicine to a patient. In a similar way, other entities are connected via relationships in a meaningful way. Cardinality ratios [1] for the entities connected to a relationship are explained in the next section.

2) Cardinalities

Binary Relationship

We can see a binary relationship in the ER diagram.

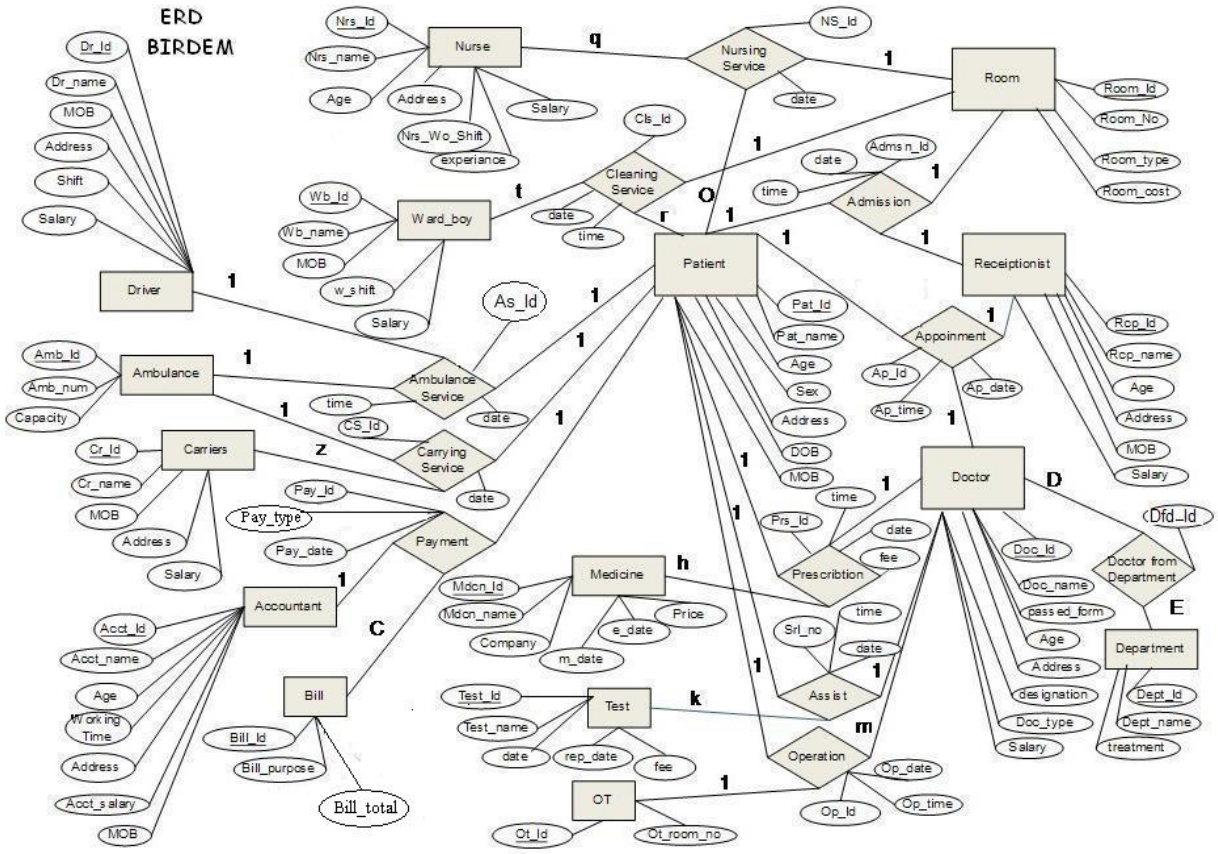

Figure 1. ER diagram of BIRDEM

As can be seen in Figure 2, 1 Doctor can be from 1 or many Departments . 1 Department may have 1 or many Doctors . So it is a many to many relationship named Doctor from Department (in the diamond).

