CSC 401

DEVELOPMENT OF A MEDICAL RECORD DATABSE SYSTEM

GROUP 1

NAME	MATRIC NUMBER
AMUDA TOSIN	**090805009**
MATTHEW OMOLABAKE	**090805028**
OPEOLUWA JOSEPH	**090805048**
OLATUNJI MICHAEL	**090805044**
FAPOUNDA OLUWASEUN	**100805032**
SALAMI SHEHU TIJANI	**090805056**
OSINUGA OLUWASEUN	**090805049**

Tosin Amuda

UNIVERSITY OF LAGOS

CONTENTS

CONTENTS i

ACKNOWLEDGEMENT ii

INTRODUCTION 1

PROBLEM STATEMENT 1

BUSINESS RULE 2

CONCEPTUAL SCHEMA 3

ER DIAGRAM 3

RELATIONAL MODEL 5

Functional Dependencies 5

Relations 6

KEYS DEFINITION 7

CANDIDATE KEYS DEFINITION 7

FOREIGN KEYS DEFINITION 7

Domain Key Normal Form Definitions 9

POPULATING THE TABLES 13

QUERIES 18

1. JOINING TWO RELATIONS 18
2. JOINING THREE RELATIONS 19
3. JOINING FIVE RELATIONS 20

CONCLUSION 21

EXPERIENCE 21

REFERENCE 22

ACKNOWLEDGEMENT

We wish to acknowledge to excellent support that we have received to complete this paper. We would like to thank our instructor, Professor H.O.D Longe for your labour of love as you toil to bequeath us knowledge and introduce us into the field of Database.

Our heartfelt gratitude goes to Google.Com which gave answers to every possible question.

Finally our deepest gratitude however, goes out to the Almighty God who has been very good to us throughout from the beginning to the end of this project.

INTRODUCTION

PROBLEM STATEMENT

The medical record system is a database management system that uses database technology to construct, maintain and manipulate various kinds of data about a person's medical history and care across time. The DBMS can track and update all the information of registered patients in the medical Centre during a particular time span.

Medical records are created when a patient receive treatment from a health professional. Records may include the patients:

- personal information
- medical history
- details about lifestyle and family medical history
- laboratory test results
- medications prescribed
- reports that indicate the results of operations and other medical procedures
- disabilities, allergies and;
- life or accidental insurance with private insurers or government programs.

The medical record serves a variety of purposes and is essential to the proper functioning of the medical practice—especially in today's complicated health care environment. The medical record is a key instrument used in planning, evaluating, and coordinating patient care in both the inpatient and the outpatient settings. The content of the medical record is essential for patient care, accreditation (if applicable to the practitioner), and reimbursement purposes.

The medical record administrator may be a clinician, billing manager, coder, or anyone assigned the responsibility in the medical office. The medical record administrator is responsible for filing patient information in the medical record. He or she is also responsible for knowing medical insurance contract requirements, legal requirements pertaining to privacy and confidentiality of the patient.