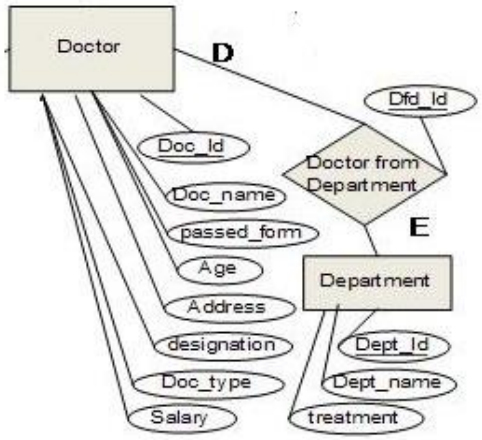

Figure 2. Example of Binary Relationship

Ternary Relationship

As seen in Figure 3, 1 Receptionist can admit 1 Patient in 1 Room in a certain date and time. 1 Receptionist can admit in 1 Room 1 Patient in a certain date and time. In 1 Room, 1 Patient is admitted by 1 Receptionist in a certain date and time. So the relationship is a ternary relationship named Admission (in the diamond) with the cardinality ratio from Patient to Receptionist to Room being 1 to 1 to 1.

Similar logic applies to other ternary relationships in the ER diagram of Figure1.

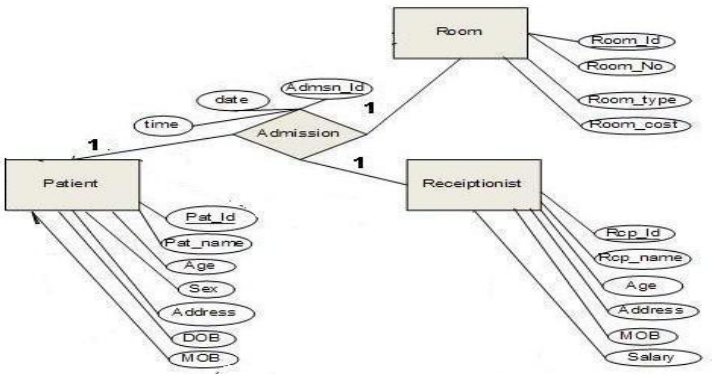

Figure 3. Example of ternary relationship

B. Relational Model

The entire ER diagram in sec A.1 can be converted to a Relational Model (relational tables) [1,2,5] as shown below. The attribute(s) of a relation which serve as a primary key of the table are underlined. Those attribute(s) which represent foreign keys are indicated as fk underneath .

i)Patient(Pat_id, Pat_name, age, sex, Address, DOB, MOB)

Here DOB refers to patient's date of birth and MOB refers to mobile number.

ii)Room(Room_id, Room_No, Room_type, Room_cost)

iii)Admission(admsn_id, Pat_id, Room_id, Rcp_id, date, time)
(fk) (fk) (fk)

This is a junction table among Patient, Room and Receptionist tables. Primary key of the Patient table goes to Admission table as foreign key. Primary key of the Room table goes to Admission table as foreign key. Primary key of the Receptionist table goes to Admission table as foreign key. Admsn_id is a primary key in the Admission table. Since the cardinality ratio from Patient to Receptionist to Room is 1 to 1 to 1, Pat_Id from Patient table, Room_Id from Room table and Rcp_Id from Receptionist table become foreign keys in the Admission junction table.

iv)Receptionist(rcp_id, rcp_name, Age, Address, MOB, Shifting)

Here Shifting refers to morning, afternoon or night shifts.

v)Doctor Table(Doc_id, Doc_name, Age, Address, Salary, MOB, Designation, Passed_from)

Here Passed_from indicates passed from which institution.

vi)Appointment(Ap_id, Pat_id, Doc_id, Rcp_id, Ap_date,
(fk) (fk) (fk)
Ap_time)

This is a junction table between Patient, Receptionist & Doctor tables.

vii)Bill(Bill_id, Bill_purpose, Bill_total)

Here Bill_purpose refers to the cause e.g blood test for which the bill is paid.

viii)Accountant(Acct_id, Acct_name, Age, Address, MOB, Working_time, Acct_salary)

ix)Payment Table (Pay_id, Bill_id, Pat_id, Acct_id, Pay_type,
(fk) (fk)
Pay_date)

This is a junction table between Patient, Bill & Accountant tables. Primary key of the Patient Table goes to Payment table as foreign key. Primary key of the Accountant Table goes to Payment table as foreign key. Primary Key of the Bill Table becomes a part of the primary key in Payment table. (Pay_id, Bill_id) is the primary key in the Payment table. Since the cardinality ratio from Patient to Accountant to Bill is 1 to 1 to C (which stands for *many*), Pat_Id from Patient table and Acct_id from Accountant table become foreign Keys in the Payment table while Bill_id from Bill table becomes a part of the primary key in the Payment table. In this table, pay_type refers to the type of payment such as cash, pay order, check or credit card.

x)Medicine(Mdcn_id, Mdcn_name, company, m_date, e_date, price)

m_date refers to manufacture date. e_date refers to expiry date.

xi)Prescription(Prs_id, Doc_id, Mdcn_id, Pat_id, date, fee)
(fk) (fk)

This is a junction table between Patient, Doctor & Medicine tables.

xii)Test(Test_id, Test_name, date, rep_date, fee)

Here rep-date refers to the date the report of the test will be given.

xiii)Assist(Srl_no, Pat_id, Doc_id, Test_id, time, date)
(fk) (fk)

This is a junction table between Patient, Doctor & Test tables. Srl_no corresponds to serial number of conducted test.

xiv)OT (Ot_id, Ot-room_no)

Here OT refers to Operation Theater.

xv) Operation(Op_id, Doc_id, Pat_id, Ot_id, Op_date,
(fk) (fk)
Op_time)

This is a junction table between Patient, Doctor & OT tables.

xvi) Department(Dept_id, Dept_name, treatment)

xvii) Doctor_from_Department(Dfd_id, Doc_id, Dept_id)

This is a junction table between Doctor & Department tables.

xviii)Nurse(Nrs_id, Nrs_name, Age, Address, MoB, Nrs_wo_shift, experience, salary)

Here Nrs_wo_shift refers to the working shift of the nurse such as morning, afternoon or night shifts.

xix) Nursing_Service(Ns_id, Pat_id , Nrs_id, Room_id)
(fk)

This is a junction table between Patient, Room & Nurse tables.

xx) Ward_Boy(Wb_id, Wb_name, MoB, w_shift, Salary)

Here w_shift refers to working shift of a ward boy.

xxi)Cleaning _Service(Cls_id, Pat_id, Wb_id, Room_id)
(fk)

This is a junction table between Patient , Room & Ward_Boy tables.

xxii)Driver(Dr_id, Dr_name, MoB, Address, Shift, Salary)

xxiii)Ambulance(Amb_id, Amb_num, Capacity)

xxiv)Ambulance_Service(As_id, Pat_id, Dr_id, Amb_id)
(fk) (fk) (fk)

This is a junction table between Patient, Driver & Ambulance tables.

xxv)Carriers(Cr_id, Cr_name, MOB, Address, Salary)

Carriers are those persons who carry patients from the ambulance to the hospital premises.

xxvi)Carrying_Service Table(CS_id, Cr_id, Amb_id, Pat_id)
(fk) (fk)

This is a junction table between Patient, Ambulance & Carriers tables.

C. Functional Dependencies and Normalization

1) Fulfillment of all normal forms

Example:

Room Table:-

Room(Room_id, Room_No, Room_type, Room_cost)

{Room_id} => {Room_No}

Here Room-id corresponds to serial numbers like 1, 2, 3, 4 etc. Room_No corresponds to designated numbers of rooms like 206, 307, 508 etc. A Functional Dependency exists from Room_No to Room_id because two different Room_Nos cannot correspond to the same Room_id.

Similarly, the following functional dependencies exist:

{Room_id} => {Room_type}

{Room_id} => {Room_cost}

Considering the above full functional dependencies, we now verify whether the relation fulfills Codd's stringent rules of normalization. [1,2,5]

The relation is in 1NF because the attributes of the relation do not have sub attributes.

The relation is in 2NF because non-primary keys are fully functionally dependent on primary key.

The relation is in 3NF because no transitive dependency exists from non-primary key to primary key.

The relation is in BCNF because there is no part of primary key that is fully functionally dependent on non primary key.