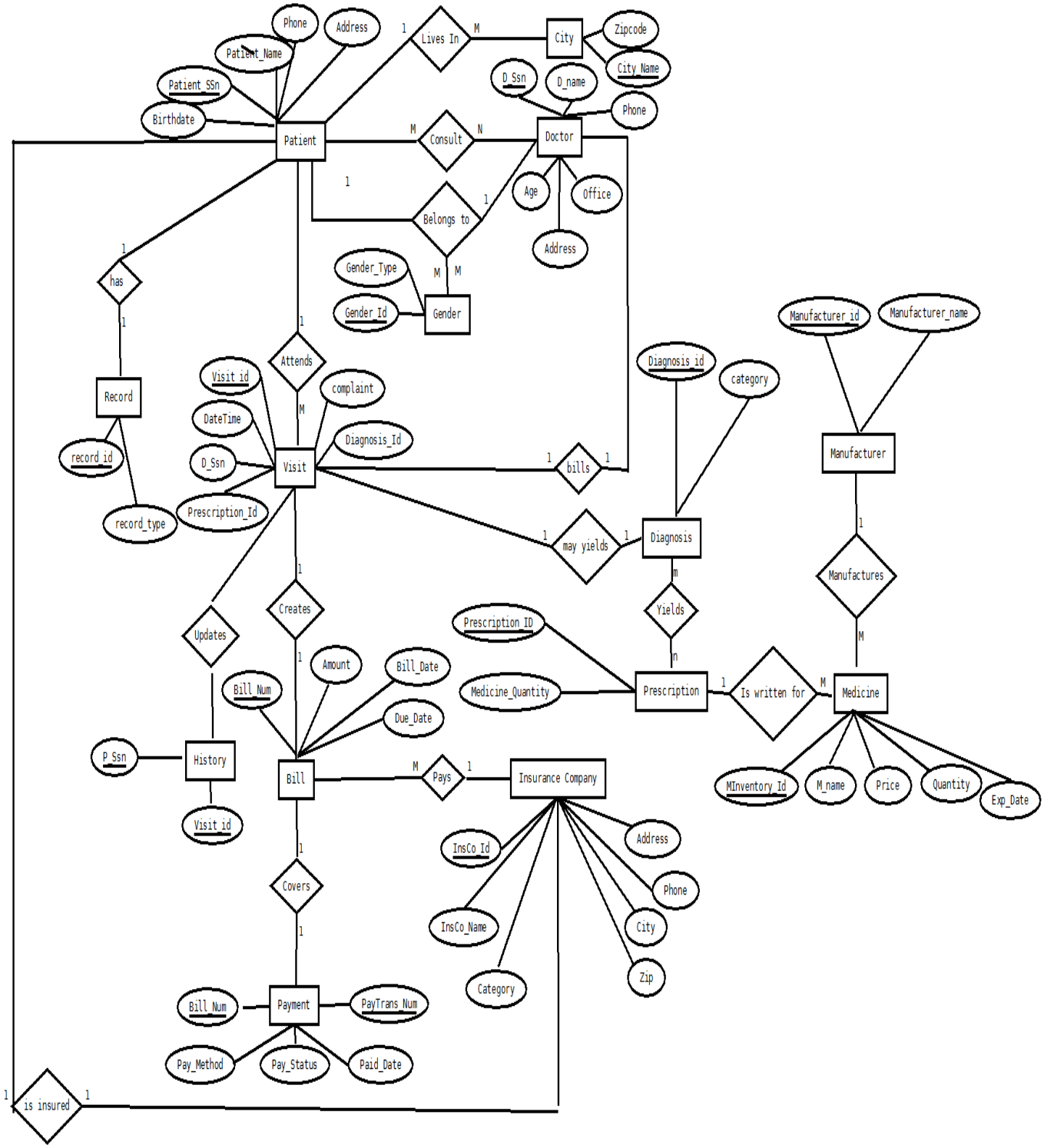
BUSINESS RULE

The business rule for a medical record database:

- A patient medical record is registered on the first visit to the doctor.
- A patient can make many appointments with one or more doctors.
- A doctor can accept appointments with many patients.
- However, each appointment is made with only one doctor, and each appointment references a single patient
- If an appointment yields a visit with a doctor, the visit yields a diagnosis and, when appropriate treatment.
- With each visit, the patient's history is updated.
- Each visit creates a bill. Each patient visit is billed by one doctor, and each doctor can bill many patients.
- Each bill must be paid. However, a bill may be paid in many installments, and a payment may cover more than one bill.
- A patient may pay the bill directly, or the bill may be the basis for a claim submitted to an insurance company.
- If a bill is paid by an insurance company, the deductible is submitted to the patient for payment.

CONCEPTUAL SCHEMA

ER DIAGRAM



Entity	Attributes
Doctor	D_name, D_SSn, Age, Phone, address, Office
Patient	P_Name, P_Ssn, BirthDate, Phone, address
City	ZIP, city_name
Gender	Gender_id, Gender_type
Visit	Visit_id, date, time, complaint , Diagnosis_Id, D_Ssn, Prescription_id
History	P_Ssn, Visit_id
Insurance	InsCo_Id, InsCo_Name, Category, Phone, Address, City, Zip
Bill	Bill_Num, Amount, Bill_Date, Due_Date
Payment	Bill_Num, PayTrans_Num, Pay_Method, Pay_Status, Paid_Date
Diagnosis	Diagnosis_Id, category
Prescription	Prescription_Id, medicine_Quantity
Record	record_id, record_type
Medicine	MInventory_Id, M_name, Price, Quantity, Exp_Date
Manufacturer	Manufacturer_id, Manufacturer_name

RELATIONAL MODEL

Functional Dependencies

Relation	Funtional dependencies
Doctor	D_SSn→D_name, Age, Phone, address, Office
	ZIP, D_Ssn → city_name
	Gender_id, D_Ssn → Gender_type
Patient	P_Ssn→ P_Name, Age, BirthDate, Phone, address
	ZIP, P_Ssn → city_name
	Gender_id, P_Ssn → Gender_type
City	ZIP→ city_name
Gender	Gender_id → Gender_type
Visit	Visit_id→date, time, complaint , Diagnosis_Id, D_Ssn, Prescription_id
History	P_Ssn, Visit_id→date, time, complaint , Diagnosis_Id, D_Ssn, Prescription_id
Insurance	InsCo_Id, InsCo_Name, Phone, Address
	InsCo_Id, P_Ssn→ InsCo_Name, Phone, Address
	ZIP, InsCo_Id → city_name
Bill	Bill_Num→Amount, Bill_Date, Due_Date
	Bill_Num, P_Ss, Diagnosis_Id, Prescription_Id→Amount, Bill_Date, Due_Date
Payment	PayTrans_Num→ Pay_Method, Pay_Status, Paid_Date
	Bill_Num, PayTrans_Num →Pay_Method, Pay_Status, Paid_Date
Diagnosis	Diagnosis_Id→ category
Prescription	Prescription_Id→ medicine_Quantity
Record	Record_id→Record_type
	Record_id, P_Ssn→ Record_type
Medicine	MInventory_Id→ M_name, Manufacturer, Price, Quantity, Exp_Date
	MInventory_Id, prescription_Id→M_name, Price, Quantity, Exp_Date
Manufacturer	Manufacturer_Id→Manufacturer_name

Relations

Based on the functional dependencies above, we normalized all relations and converted the ER model into a relational model:

Doctor *(D_Ssn,D_name, Age, Phone, address, Office)*
Patient *(P_Ssn, P_Name, BirthDate, Phone, address)*
Gender *(Gender_id, Gender_name)*
City *(ZIP, city_name)*
DoctorGender *(Gender_id, D_Ssn)*
DoctorCity *(ZIP, D_SSn)*
PatientGender *(Gender_id, P_Ssn)*
PatientCity *(ZIP, P_Ssn)*
Visit *(Visit_id, date, time, complaint, Diagnosis_Id, D_Ssn, Prescription_id)*
History *(P_Ssn, Visit_id)*
Insurance *(InsCo_Id, InsCo_Name, Phone, Address)*
Patient_Insurance *(InsCo_Id, P_Ssn)*
InsuranceCity *(InsCo_Id, ZIP)*
Bill *(Bill_Num,Amount, Bill_Date, Due_Date)*
PatientBill *(Bill_Num, P_Ssn, Diagnosis_Id, Prescription_Id)*
Payment *(PayTrans_Num, Pay_Method, Pay_Status, Paid_Date)*
BillPayment *(Bill_Num, PayTrans_Num)*
Diagnosis *(Diagnosis_Id, category)*
Prescription *(Prescription_Id, medicine_Quantity)*
Record *(record_id,record_type)*
PatientRecord *(record_id, P_Ssn)*
Medicine *(MInventory_Id, Manufacturer_id, M_name, Price, Quantity, Exp_Date)*
Manufacturer *(Manufacturer_Id, Manufacturer_name)*
PrescriptionMedicine *(MInventory_Id, prescription_Id)*

KEYS DEFINITION

CANDIDATE KEYS DEFINITION

Relations	Candidate keys (Primary Keys)
Doctor	D_Ssn
Patient	P_Ssn
Gender	Gender_id
City	ZIP
Insurance	InsCo_Id
Bill	Bill_Num
Payment	PayTrans_Num
Diagnosis	Diagnosis_id
Prescription	Prescription_id
Record	Record_id
Medicine	MInventory_Id
Manufacturer	Manufacturer_Id
Visit	Visit_Id

FOREIGN KEYS DEFINITION

Relations	Foreign Keys
DoctorGender	Gender_id REFERENCES Gender.Gender_id D_Ssn REFERENCES Doctor.D_Ssn
DoctorCity	D_Ssn REFERENCES Doctor.D_Ssn ZIP REFERENCES City.ZIP
PatientGender	Gender_id REFERENCES Gender.Gender_id P_Ssn REFERENCES Patient.P_Ssn
PatientCity	P_Ssn REFERENCES Patient.P_Ssn ZIP REFERENCES City.ZIP
Visit	Diagnosis_Id REFERENCES Diagnosis.Diagnosis_Id D_Ssn REFERENCES Doctor.D_Ssn Prescription_id REFERENCES Prescription. Prescription_id
History	P_Ssn REFERENCES Patient.P_Ssn Visit_id REFERENCES Visit.Visit_id
Patient_Insurance	InsCo_Id REFERENCES Insurance.InsCo_Id P_Ssn REFERENCES Patient.P_Ssn
InsuranceCity	InsCo_Id REFERENCES Insurance.InsCo_Id ZIP REFERENCES City.ZIP
PatientBill	Bill_Num REFERENCES Bill.Bill_Num P_Ssn REFERENCES Patient.P_Ssn Diagnosis_Id REFERENCES Diagnosis.Diagnosis_Id Prescription_id REFERENCES Prescription.Prescription_id
BillPayment	Bill_Num REFERENCES Bill.Bill_Num PayTrans_Num REFERENCES Payment.PayTrans_Num
PatientRecord	record_id REFERENCES Record.record_id P_Ssn REFERENCES Patient.P_Ssn

Medicine	Manufacturer_id REFERENCES Manufacturer.Manufacturer_id
PrescriptionMedicine	MInventory_Id REFERENCES Medicine.MInventory_Id Prescription_id REFERENCES Prescription.Prescription_id

Domain Key Normal Form Definitions

Relations	Constraints
DoctorGender	**Key:** (foreign)Gender_id (foreign) D_Ssn **Domain:** Gender_id: 10(Varchar), D_Ssn: 10(Varchar)
DoctorCity	**Key:** (foreign)ZIP (foreign)D_Ssn **Domain:** ZIP: 10(Varchar), D_Ssn: 10(Varchar)
PatientGender	**Key:** (foreign)Gender_id (foreign)P_Ssn **Domain:** Gender_id: 10(Varchar), P_Ssn: 10(Varchar)
PatientCity	**Key:** (foreign)ZIP (foreign) P_Ssn **Domain:** ZIP: 10(Varchar), P_Ssn: 10(Varchar)
Visit	**Key:** (primary)Visit_id (foreign)Diagnosis_Id (foreign)D_Ssn (foreign)Prescription_id **Domain:** Visit_id: 10(Varchar), date: Date, time: DateTime, complaint: text(255) , Diagnosis_Id: 10(Varchar), D_Ssn: 10(Varchar), Prescription_id: 10(Varchar)
History	**Key:** P_Ssn + Visit_id→date, time, complaint , Diagnosis_Id, D_Ssn, Prescription_id **Domain:** P_Ssn: 10(Varchar) ,Visit_id: 10(Varchar)
Patient_Insurance	**Key:** (Foreign)P_SSn (Foreign)InsCo_Id

	Domain: P_SSn: 10(Varchar), InsCo_Id:10(Varchar)
InsuranceCity	**Key:** (foreign) InsCo_Id (foreign) ZIP **Domain:** InsCo_Id: 10(Varchar), ZIP: 10(Varchar)
PatientBill	**Key:** (Foreign)Bill_Num (Foreign)P_Ssn (Foreign)Diagnosis_Id (Foreign) Prescription_Id) **Domain:** Bill_Num: 10(Varchar), P_Ssn: 10(Varchar) Diagnosis_Id: 10(Varchar), Prescription_Id: 10(Varchar)
BillPayment	**Key:** (Foreign)Bill_Num (Foreign)PayTrans_Num **Domain:** Bill_Num: 10(Varchar), PayTrans_Num:10(Varchar)
PatientRecord	**Key:** (Foreign)record_id (Foreign) P_Ssn **Domain:** record_id: 10(Varchar), P_Ssn: 10(Varchar)
Medicine	**Key:** (Primary)MInventory_Id (Foreign)Manufacturer_id **Domain:** MInventory_Id: 10(Varchar), Manufacturer_id: 10(Varchar), M_name: 30 (String), Price: 10(int), Quantity: 10(in), Exp_Date: Date
PrescriptionMedicine	**Key:** (Foreign) MInventory_Id (Foreign) prescription_Id **Domain:** MInventory_Id: 10(Varchar) , prescription_Id: 10(Varchar)
Doctor	**Key:** (Primary Key) D_Ssn **Domain:**

	D_Ssn: 10(Varchar), D_name: 30(String), Age: 3(int), Phone:15(Varchar), address: 50(Varchar), Office: 50(Varchar)
Patient	**Key:** (Primary)P_Ssn **Domain:** P_Ssn:10(Varchar), P_Name:30(String), BirthDate: Date, Phone: 15(Varchar), address: 50(Varchar)
Gender	**Key:** (primary) Gender_id **Domain:** Gender_id: 10(Varchar), Gender_name: male, female, hermaphrodite
City	**Key:** (primary) ZIP **Domain:** ZIP: 10(Varchar) , city_name: 30(String)
Insurance	**Key:** (Primary)InsCo_Id **Domain:** InsCo_Id: 10(Varchar), InsCo_Name: 30(String), Phone:15(Varchar), Address: 30(Varchar)
Bill	**Key:** (Primary)Bill_Num **Domain:** Bill_Num 10(Varchar) ,Amount 10(int), Bill_Date: Date, Due_Date: Date
Payment	**Key:** (Primary)PayTrans_Num **Domain:** PayTrans_Num: 10(Varchar) , Pay_Method : (insurance, self), Pay_Status:(paid, pending, not paid), Paid_Date: Date
Diagnosis	**Key:** (Primary)Diagnosis_Id **Domain:** Diagnosis_Id: 10(Varchar) , category: 30(String)
Prescription	**Key:** (Primary)Prescription_Id **Domain:** Prescription_Id: 10 (Varchar), medicine_Quantity:

	10(int)
Record	**Key:** (primary) record_id **Domain:** record_id: 10(Varchar), record_type: 30(VarChar)
Medicine	**Key:** (Primary)MInventory_Id (Foreign)Manufacturer_id **Domain:** MInventory_Id: 10(Varchar), Manufacturer_id:10(Varchar), M_name: 30(String), Price: 10(int), Quantity:10(int), Exp_Date: Date
Manufacturer	**Key:** (Primary)Manufacturer_Id **Domain:** Manufacturer_Id:10(Varchar), Manufacturer_name: 30(String)

POPULATING THE TABLES

1. **Doctor table**

D_Ssn	D_name	Age	Address	office	phone
999001	Ade Smith	52	48, Olorode street,apapa	2, lawani street, akoka	08027168240
999002	Bola Taiwo	34	3, broadstreet, lagos island	4, burma street, Apapa	08033160345
999003	Omotola Lawal	30	3, Malvin road, akoka	47, Pako street, festac	08045623100
999004	Chidinma Okafor	41	34, Alvan Nkoku lane, Unilag	University of lagos	08145672311
999005	Taiwo Olorede	44	8, Ebenezer street, Surulere	91, Aguda lane, surulere	08099347823
999006	Nelly Obot	35	2A, mushin road, surulere	LUTH	08045238908

2. **Patient table**

P_Ssn	P_name	BirthDate	Phone	Address
p001	Kemi Oyinlade	1990-10-15	08045672341	2B, festac town, Festac
p002	Nne Nwachukwu	1982-1-5	0803189222	50, Boatman street, Isheri road
p003	Michael Biodun	1982-4-23	08066128440	Jaja hall, University of Lagos
p004	Ann Ogbuagu	1992-5-11	08167459900	6B, fola agoro, Akoka

3. **Gender table**

Gender_id	Gender_name
gd1	Male
gd2	Female
gd3	Hermaphrodite

4. **City table**

Zip	city_name
.00211	Maryland Lagos
.00213	Akoka Lagos
.00253	Mushin Lagos
.00271	ikeja Lagos

5. **Insurance table**

InsCo_Id	InsCo_name	Phone	Address
INS001	FUG	012244556	4A, Broad street, Lagos island
INS002	BISCO INSURANCE	014567890	23, tokyo street, Mushin road
INS003	INTERCONTINENTAL	08033145646	University of Lagos

6. **Bill table**

Bill_Num	Amount	Bill_Date	Due_Date
B0001	5,000	2013-1-23	2013-3-23
B0002	2,000	2013-2-5	2013-4-5
B003	13,000	2012-10-9	2012-12-9
B0004	1,500	2012-12-22	2013-2-22

7. **Payment table**

PayTrans_Num	Pay_Method	Pay_status	Paid_Date
P0001	Insurance	Paid	2013-2-15
P0002	Self	Pending	2012-12-04

8. **Diagnosis table**

Diagnosis_id	category
dg0001	Hypertension
dg0002	HIV/ AIDS
dg0003	Conjuctivis
dg0004	Malaria

9. **Record table**

record_id	record_type
bg001	Blood Group A
bg002	Blood Group B
gen01	genotype AA
gen02	genotype AS
here001	Sickle cell anemia
here002	Down Syndrome
aller001	Seasonal allergy
aller002	drug allergy

10. **Medicine table**

MInventory_Id	Manufacturer_Id	M_name	Price	quantity	Exp_Date
001	Man004	Paracetamol	N50	1	2015-10-10
002	Man003	Amoxil	N230	2	2014-1-5
003	Man005	Ampiclox	N1000	3	2014-02-04
004	Man003	Novalgin	N100	1	2013-12-07

11. **Prescription table**

prescription_id	medicine_quantity
p001	2
p002	1
p003	1.5
p004	4

12. **Manufacturer table**

Manufacturer_Id	Manufacturer_name
Man001	M & B
Man002	Emzor
Man003	Pfizer
Man004	Beecham
Man005	Hulet

13. **DoctorGender table**

Gender_id	D_Ssn
gd1	999001
gd2	999002
gd2	999003
gd2	999004
gd1	999005
gd2	999006

14. **DoctorCity table**

Zip	D_Ssn
100213	999001
100213	999004

15. **PatientGender table**

P_Ssn	Gender_id
p001	gd2
p002	gd2
p003	gd1
p004	gd2

16. **PatientCity table**

Zip	P_Ssn
100213	p003
100213	p004

17. **Visit table**

Visit_id	Visit_date	Time	Complaint	Diagnosis_id	D_Ssn	prescription_id
V0001	2012-10-3	12.00	Constant headache	dg0001	999003	p002
V0002	2013-01-14	15.00	high temperature	dg0004	999006	p004
V0004	2013-01-03	13.00	high fever	dg0004	999003	p004

18. **History table**

Visit_id	P_Ssn
V0001	p001
V0002	p001

19. **Patient_Insurance table**

P_Ssn	InsCo_Id
p001	INS001
p002	INS001
p003	INS003

20. **InsuranceCity table**

InsCo_Id	Zip
INS003	100213

21. **patientBill table**

Bill_Num	P_Ssn	Diagnosis_id	prescription_id
B0001	p001	dg0001	p002

22. **BillPayment**

Bill_Num	PayTrans_Num
B0001	P0001
B003	P0002

23. **PatientRecord table**

record_id	P_Ssn
bg001	p001
gen01	p001
bg001	p002
gen02	p002
aller002	p002
here002	p002

24. PrescriptionMedicine table

prescription_id	MInventory_Id
p001	001
p002	003

QUERIES

1. JOINING TWO RELATIONS

SHOWING ALL THE PATIENTS WITH INSURANCE

```
SELECT patient.P_name, insurance.InsCo_name FROM insurance
JOIN patient_insurance ON insurance.InsCo_Id = patient_insurance.InsCo_Id
JOIN patient ON patient_insurance.P_Ssn = patient.P_Ssn;
```

SQL query:

```
SELECT patient.P_name, insurance.InsCo_name
FROM insurance
JOIN patient_insurance ON insurance.InsCo_Id = patient_insurance.InsCo_Id
JOIN patient ON patient_insurance.P_Ssn = patient.P_Ssn
LIMIT 0 , 30;
```

Show : 30 row(s) starting from record # 0

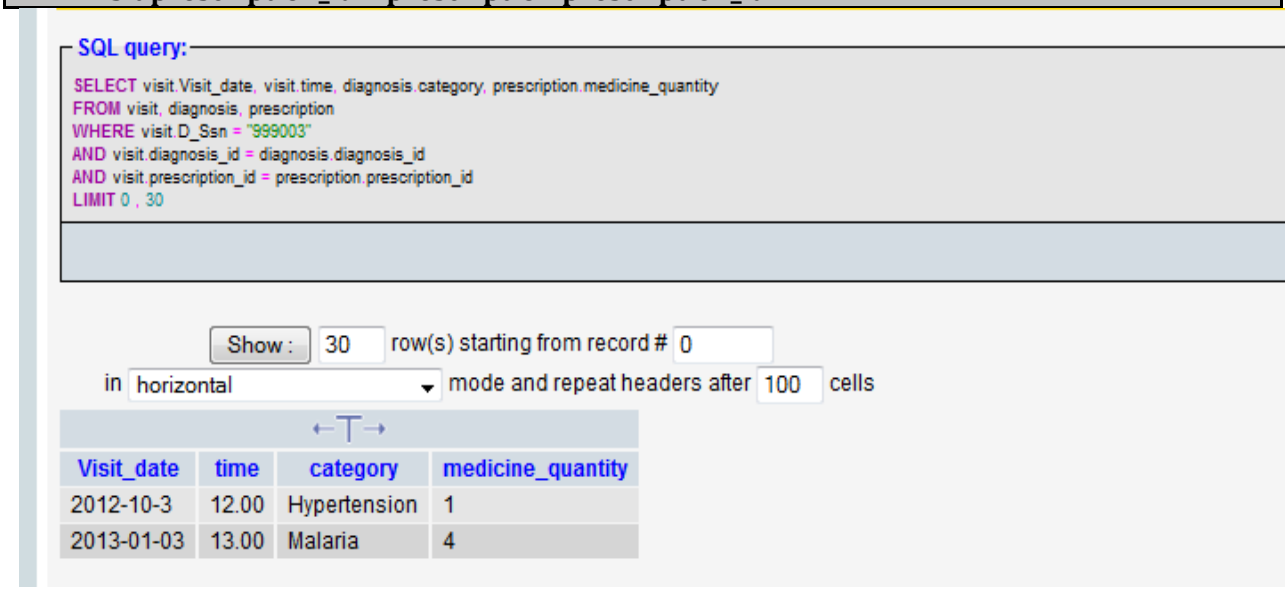
in horizontal mode and repeat headers after 100 cells

P_name	InsCo_name
Kemi Oyinlade	FUG
Nne Nwachukwu	FUG
Michael Biodun	INTERCONTINENTAL

2. JOINING THREE RELATIONS

GETTING ALL THE VISITS GRANTED BY A PARTICULAR DOCTOR

```
SELECT visit.Visit_date, visit.time, diagnosis.category, prescription.medicine_quantity
FROM visit, diagnosis, prescription
WHERE visit.D_Ssn = "999003"
AND visit.diagnosis_id = diagnosis.diagnosis_id
AND visit.prescription_id = prescription.prescription_id
```



3. JOINING FIVE RELATIONS

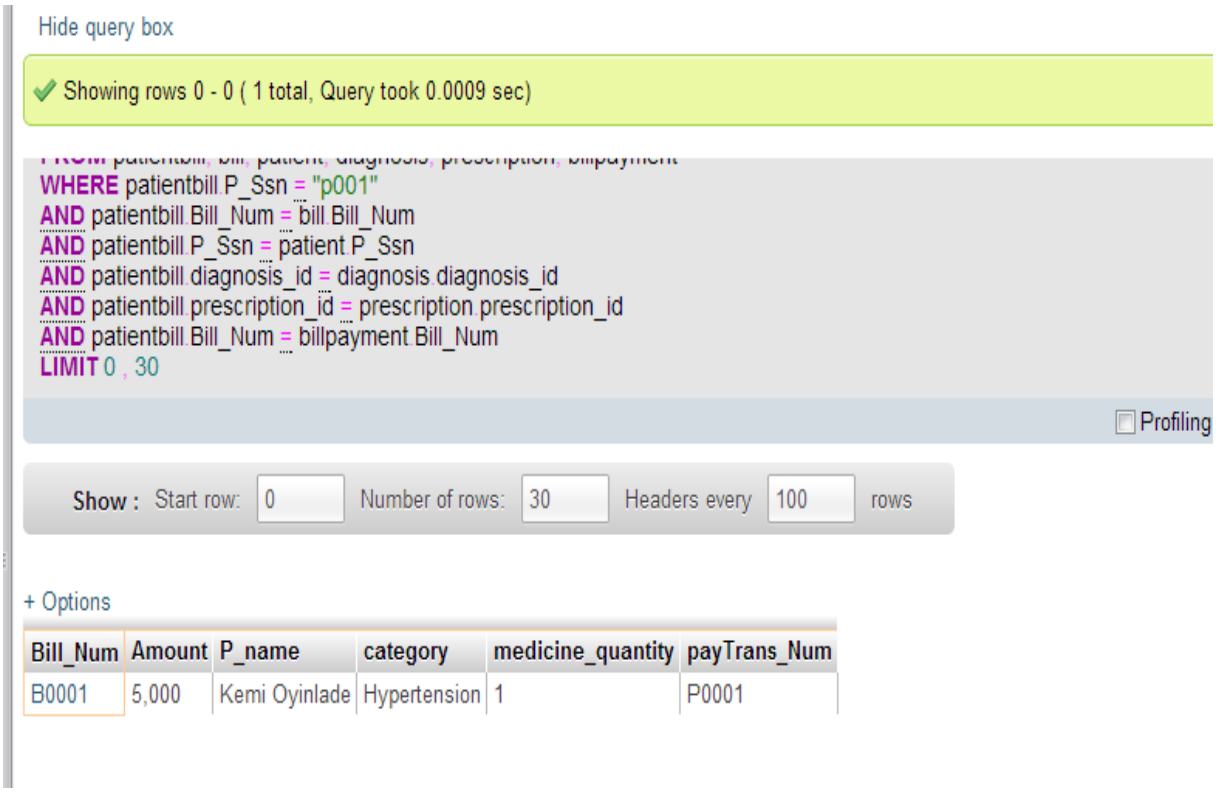
GETTING A PATIENT'S BILL DETAILS AND PAYMENT NUMBER

```
SELECT patientbill.Bill_Num, Bill.Amount, patient.P_name, diagnosis.category,
prescription.medicine_quantity, billpayment.payTrans_Num

from patientbill, bill, patient, diagnosis, prescription,billpayment

where patientbill.P_Ssn="p001"
AND

patientbill.Bill_Num = bill.Bill_Num
AND
patientbill.P_Ssn = patient.P_Ssn
AND
patientbill.diagnosis_id = diagnosis.diagnosis_id
AND
patientbill.prescription_id = prescription.prescription_id
AND
patientbill.Bill_Num = billpayment.Bill_Num;
```



CONCLUSION

EXPERIENCE

Merging the relations was made easy due to the fact that foreign keys existed that referenced the various tables that were merged.

The SQL statement became complex and more error-prone as the number of relations to be merged increased.

The Database Management System used is MYSQL.

REFERENCE

- Lecture notes
- http://www.wikipedia.com
- http://www.books.google.com
- http://www.business.mcmaster.ca
- Database Application Design by Dragomir R. Radev
- A Normal Form for relational databases that is not based on domains and keys(Article) By Ronald Fagin.