Similarly, all other relations of the system follow Codd's rules of normalization except the Payment table which is explained below.

2) Violation of Normal forms

Payment Table :-

This is a junction table between Patient, Bill & Accountant tables.

Payment(Pay_id, Pat_id, Bill_id, Acct_id, Pay_type,Pay_date)

For the above relation, the following functional dependencies exist:

{Pay_id}=> {Pay_type, Pay_date, Pat_id}

Two different patient ids, payment dates and payment types cannot correspond to the same payment id. So Pay_type, Pay-date and Pat_id are fully functionally dependent on Pay_id.

{Bill_id}=> {Acct_id , Pat_id}

Similarly two different accountant ids and patient ids cannot correspond to the same bill id. So Acct_Id and Pat_id are fully functionally dependent on Bill_id.

Based on the above functional dependencies:

The relation is in 1NF.

The relation is not in 2NF because all non-primary keys are not fully functionally dependent on the primary key (Pay_id, Bill_id). So we split the relation to make it 2NF.

Payment1(Pay_id, Pay_Type, Pay_date, Pat_id)

Payment2 (BP_id, Bill_id, Acct_id, Pat_id) {BP_id stands for Bill Pay id and corresponds to serial numbers of all bills such as 1, 2, 3 etc in ascending order. Bill_id on the other hand corresponds to the bill code numbers}.

The relations are now in 2NF.

The relations are in 3NF.

The relations are in BCNF.

D. Implementation in SQL Server

SQL server is a modern software where we can store huge amount of information via a database. In fact we have

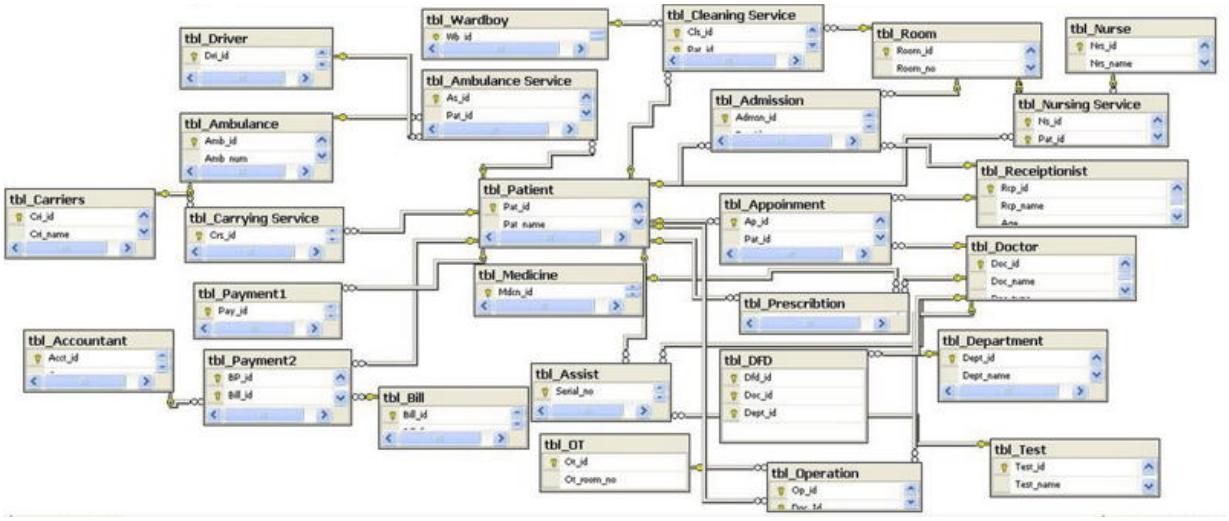

Figure 4. SQL Server Implementation Diagram

implemented our database system in SQL server. In this server we can store data easily, retrieve data speedily and execute the queries conveniently by SQL query language. We can create relations easily by code and also manually (drag and drop method like MS.Access).

III. USAGE OF THE SYSTEM

We can easily retrieve various data based on our demands using SQL Queries. Sample data has been entered in the relational tables. Some useful queries that can be imposed on our designed system but cannot be retrieved that speedily from the existing partly manual and partly computerized information system of BIRDEM are:

a) Ambulance Id 100 carried which patients to BIRDEM whom receptionist Sadia gave doctors' appointment for heart ailments?

b) Which patients were billed for above Tk5000 having gone through kidney tests suggested by Doctor Selima?

c) Doctors from which departments prescribed mainly antibiotics to patients?

d) How many patients with irregular ECG reports went though a heart surgery in Cardiology Department?

Sample queries directed on our system together with SQL expressions and results follow next.

Question 1

Which tests are suggested by doctor Selima to which patients ?

Query 1:

```
Select Pat_name, Doc_name, Test_name from tbl_Patient, tbl_Doctor, tbl_Test, tbl_Assist where Doc_name = 'Selima' and tbl_Doctor.Doc_id = tbl_Assist.Doc_id and tbl_Patient.Pat_id = tbl_Assist.Pat_id and tbl_Test.Test_id = tbl_Assist.Test_id
```

Output 1:

	Pat_name	Doc_name	Test_name
1	Moni	Selima	Blood
2	Karim	Selima	Blood
3	Moni	Selima	Engiogram

Question 2

Which doctors prescribed which medicine to patient Mamun ?

Query 2:

```
Select Pat_name, Doc_name, Mdcn_name from tbl_Patient, tbl_Doctor, tbl_Medicine, tbl_Prescription where Pat_name = 'Mamun' and tbl_Patient.Pat_id = tbl_Prescription.Pat_id and tbl_Doctor.Doc_id = tbl_Prescription.Doc_id and tbl_Medicine.Mdcn_id = tbl_Prescription .Mdcn_id
```

Output 2:

	Pat_name	Doc_name	Mdcn_name
1	Mamun	Mohammed	Omidon

Question 3 :

Which doctors are from which departments and they passed from which college and got salaries below Tk 20000 ?

Query 3

```
Select Doc_name, Passed_from, Dept_name from tbl_Doctor, tbl_Department, tbl_DFD where Salary <20000 and tbl_Doctor.Doc_id = tbl_DFD.Doc_id and tbl_Department.Dept_id = tbl_DFD.Dept_id
```

(Table tbl_DFD in SQL Server implementation diagram corresponds to the relationship Doctor From Department in ER diagram).

Output 3:

	Doc_name	Passed_from	Dept_name
1	Kibria	SMC	Cardiology
2	Rahima	KMC	Medicine

IV. CONCLUSION

Our database contains all the information needed to be maintained in a BIRDEM hospital. As we have computerized the entire system via a database, the maintenance is very convenient and efficient and also retrieval of data according to demand is speedy. The existing system of BIRDEM is partly manual and partly computerized and it becomes a tedious process to keep track of all the information partly in paper files and partly on computers. Therefore, our designed system is a good and useful implementation. We can further improvise it by enhancing its security. An initiative has also been taken to use Microsoft Visual Studio 2008 and the programming language C# for developing user friendly interfaces to the current database system. That way a software has been developed which is used to interface with the SQL Server and hence data accessed, retrieved and searched for far better in a more efficient and convenient form.

REFERENCES

[1] A. Silbershatz, H. Korth., S. Sudarshan, Database System Concepts, McGraw Hill, 5th Edition, 2005

[2] Database System Concepts Slides: http://www.cse.iitb.ac.in/sudarsha/db-book/slide-dir/ch1.pdf, Accessed 2010

[3] Diabetic Association of Bangladesh: http://www.dab-bd.org/, Accessed 2010

[4] BIRDEM hospital: http://www.dab-bd.org/project/birdem/birdem.php, Accessed 2010

[5] R. Elmasri., S. B. Navathe, Fundamentals of Database Systems, Addison-Wesley, 3rd Edition, 2000.

AUTHORS' PROFILES

The first author of the paper, Rosina Surovi Khan is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. Her medium of instruction has always been in English. She obtained her MSc in Software Technology from Stuttgart University of Applied Sciences, Germany. She also completed PhD coursework in Computer Science from the University of Texas at Arlington, USA.

The second author of this paper, Mohammad Saber is also currently working as an Assistant Professor in the Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. He obtained his MSc in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET).