

CPS510: Database Systems 1

Section 10

Dr. S. B. Tajali

November 29, 2024

Eric Wong: 501174088

Mikayla Morrison: 501182358

Rohan Manoharan: 501189408

Patient Information and Record Management System: PatientLink

Table of Contents

Introduction (Assignment 1).....	2
System Design Progress.....	4
Final Entity-Relationship Diagram (Assignment 2).....	5
Final Design Overview (Assignment 6, 7, 8).....	6
Query and Results Analysis (Assignment 4 and 5).....	11
User-Interface (Assignment 9).....	22
Appendix A: Original Conceptual Design (Assignment 1).....	27
Appendix B: 3NF Justification (Assignment 7).....	30
Appendix C: Hypothetical Bernstein's Algorithm Application.....	35
and Hypotential BCNF Decomposition (Assignment 8).....	35

Introduction (Assignment 1)

Background

As the medical industry continues to evolve, technology has created tremendous opportunities to enhance the quality and efficiency of patient care. While physical technologies have made remarkable advancements, databases have played a vital role in advancing research, developing innovative treatments, and streamlining operations in medical institutions worldwide.

In alignment with global initiatives to digitize health services, electronic patient records have become a cornerstone for healthcare providers in accessing vital medical information. With this in mind, our team developed the PatientLink database. This system serves as a reliable, efficient, and secure platform, enabling medical staff to access patient records from any device, at any time.

Purpose

The purpose of this document is to outline the key features of the completed PatientLink database. Additionally, it will serve as a comprehensive record of the steps taken to develop its logical design, showcasing the extent to which the initial requirements were met upon completion.

Application Description

PatientLink is a secure database designed to provide doctors, nurses, and medical receptionists with a comprehensive, centralized view of patient medical records. Developed specifically for clinic environments, the system streamlines the management and accessibility of patient data, enabling healthcare providers to retrieve information quickly and accurately, ensuring high-quality and efficient care.

The development of PatientLink was guided by the following assumptions:

Clinic-Based Focus: The system is tailored for use in clinics, not hospitals. This design ensures that the database efficiently supports routine and outpatient care without features for inpatient and overnight services.

Geographic Limitation: The system is designed for use in Ontario only. Data formatting standards are consistent with provincial norms, including health card numbers, phone numbers, and credit card numbers, enhancing data validation and consistency.

Compliance with Ontario's Personal Health Information Protection Act: The system must comply with legal standards for patient record maintenance and retention in an effort to protect patient data and enforce privacy. As such, the system must be in accordance with the Personal Health Information Protection Act, 2004, S.O. 2004, c. 3, Sched. A¹.

The records managed by PatientLink include:

Patient Contact Information: Includes the patient's full name, address, phone numbers, email, and other contact preferences. Ensures that healthcare providers can quickly reach out for appointment confirmations, follow-ups, or urgent communications.

Emergency Contact Information: Identifies the names and contact details of individuals that are to be notified in case of an emergency with a patient.

Appointments: Tracks all past and upcoming appointments, including dates, times, diagnosis, treatment plan, and the healthcare professional involved. Allows for booking, rescheduling or cancellation, minimizing disruptions to the patient's care routine.

Billing Information: Simplifies financial management by maintaining detailed records of patient invoices, payment statuses, and insurance details where applicable.

Medical History: Provides a comprehensive record of the patient's past diagnoses, treatments, prescriptions, and surgical procedures.

Immunization Information: Tracks vaccinations and immunization schedules, ensuring compliance with public health requirements.

Prescriptions and Medications: Maintains a complete history of medications prescribed to the patient, including dosage and duration. Allows medical staff to review past prescriptions to prevent errors and renew prescriptions.

¹ Law document english view. Ontario.ca. (2018, November 19). <https://www.ontario.ca/laws/statute/04p03/v7>

System Design Progress

The completed logical design of PatientLink closely aligns with the conceptual design outlined in Assignment 1. This includes all tables, attributes, data types, and data formats detailed in Appendix A.

The most notable difference between the original conceptual design in Appendix A and the logical design is the addition of a primary key to every table. This modification ensures that each tuple in each table has a unique identifier; each primary key also serves as a foreign key that associates each patient with a specific set of records. This inclusion facilitated the creation of tables that complied with the 3NF normalization forms. The justification and tables outlined in Appendix B demonstrate that each table follows the 3NF normalization form.

The Emergency Contact Information Table was decomposed into two separate tables: Emergency Contacts and Emergency Relations. This decomposition was necessary because the original model allowed patients to have multiple emergency contacts associated with the same EmergencyContactsID, which increased the likelihood of data redundancy and anomalies.

To address this, the Emergency Contacts table was created with EmergencyContactsID as the primary key, while the Emergency Relations table was created with a composite primary key consisting of EmergencyContactsID and PatientID.

To further eliminate redundancy and minimize anomalies, all tables were normalized to BCNF. While every table complied with BCNF after 3NF normalization, hypothetical applications of Bernstein's Algorithm and BCNF decomposition were performed to demonstrate an understanding of BCNF normalization.

The actual application of Bernstein's Algorithm in regard to the Billing Information table (Appendix C) was unnecessary because we eliminated the possibility of using the "policy number" attribute as a primary key for the table early in development. Similarly, the hypothetical BCNF decomposition (also in Appendix C) was not required since the "full name" attribute was neither a primary nor a foreign key as the decomposition suggests. This exercise became a useful

frame of reference for the development of complex queries that would rely on joins involving foreign keys.

Final Entity-Relationship Diagram (Assignment 2)

Based on the refinement of the original conceptual design, the Entity-Relationship Diagram for PatientLink is as follows:

Final Design Overview (Assignment 6, 7, 8)

Based on the refinement of the original conceptual design, the tables for the PatientLink database are as follows:

Patient Information

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	PATIENTID		CHAR(16 BYTE)		Yes	(null)		1		(null)
2	FULLNAME		VARCHAR2(50 BYTE)		Yes	(null)		2		(null)
3	HEALTHCARDNUMBER		CHAR(10 BYTE)		Yes	(null)		3		(null)
4	SEX		VARCHAR2(15 BYTE)		Yes	(null)		4		(null)
5	GENDER		VARCHAR2(25 BYTE)		Yes	(null)		5		(null)
6	DATEOFBIRTH		DATE		Yes	(null)		6		(null)
7	DATEOFDEATH		DATE		Yes	(null)		7		(null)
8	BLOODTYPE		VARCHAR2(5 BYTE)		Yes	(null)		8		(null)
9	BILLINGINFORMATIONID		CHAR(10 BYTE)		Yes	(null)		9		(null)
10	EMERGENCYCONTACTSID		CHAR(10 BYTE)		Yes	(null)		10		(null)

BCNF: All columns are determined by PatientID.

PatientID → {FullName, Sex, Gender, DoB, DoD, BloodType, BillingInformationID, EmergencyContactsID, HealthcardNumber}

Patient Contact Information

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	PATIENTID		CHAR(16 BYTE)		Yes	(null)		1		(null)
2	ADDRESS		VARCHAR2(255 BYTE)		Yes	(null)		2		(null)
3	EMAIL		VARCHAR2(50 BYTE)		Yes	(null)		3		(null)
4	PHONENUMBER		NUMBER		Yes	(null)		4		(null)

BCNF: All columns are determined by PatientID.

Functional Dependency: Patient ID → {Address, Email, Phone Number}

Emergency Contact

↕	COLUMN_NAME	↕	DATA_TYPE	↕	NULLABLE	DATA_DEFAULT	↕	COLUMN_ID	↕	COMMENTS
1	EMERGENCYCONTACTSID		CHAR(10 BYTE)		No	(null)		1		(null)
2	FULLNAME		VARCHAR2(50 BYTE)		Yes	(null)		2		(null)
3	PHONENUMBER		NUMBER		Yes	(null)		3		(null)
4	EMAIL		VARCHAR2(50 BYTE)		Yes	(null)		4		(null)
5	SECONDARYPHONENUMBER		NUMBER		Yes	(null)		5		(null)

BCNF: All columns are determined by EmergencyContactsID.

Functional Dependency: EmergencyContactsID → {FullName, PhoneNumber, Email, SecondaryPhoneNumber}

Emergency Relations

↕	COLUMN_NAME	↕	DATA_TYPE	↕	NULLABLE	DATA_DEFAULT	↕	COLUMN_ID	↕	COMMENTS
1	EMERGENCYCONTACTSID		CHAR(10 BYTE)		Yes	(null)		1		(null)
2	PATIENTID		CHAR(16 BYTE)		Yes	(null)		2		(null)
3	RELATION		VARCHAR2(50 BYTE)		Yes	(null)		3		(null)
4	PRIORITY		VARCHAR2(3 BYTE)		Yes	(null)		4		(null)

BCNF: All columns are determined by EmergencyContactsID & PatientID.

Functional Dependency: EmergencyContactsID, PatientID → {Relation, Priority}

Allergies

↕	COLUMN_NAME	↕	DATA_TYPE	↕	NULLABLE	DATA_DEFAULT	↕	COLUMN_ID	↕	COMMENTS
1	ALLERGIESID		CHAR(10 BYTE)		Yes	(null)		1		(null)
2	TYPE		VARCHAR2(255 BYTE)		Yes	(null)		2		(null)
3	SEVERITY		VARCHAR2(50 BYTE)		Yes	(null)		3		(null)
4	ADDITIONALCOMMENTS		VARCHAR2(255 BYTE)		Yes	(null)		4		(null)

BCNF: All columns are determined by AllergiesID.

Functional Dependency: AllergiesID → {Type, Severity, Additional Comments}

Medical Staff

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	ID		CHAR(16 BYTE)		Yes	(null)		1		(null)
2	FULLNAME		VARCHAR2(50 BYTE)		Yes	(null)		2		(null)
3	ROLE		VARCHAR2(50 BYTE)		Yes	(null)		3		(null)

BCNF: All columns are determined by ID.

Functional Dependency: $ID \rightarrow \{FullName, Role\}$

Appointment Information

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	APPOINTMENT_ID		CHAR(10 BYTE)		Yes	(null)		1		(null)
2	PATIENTID		CHAR(16 BYTE)		Yes	(null)		2		(null)
3	DATEOFAPPOINTMENT		DATE		Yes	(null)		3		(null)
4	REASONFORVISIT		VARCHAR2(255 BYTE)		Yes	(null)		4		(null)
5	PHYSICIAN		VARCHAR2(50 BYTE)		Yes	(null)		5		(null)
6	DIAGNOSIS		VARCHAR2(255 BYTE)		Yes	(null)		6		(null)
7	TREATMENTPLAN		VARCHAR2(255 BYTE)		Yes	(null)		7		(null)

BCNF: All values are determined by Appointment_ID.

Functional Dependency: Appointment ID $\rightarrow \{Patient ID, Date of Appointment, Reason for Visit, Physician, Diagnosis, Treatment Plan\}$

Billing Information

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	BILLINGINFORMATIONID		CHAR(10 BYTE)		Yes	(null)		1		(null)
2	ADDRESS		VARCHAR2(255 BYTE)		Yes	(null)		2		(null)
3	CARDNUMBER		CHAR(16 BYTE)		Yes	(null)		3		(null)
4	BILLSID		CHAR(10 BYTE)		Yes	(null)		4		(null)
5	INSURANCECOVERAGEID		CHAR(10 BYTE)		Yes	(null)		5		(null)

BCNF: All values are determined by BillingInformationID.

Functional Dependency: Billing Information ID $\rightarrow \{Address, Card Number, Bills ID, Insurance Coverage ID\}$

Billing History

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	BILLSID		CHAR(10 BYTE)		Yes	(null)		1 (null)		
2	CHARGENAME		VARCHAR2(50 BYTE)		Yes	(null)		2 (null)		
3	DATEOFCHARGE		DATE		Yes	(null)		3 (null)		
4	ORIGINALCHARGE		NUMBER(7,2)		Yes	(null)		4 (null)		
5	OUTSTANDINGPAYMENT		NUMBER(7,2)		Yes	(null)		5 (null)		
6	PAID		NUMBER(7,2)		Yes	(null)		6 (null)		

BCNF: All values are determined by BillsID.

Functional Dependency: Bills ID → {Charge Name, Date of Charge, Original Charge, Outstanding Payment, Paid}

Insurance Information

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	INSURANCECOVERAGEID		CHAR(10 BYTE)		Yes	(null)		1 (null)		
2	INSURANCEPROVIDER		VARCHAR2(50 BYTE)		Yes	(null)		2 (null)		
3	POLICYNUMBER		VARCHAR2(20 BYTE)		Yes	(null)		3 (null)		
4	COVERAGE TYPE		VARCHAR2(100 BYTE)		Yes	(null)		4 (null)		
5	CURRENTCOVERAGE		NUMBER		Yes	(null)		5 (null)		

BCNF: All values are determined by InsuranceCoverageID.

Functional Dependency: Insurance Coverage ID → {Insurance Provider, Policy Number, Coverage Type, Current Coverage}

Medical History

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	PATIENTID		CHAR(16 BYTE)		Yes	(null)		1 (null)		
2	ALLERGIESID		CHAR(10 BYTE)		Yes	(null)		2 (null)		
3	FAMILYHISTORY		VARCHAR2(255 BYTE)		Yes	(null)		3 (null)		
4	PREVIOUSOPERATIONSID		CHAR(10 BYTE)		Yes	(null)		4 (null)		
5	PRESCRIPTIONSID		CHAR(10 BYTE)		Yes	(null)		5 (null)		
6	IMMUNIZATIONSID		CHAR(10 BYTE)		Yes	(null)		6 (null)		
7	COMMENTS		VARCHAR2(255 BYTE)		Yes	(null)		7 (null)		

BCNF: All values determined by PatientID.

Functional Dependency: PatientID → {Allergies ID, Family History, Previous Operations ID, Prescriptions ID, Immunizations ID, Comments}

Previous Operations

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	PREVIOUSOPERATIONSID		CHAR(10 BYTE)		Yes	(null)		1 (null)		
2	OPERATIONDATE		DATE		Yes	(null)		2 (null)		
3	SURGERYTYPE		VARCHAR2(50 BYTE)		Yes	(null)		3 (null)		
4	INVASIVENESS		VARCHAR2(30 BYTE)		Yes	(null)		4 (null)		
5	DURATIONOFSTAY		VARCHAR2(50 BYTE)		Yes	(null)		5 (null)		
6	DATEOFRELEASE		DATE		Yes	(null)		6 (null)		
7	ATTENDINGPHYSICIAN		VARCHAR2(50 BYTE)		Yes	(null)		7 (null)		
8	COMPLICATIONS		VARCHAR2(255 BYTE)		Yes	(null)		8 (null)		
9	NOTES		VARCHAR2(255 BYTE)		Yes	(null)		9 (null)		

BCNF: All values determined by Previous Operations ID.

Functional Dependency: Previous Operations ID → {Operation Date, Surgery Type, Invasiveness, Duration of Stay, Date of Release, Attending Physician, Complications, Notes}

Immunization Information

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	IMMUNIZATIONSID		CHAR(10 BYTE)		Yes	(null)		1 (null)		
2	VACCINENAME		VARCHAR2(255 BYTE)		Yes	(null)		2 (null)		
3	VACCINEMANUFACTURER		VARCHAR2(255 BYTE)		Yes	(null)		3 (null)		
4	DOSAGENUMBER		VARCHAR2(25 BYTE)		Yes	(null)		4 (null)		
5	REMAININGDOSAGES		NUMBER		Yes	(null)		5 (null)		
6	DATEOFDOSAGE		DATE		Yes	(null)		6 (null)		
7	ADMINISTRATIONROUTE		VARCHAR2(50 BYTE)		Yes	(null)		7 (null)		
8	NEXTDOSAGE		DATE		Yes	(null)		8 (null)		

BCNF: All values determined by Immunizations ID.

Functional Dependency: Immunizations ID → {Vaccine Name, Vaccine Manufacturer, Dosage Number, Remaining Dosages, Date of Dosage, Administration Route, Next Dosage}

Prescriptions and Medication

⚡	COLUMN_NAME	⚡	DATA_TYPE	⚡	NULLABLE	DATA_DEFAULT	⚡	COLUMN_ID	⚡	COMMENTS
1	PRESCRIPTIONSID		CHAR(10 BYTE)		Yes	(null)		1		(null)
2	CURRENTLYPRESCRIBED		NUMBER(1,0)		Yes	0		2		(null)
3	MEDICATIONID		CHAR(8 BYTE)		Yes	(null)		3		(null)
4	PRESCRIPTIONNAME		VARCHAR2(100 BYTE)		Yes	(null)		4		(null)
5	DOSAGE		VARCHAR2(50 BYTE)		Yes	(null)		5		(null)
6	FREQUENCY		NUMBER		Yes	(null)		6		(null)
7	DATEPRESCRIBED		DATE		Yes	(null)		7		(null)
8	REFILLS_REMAINING		NUMBER		Yes	(null)		8		(null)
9	PRESCRIBINGPHYSICIAN		VARCHAR2(50 BYTE)		Yes	(null)		9		(null)

BCNF: All values determined by Prescription ID.

Functional Dependency: Prescriptions ID → {Currently Prescribed, Medication ID, Prescription Name, Dosage, Frequency, Date Prescribed, Refills Remaining, Prescribing Physician}

Query and Results Analysis (Assignment 4 and 5)

In order to simplify the use of PatientLink, advanced and regular queries were created to highlight the key functions of the database. These queries were labelled as ‘Default Queries’ that the user could select in the UI to produce tables with specific formats for frequently requested information. The Default Queries are as follows:

Query 1: Allergy Types

⚡	TYPE
1	Carrots
2	Dairy
3	Gluten
4	Grass
5	Shellfish
6	Tree Nuts

SELECT DISTINCT type

FROM Allergies

ORDER BY type;

RA Notation: $\pi_{\text{type}}(\text{allergies})$

Query 2: Subcutaneous Vaccination Types

	VACCINENAME
1	Chickenpox
2	Covid
3	Influenza
4	Tetanus

```
SELECT DISTINCT Ii.VaccineName
FROM Immunization_Information Ii
WHERE Ii.AdministrationRoute = 'Subcutaneous';
```

RA Notation: $\pi_{\text{VaccineName}} (\sigma_{\text{AdministrationRoute} = \text{"Subcutaneous"}} (\rho_{\text{ii}} (\text{immunization_information})))$

Query 3: Previous Operations Count

	TYPE	AMOUNT
1	Non-Invasive	1
2	Invasive	2

```
SELECT Po.Invasiveness AS Type, COUNT(Po.Invasiveness) AS Amount
FROM Previous_Operations Po
GROUP BY Po.Invasiveness;
```

RA Notation: $\pi_{\text{po.invasiveness} \rightarrow \text{Type}, \text{F COUNT}(\text{po.invasiveness} \rightarrow \text{Amount})} (\rho_{\text{po}} (\text{Previous_Operations}))$

Query 4: Appointment Information

	REASONFORVISIT	PHYSICIAN	DATEOFAPPOINTMENT
1	Presented symptoms of a concussion.	Viktoriya Yancy	20-01-02
2	Random stomach issues, unable to pinpoint cause	Viktoriya Yancy	21-07-02
3	Complaint about wrist pain	Viktoriya Yancy	22-01-24
4	Unable to focus in class	Viktoriya Yancy	22-01-24
5	Persistent Headache.	Viktoriya Yancy	23-07-05
6	Tripping on a wall, unable to reach desk.	Viktoriya Yancy	24-08-31

```
SELECT Appointment_Information.reasonforvisit, Appointment_Information.physician,
Appointment_Information.dateofappointment
FROM Appointment_Information
WHERE physician = 'Viktoriya Yancy'
ORDER BY dateofappointment;
```

RA Notation: $\pi_{\text{ReasonForVisit, Physician, DateOfAppointment}} (\sigma_{\text{Physician} = \text{"Viktoriya Yancy"}} (\text{Appointment_Information}))$

Query 5: Billing Information ID

	BILLINGINFORMATIONID	FULLNAME	PATIENTID
1	BI23456789	Felix Nguyen	CD10293847561029
2	BI12345678	Katrina Mei	XY12345678901234
3	BI98765432	Jackie Lam	JL56295638593718
4	BI45678901	Argel Hunos	AH34850278583754
5	BI87654321	Rohan Manoharan	OP56473829105647
6	BI34567890	Mikayla Morrison	KL37465019283746
7	BI76543210	Vanja Dorovic	MN47561029384756
8	BI11223344	Eleanor Tien	EF56473829105647
9	BI99887766	Eric Wong	AB98765432109876
10	BI55667788	Lily Su	GH01928374650192
11	BI10101011	Erkhes Unur	UV37465019283746
12	BI20202030	Cindy Hua	QR01928374650192

```

SELECT Billing_Information.billinginformationid, Patient_Information.FullName,
Patient_Information.PatientID
FROM Billing_Information
INNER JOIN patient_information ON
Billing_Information.BillingInformationID=Patient_Information.BillingInformationID
ORDER BY Patient_Information.dateofbirth;

```

RA Notation: $\pi_{\text{Billing_Information.BillingInformationID, PatientInformation.FullName, PatientInformation.PatientID}}$

$(\sigma_{\text{Billing_Information.BillingInformationID = Patient_Information.BillingInformationID}} (\text{Billing_Information} \times \text{Patient_Information}))$

Query 6: Billing History

	ORIGINALCHARGE	CHARGENAME	ADDRESS	DATEOFCHARGE
1	85950.2	Limb Lenghtening Surgery	188 Queens Quay W, Toronto	24-09-14
2	900	Terrorism	76 Runnymede Ave, Toronto	24-10-01
3	313.3	Blood Work	40 King St W, Toronto	24-05-20
4	225.86	X-Ray Referral	730 Rowntree Dairy Rd, Vaughan	22-04-04
5	200	Victim	81 Bay Street, Toronto	24-10-01
6	120	Lactose Intolerance Therapy	96 Mowat Ave, Toronto	24-09-25
7	69	Contact Therapy	7600 Kennedy Rd, Markham	24-08-31
8	65	Felix Incident	53 Birchmount Rd, Markham	24-10-01
9	40	Appointment Cancellation Fee	1950 Meadowvale Blvd, Mississauga	23-09-11
10	30	Fall	123 Terra Cotta, Brampton	24-10-01
11	18.98	Cough Medication Prescription	36 Regatta Ave, Richmond Hill	23-02-06
12	9.11	Ibuprofen	500 Queen St S, Bolton	24-09-11

```

SELECT billing_history.originalcharge, billing_history.chargename, billing_information.address,
billing_history.dateofcharge
FROM Billing_History
INNER JOIN Billing_Information ON Billing_Information.BillsID=Billing_History.BillsID
ORDER BY billing_history.originalcharge DESC;

```

RA Notation: $\pi_{\text{billing_history.originalcharge, billing_history.chargename, billing_information.address, billing_history.dateofcharge}}$
 $(\sigma_{\text{Billing_Information.BillsID = Billing_History.BillsID}}(\text{Billing_Information} \times \text{Billing_History}))$

Query 7: Emergency Contact Information

	PATIENT	CONTACT	RELATION
1	Argel Hunos	Jasper	Dog
2	Eleanor Tien	Wolf #2	Friend
3	Eric Wong	Confucious	Alter-Ego
4	Erkhes Unur	Keqing	Friend
5	Felix Nguyen	Eric Wolfe	Friend
6	Jackie Lam	Joe Biden	The President
7	Katrina Mei	Liam Beenken	Boyfriend
8	Lily Su	Whispurr	Cat
9	Mikayla Morrison	Ryomen Sukuna	Pookie
10	Rohan Manoharan	Argel Hunos	Spouse
11	Vanja Dorovic	Violet Dorovic	Spouse

```

SELECT patient_information.fullname AS Patient, emergency_contact.fullname AS Contact,
emergency_relations.relation FROM emergency_contact
INNER JOIN patient_information ON
emergency_contact.emergencycontactsid=patient_information.emergencycontactsid
INNER JOIN emergency_relations ON
emergency_contact.emergencycontactsid=emergency_relations.emergencycontactsid
ORDER BY Patient_Information.Fullname;

```

RA Notation: $\pi_{\text{patient_information.fullname} \rightarrow \text{Patient, emergency_contact.fullname} \rightarrow \text{Contact,}}$
 $\text{emergency_relations.relation} \rightarrow \text{relation}(\sigma_{\text{emergency_contact.emergencycontactsid=patient_information.emergencycontactsid,}}$
 $\text{emergency_contact.emergencycontactsid=emergency_relations.emergencycontactsid}(\text{patient_information} \times \text{emergency_contact,}$
 $\text{emergency_relations} \times \text{emergency_contact}))$

Query 8: Patient Contact Information

⚡ FULLNAME	⚡ ADDRESS	⚡ EMAIL
1 Argel Hunos	123 Terra Cotta, Brampton	robloxislife@css.ca
2 Cindy Hua	36 Regatta Ave, Richmond Hill	seaToad4Life@hotmail.com
3 Eleanor Tien	188 Queens Quay W, Toronto	eleadoor@rbcwall.com
4 Eric Wong	81 Bay Street, Toronto	finance@bro.ca
5 Erkhes Unur	76 Runnymede Ave, Toronto	dotasucks@gmail.com
6 Felix Nguyen	730 Rowntree Dairy Rd, Vaughan	cantRead@google.com
7 Jackie Lam	53 Birchmount Rd, Markham	hoi4@life.ca
8 Katrina Mei	40 King St W, Toronto	okmei@amazon.ca
9 Lily Su	96 Mowat Ave, Toronto	silly@tu cows.ca
10 Mikayla Morrison	500 Queen St S, Bolton	mik@suku.ca
11 Rohan Manoharan	7600 Kennedy Rd, Markham	rodog@roskip.com
12 Vanja Dorovic	1950 Meadowvale Blvd, Mississauga	djVanjaaa@icloud.com

```
SELECT Patient_Information.fullname AS Fullname, Patient_Contact_Information.address,  
Patient_Contact_Information.email FROM Patient_Contact_Information  
INNER JOIN Patient_Information ON  
Patient_Contact_Information.PatientID=Patient_Information.PatientID  
ORDER BY Patient_Information.fullname;
```

RA Notation: $\pi_{\text{Patient_Information.fullname} \rightarrow \text{Fullname}, \text{Patient_Contact_Information.address},$

$\text{Patient_Contact_Information.email}(\sigma_{\text{Patient_Contact_Information.PatientID}=\text{Patient_Information.PatientID}}(\text{Patient_Contact_Information}$
 $\times \text{Patient_Information}))$

Query 9: Medical History IDs

⚡ FULLNAME	⚡ PATIENTID	⚡ ALLERGIESID	⚡ PRESCRIPTIONSID
1 Jackie Lam	JL56295638593718	AL23456789	PR98765432
2 Eric Wong	AB98765432109876	AL90123456	PR99887766
3 Erkhes Unur	UV37465019283746	AL13579246	PR10101011
4 Lily Su	GH01928374650192	AL01234567	PR55667788
5 Katrina Mei	XY12345678901234	AL12345678	PR12345678

```
SELECT Patient_Information.Fullname AS Fullname, Medical_History.PatientID,  
Medical_History.AllergiesID, Medical_History.PrescriptionsID FROM Medical_History  
INNER JOIN Patient_Information ON  
Medical_History.PatientID=Patient_Information.PatientID  
WHERE Medical_History.AllergiesID IS NOT NULL AND Medical_History.PrescriptionsID IS  
NOT NULL;
```


RA Notation: $\pi_{\text{Patient_Information.fullname} \rightarrow \text{Fullname}, \text{Medical_History.PatientID}, \text{Medical_History.AllergiesID}, \text{Medical_History.PrescriptionsID}}$
 $(\sigma_{\text{Medical_History.PatientID}=\text{Patient_Information.PatientID}, \text{Medical_History.AllergiesID} \neq \text{NULL} \wedge \text{Medical_History.PrescriptionsID} \neq \text{NULL}}(\text{Medical_History} \times \text{Patient_Information}))$

Query 10: Patient Blood Type

	PATIENT_COUNT	BLOODTYPE
1	1	AB+
2	2	B+
3	1	O-
4	2	O+
5	1	AB-
6	1	A+
7	1	O
8	3	A-

SELECT COUNT(Patient_Information.Bloodtype) AS Patient_Count,
 Patient_Information.Bloodtype
 FROM Patient_Information
 GROUP BY Patient_Information.Bloodtype;

RA Notation: $\pi_{\text{COUNT (Bloodtype)}} \rho_{\text{COUNT}}(\rho_{\text{Patient_Count}}(\text{Patient_Information}))$

Query 11: Medical Staff

	ID	NAME	ROLE
1	MNL7854029134652	Adam Neville	Nurse
2	GFT3290184726581	Jax Alfons	Nurse
3	RJP1092847563987	Kaidi Quincy	Receptionist

SELECT Ms.ID, Ms.Fullname
 AS Name, Ms.Role FROM Medical_Staff Ms
 WHERE Ms.Role = 'Nurse' OR Ms.Role = 'Receptionist'
 ORDER BY Ms.Role, Ms.Fullname;

RA Notation: $\pi_{\text{ID, Fullname}} \rho_{\text{Name, Role}}(\sigma_{\text{Role}='Nurse' \vee \text{Role}='Receptionist'}(\text{Medical_Staff}))$

Query 12: Insurance Information

	NAME	INSURANCEPROVIDER	COVERAGE TYPE	CURRENT COVERAGE
1	Eric Wong	BelairDirect	Point-of-Service Plan	500
2	Argel Hunos	BelairDirect	Preferred Provider Organization	5000
3	Mikayla Morrison	Desjardins	Point-of-Service Plan	1500
4	Katrina Mei	Desjardins	Preferred Provider Organization	2500
5	Rohan Manoharan	Desjardins	Point-of-Service Plan	1750
6	Lily Su	Manulife	Catastrophic Plan	2000
7	Felix Nguyen	Manulife	Health Maintenance Organization	1275
8	Cindy Hua	SunlifeFinancial	Point-of-Service Plan	1750
9	Erkhes Unur	SunlifeFinancial	Catastrophic Plan	1000
10	Jackie Lam	SunlifeFinancial	Catastrophic Plan	2000

```

SELECT Patient_Information.Fullname AS Name, Insurance_Information.InsuranceProvider,
Insurance_Information.CoverageType, Insurance_Information.CurrentCoverage
FROM Insurance_Information
INNER JOIN Billing_Information ON Billing_Information.InsuranceCoverageID =
Insurance_Information.InsuranceCoverageID
INNER JOIN Patient_Information ON Billing_Information.BillingInformationID =
Patient_Information.BillingInformationID
ORDER BY Insurance_Information.InsuranceProvider;

```

RA Notation:

$$\pi_{\text{Patient_Information.Fullname}, \text{Insurance_Information.InsuranceProvider}, \text{Insurance_Information.CoverageType}, \text{Insurance_Information.CurrentCoverage}} \left(\left(\left(\text{Insurance_Information} \bowtie_{\text{Billing_Information.InsuranceCoverageID} = \text{Insurance_Information.InsuranceCoverageID}} \text{Billing_Information} \right) \bowtie_{\text{Billing_Information.BillingInformationID} = \text{Patient_Information.BillingInformationID}} \text{Patient_Information} \right) \right)$$

Query 13: Prescriptions

	NAME	PRESCRIPTION	DOSAGE	DATE PRESCRIBED
1	Eleanor Tien	Vitamin D Supplements	1	24-08-31
2	Jackie Lam	ADHD Pills	2	23-04-11
3	Erkhes Unur	Anti-Depressants	3	22-04-12
4	Eric Wong	Sleeping Pills	2	22-01-29
5	Lily Su	Lactase	1	21-07-02

```

SELECT Pi.Fullname AS Name, Pm.PrescriptionName AS Prescription, Pm.Dosage,
Pm.DatePrescribed FROM Prescriptions_and_Medication Pm
INNER JOIN Medical_History Mh on Pm.PrescriptionsID = Mh.PrescriptionsID
INNER JOIN Patient_Information Pi on Mh.PatientID = Pi.PatientID
WHERE Pm.CurrentlyPrescribed = 1
ORDER BY Pm.DatePrescribed DESC;

```

RA Notation: $\pi_{pi.fullname \rightarrow name, pm.prescriptionname \rightarrow prescription, pm.dosage, pm.dateprescribed} (\sigma_{pm.currentlyprescribed = 1} ((\rho_{pm} (prescriptions_and_medication))) \bowtie_{pm.prescriptionsid = mh.prescriptionsid} (\rho_{mh} (medical_history))) \bowtie_{mh.patientid = pi.patientid} (\rho_{pi} (patient_information))))$

Query 14: Prescriptions and Past Appointments

NAME	PRESCRIPTION	DOSE	DATEPRESCRIBED	DIAGNOSIS	TREATMENTPLAN
1 Eleanor Tien	Vitamin D Supplements	1	24-08-31	Dwarfism	Limb lengthening surgery, calcium & vitamin D supplements
2 Eric Wong	Sleeping Pills	2	22-01-29	Lack of sleep	Sleep at an earlier time, Take melatonin gummies if needed
3 Lily Su	Lactase	1	21-07-02	Lactose Intolerance	Lactaid pills, lactose intolerance therapy

SELECT Pi.Fullname AS Name, Pm.PrescriptionName AS Prescription, Pm.Dosage AS Dose, Pm.DatePrescribed, Ai.Diagnosis, Ai.TreatmentPlan FROM Prescriptions_and_Medication Pm INNER JOIN Medical_History Mh on Pm.PrescriptionsID = Mh.PrescriptionsID INNER JOIN Patient_Information Pi on Mh.PatientID = Pi.PatientID INNER JOIN Appointment_Information Ai on Mh.PatientID = Ai.PatientID WHERE (Pm.CurrentlyPrescribed = 1) AND (Pm.DatePrescribed BETWEEN (Ai.DateOfAppointment - 30) AND (Ai.DateOfAppointment + 30)) ORDER BY Pm.DatePrescribed DESC;

RA Notation: $\pi_{pi.fullname \rightarrow name, pm.prescriptionname \rightarrow prescription, pm.dosage \rightarrow dose, pm.dateprescribed, ai.diagnosis, ai.treatmentplan} (\sigma_{pm.currentlyprescribed = 1 \text{ AND } (ai.dateofappointment - 30 \leq pm.dateprescribed \text{ AND } pm.dateprescribed \leq ai.dateofappointment + 30)} ((\rho_{pm} (prescriptions_and_medication))) \bowtie_{pm.prescriptionsid = mh.prescriptionsid} (\rho_{mh} (medical_history))) \bowtie_{mh.patientid = pi.patientid} (\rho_{pi} (patient_information))) \bowtie_{mh.patientid = ai.patientid} (\rho_{ai} (appointment_information)))$

Query 15: Insurance Count

PATIENTCOUNT	CURRENTCOVERAGE
1	5000
2	2500
3	1275
4	2000
5	1750
6	1500

SELECT COUNT(Insurance_Information.CurrentCoverage) AS PatientCount, Insurance_Information.CurrentCoverage FROM Insurance_Information GROUP BY Insurance_Information.CurrentCoverage HAVING (Insurance_Information.CurrentCoverage) > 1000;

RA Notation: $\pi_{\text{COUNT}(\text{CurrentCoverage}) \rightarrow \text{PatientCount}, \text{insurance_information.CurrentCoverage}} (\sigma_{\text{insurance_information.CurrentCoverage} > 1000} (\text{CurrentCoverage} \text{ F}_{\text{COUNT CurrentCoverage}})(\text{insurance_information}))$

Query 16: Patient Contact Info and Allergies

	⚡ FULLNAME	⚡ ADDRESS	⚡ EMAIL
1	Katrina Mei	40 King St W, Toronto	okmei@amazon.ca
2	Vanja Dorovic	1950 Meadowvale Blvd, Mississauga	djVanjaaa@icloud.com
3	Mikayla Morrison	500 Queen St S, Bolton	mik@suku.ca
4	Lily Su	96 Mowat Ave, Toronto	silly@tucows.ca
5	Jackie Lam	53 Birchmount Rd, Markham	hoi4@life.ca
6	Argel Hunos	123 Terra Cotta, Brampton	robloxislife@css.ca
7	Erkhes Unur	76 Runnymede Ave, Toronto	dotasucks@gmail.com
8	Eric Wong	81 Bay Street, Toronto	finance@bro.ca

```
SELECT Patient_Information.fullname AS Fullname, Patient_Contact_Information.address,
Patient_Contact_Information.email
FROM Patient_Contact_Information
INNER JOIN Patient_Information ON
Patient_Contact_Information.PatientID=Patient_Information.PatientID
WHERE EXISTS
    (SELECT Patient_Information.Fullname AS Fullname, Medical_History.PatientID,
    Medical_History.AllergiesID FROM Medical_History
    WHERE Medical_History.PatientID = Patient_Information.PatientID AND
    Medical_History.AllergiesID IS NOT NULL);
```

RA Notation: $(\pi_{\text{patient_information.fullname} \rightarrow \text{fullname}, \text{patient_contact_information.address}, \text{patient_contact_information.email}} ((\text{patient_contact_information}) \bowtie_{\text{patient_contact_information.patientid} = \text{patient_information.patientid}} (\text{patient_information}))) \cap (\pi_{\text{patient_information.fullname} \rightarrow \text{fullname}, \text{medical_history.patientid}, \text{medical_history.allergiesid}} (\sigma_{\text{medical_history.patientid} = \text{patient_information.patientid AND NOT (medical_history.allergiesid} = \text{NULL})} (\text{medical_history})))$

Query 17: Prescriptions and Previous Operations

⚡	FULLNAME	⚡	HEALTHCARDNUMBER	⚡	SURGERIES_AND_PRESCRIPTIONS
1	Eleanor Tien	0652590051			Limb Lengthening Surgery
2	Eleanor Tien	0652590051			Vitamin D Supplements
3	Eric Wong	9833672318			Sleeping Pills
4	Erkhes Unur	1884229999			Anti-Depressants
5	Erkhes Unur	1884229999			Hand reattachment surgery
6	Jackie Lam	9163678326			ADHD Pills
7	Katrina Mei	7783237266			Azithromycin
8	Lily Su	9866363459			Lactase
9	Rohan Manoharan	4149813197			Muscle Pain Relief
10	Vanja Dorovic	3028099169			Wisdom Tooth Extraction

```

SELECT Patient_Information.fullname AS Fullname, Patient_Information.HealthCardNumber,
Previous_Operations.SurgeryType AS Surgeries_and_Prescriptions FROM Patient_Information
INNER JOIN Medical_History ON Patient_Information.PatientID=Medical_History.PatientID
INNER JOIN Previous_Operations ON
Medical_History.PreviousOperationsID=Previous_Operations.PreviousOperationsID
UNION

```

```

SELECT Patient_Information.fullname AS Fullname,
Patient_Information.HealthCardNumber,
Prescriptions_And_Medication.PrescriptionName
FROM Patient_Information
INNER JOIN Medical_History ON
Patient_Information.PatientID=Medical_History.PatientID
INNER JOIN Prescriptions_And_Medication ON
Medical_History.PrescriptionsID=Prescriptions_And_Medication.PrescriptionsID;

```

RA Notation: $(\pi_{\text{patient_information.fullname} \rightarrow \text{fullname}, \text{patient_information.healthcardnumber}, \text{previous_operations.surgerytype} \rightarrow \text{surgeries_and_prescriptions}} ((\text{patient_information}) \bowtie_{\text{patient_information.patientid} = \text{medical_history.patientid}} (\text{medical_history}) \bowtie_{\text{medical_history.previousoperationsid} = \text{previous_operations.previousoperationsid}} (\text{previous_operations}))) \cup$

$(\pi_{\text{patient_information.fullname} \rightarrow \text{fullname}, \text{patient_information.healthcardnumber}, \text{prescriptions_and_medication.prescriptionname}} ((\text{patient_information}) \bowtie_{\text{patient_information.patientid} = \text{medical_history.patientid}} (\text{medical_history}) \bowtie_{\text{medical_history.prescriptionsid} = \text{prescriptions_and_medication.prescriptionsid}} (\text{prescriptions_and_medication})))$

Query 18: Billing Information and Billing History

	BILLINGINFORMATIONID	FULLNAME	PATIENTID
1	BI10101011	Erkhes Unur	UV37465019283746
2	BI11223344	Eleanor Tien	EF56473829105647
3	BI23456789	Felix Nguyen	CD10293847561029
4	BI34567890	Mikayla Morrison	KL37465019283746
5	BI45678901	Argel Hunos	AH34850278583754
6	BI76543210	Vanja Dorovic	MN47561029384756

```
SELECT Billing_Information.billinginformationid, Patient_Information.FullName,
Patient_Information.PatientID
FROM Billing_Information
INNER JOIN patient_information ON
Billing_Information.BillingInformationID=Patient_Information.BillingInformationID
MINUS
(SELECT Billing_Information.billinginformationid, Patient_Information.FullName,
Patient_Information.PatientID
FROM Billing_Information
INNER JOIN Patient_Information ON
Patient_Information.BillingInformationID=Billing_Information.BillingInformationID
INNER JOIN Billing_History ON Billing_Information.BillsID=Billing_History.BillsID
WHERE Billing_History.OutstandingPayment = 0);
```

RA Notation: $(\pi_{\text{billing_information.billinginformationid, patient_information.fullname, patient_information.patientid}}$
 $((\text{billing_information}) \bowtie_{\text{billing_information.billinginformationid = patient_information.billinginformationid}}$
 $(\text{patient_information}))) - (\pi_{\text{billing_information.billinginformationid, patient_information.fullname, patient_information.patientid}}$
 $(\sigma_{\text{billing_history.outstandingpayment = 0}} ((\text{billing_information}) \bowtie_{\text{patient_information.billinginformationid =}}$
 $\text{billing_information.billinginformationid} (\text{patient_information}) \bowtie_{\text{billing_information.billsid = billing_history.billsid}}$
 $(\text{billing_history}))))$

User-Interface (Assignment 9)

The user interface for PatientLink was developed as a web-based application using PHP, CSS, and HTML. This approach was chosen because our team is experienced in developing web-based applications, enabling us to efficiently design and implement a secure graphical user interface.

Additionally, building a web-based application provided a wide range of customization options, allowing us to create a user-friendly interface tailored to PatientLink's functional and nonfunctional requirements. This approach also ensured accessibility for authorized users across various browsers and operating systems, making the system incredibly versatile and user-centric.

A web-based application also offers significant potential for future enhancements. Features such as modular design and advanced functionalities can be easily incorporated, ensuring that PatientLink remains adaptable overtime. Ultimately, usability, flexibility, and scalability made a web-based approach the optimal choice for PatientLink's graphical user interface.

A PHP-based system was implemented to manage the oracle server login and was integrated into all other PHP modules within the application. These modules included functionalities for creating and dropping database tables, populating tables with default data, generating default queries, and executing custom queries.

Instructions to Access PatientLink

1. **Connect to TMU CS VPN** - Follow the instructions provided [here](#) to establish a secure connection to the VPN.
2. **Access the PatientLink Application** - Open any web browser and navigate to the following URL: <https://webdev.scs.ryerson.ca/~e12wong/patientlink.html>
 - a. Ensure you are connected to the VPN before accessing this webpage.

PatientLink

Tables

Create Tables

Create All Tables

Drop All Tables

Drop All Tables

Records

Create All Records

Populate All Tables

Queries

Default Queries

Select a Query: 1

Query

Custom Query

Enter Custom Query here...

Custom Query

PatientLink User Instructions

1. **Reset the Database** - Start with a clean slate; Drop all existing tables by clicking the Drop All Tables button.

PatientLink

Tables

Create Tables

Create All Tables

Drop All Tables

Drop All Tables

Records

Create All Records

Populate All Tables

Queries

Default Queries

Select a Query: 1

Query

Custom Query

Enter Custom Query here...

Custom Query

successfully connected with oracle database

log:

Table dropped: ALLERGIES

Table dropped: APPOINTMENT_INFORMATION

Table dropped: BILLING_HISTORY

Table dropped: BILLING_INFORMATION

Table dropped: EMERGENCY_CONTACT

Table dropped: EMERGENCY_RELATIONS

Table dropped: IMMUNIZATION_INFORMATION

Table dropped: INSURANCE_INFORMATION

Table dropped: MEDICAL_HISTORY

Table dropped: MEDICAL_STAFF

Table dropped: PATIENT_CONTACT_INFORMATION

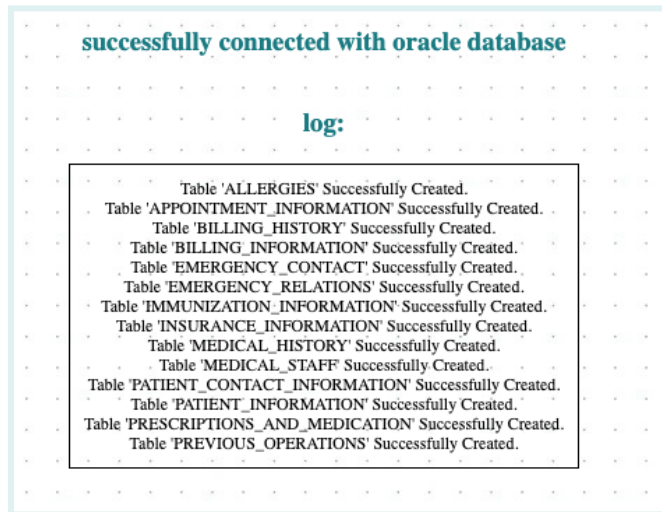
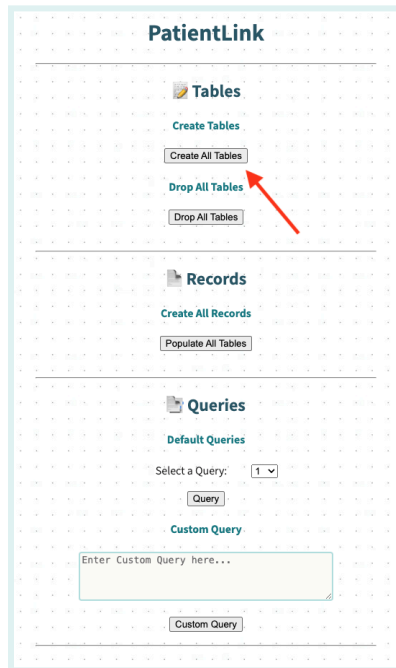
Table dropped: PATIENT_INFORMATION

Table dropped: PRESCRIPTIONS_AND_MEDICATION

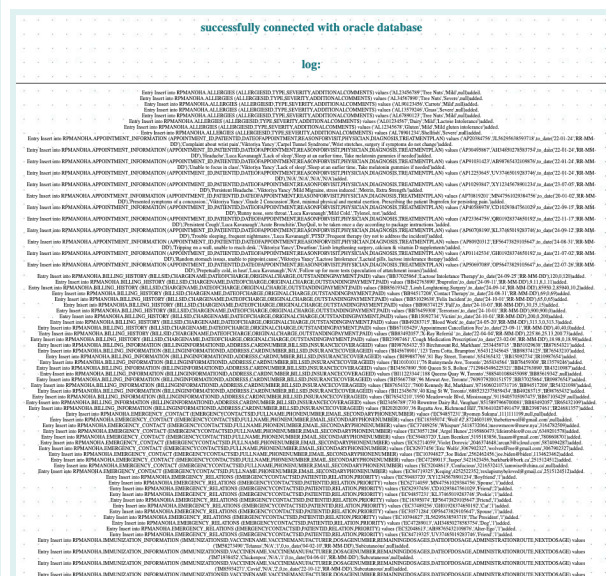
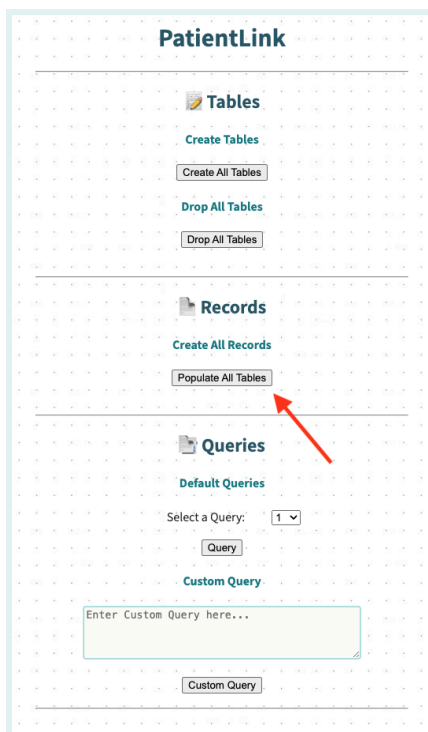
Table dropped: PREVIOUS_OPERATIONS

23

2. Set Up the Database - Create all necessary tables to prepare for data entry by clicking the Create All Tables button.



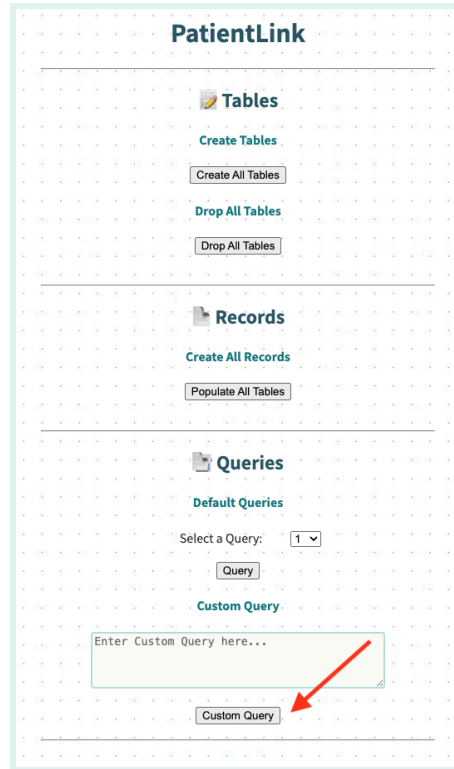
3. Populate Tables with Default Data - Create all default records for all tables by clicking the Populate All Tables button.



4. **Run Default Queries** - Select a default query from the dropdown menu. Click the Query button to display the results in a table format.
 - a. Note: Refer to the 'Query and Results Analysis' section of the documentation for the corresponding query numbers and their functions.



5. **Run Custom Queries** - Enter a custom query in the input field. Click the Custom Query button to generate a table based on the query entered.
 - a. Note: Use this option for any custom table manipulations, including: adding new records, updating existing records, and deleting specific records.
 - b. Note: Queries entered into the input field are case-insensitive. Do not include a semicolon (;) at the end of the query.



Future Developments

In the future, PatientLink will feature role-based access control using secure 16-character alphanumeric IDs to ensure data integrity and confidentiality. Users, including doctors, nurses, medical receptionists, and patients, will have tailored access permissions based on their roles.

Doctors and nurses will manage patient records, including medical history, appointments, prescriptions, and immunizations, while medical receptionists will handle administrative tasks like scheduling and billing. Patients will be able to view their health information and update personal and billing details, enhancing engagement and efficiency.

Key features will include a secure login system that provides role-specific database views and a patient registration module that generates unique IDs for new patients. These enhancements will make PatientLink a reliable and user-friendly healthcare management solution.

Appendix A: Original Conceptual Design (Assignment 1)

Patient Information

Patient ID	Full Name	Health Card #	Sex	Gender	Date of Birth	Date of Death	Blood Type	Billing Information	Emergency Contacts
Alphanumeric String - 16 characters	List of 3 Strings	Alphanumeric String - 10 characters	String	String	Date (MM/DD/YYYY) - All integers	Date (MM/DD/YYYY) - All integers	String	Points to Billing Information Table	Points to Emergency Contact Information table
Static	Dynamic	Static	Static	Dynamic	Static	Static	Static	Dynamic	Dynamic

Patient Contact Information

Patient ID	Address	Email	Phone Number
Alphanumeric String - 16 characters	String - Digit, Address, Street Type	String - Name, "@", Domain	Integer - 10 digits
Static	Dynamic	Dynamic	Dynamic

Emergency Contact Information

Full Name	Relation	Phone #	Secondary Phone #	Email	Priority
List of 3 Strings	String (Selection)	Integer - 10 Digits	Integer - 10 Digits	String - Name, "@", Domain	String (Selection)
Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic

Appointment Information

Patient ID	Date	Reason for Visit	Physician	Diagnosis	Treatment Plan
Alphanumeric String - 16 characters	(MM/DD/YYYY) - All integers	String	String	String	String
Static	Static	Static	Static	Static	Static

Billing Information

Address	Card	Bills	Insurance Coverage
String - Digit, Address, Street Type	Integer - 16 Digits	BooleanPoints to Billing History	Boolean - Points to Insurance and Payment Information Table if True
Dynamic	Dynamic	Dynamic	Dynamic

Billing History

Charge Name	Date of Charge	Original Charge	Outstanding Payment	Paid
String	Date (MM/DD/YYYY) - All integers	Float	Float	Boolean - True if outstanding payment is 0.00
Static	Static	Static	Dynamic	Dynamic

Insurance Information

Insurance Provider	Policy Number	Coverage Type	Current Coverage
String	Integer	String	Float
Dynamic	Dynamic	Dynamic	Dynamic

Medical History

Patient ID	Allergies	Family History/Risks	Previous Operations	Previous/Current Prescriptions	Immunizations	Comments
Alphanumeric String - 16 characters	Boolean - Points to Allergies Table if True	List of strings	Boolean - Points to Previous Operations Table if True	Boolean - Points to Prescriptions and Medication Table if True	Points to Immunization Information Table if True	String
Static	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic

Previous Operations

Date	Surgery Type	Invasive vs Non-Invasive	Duration of Stay	Date of Release	Attending Doctor	Complications	General Notes
(MM/DD/YYYY) - All integers	String	String (Selection)	String	Date (MM/DD/YYYY) - All integers	String	String	String
Static	Static	Static	Static	Static	Static	Static	Static

Immunization Information

Vaccine Name	Vaccine Manufacturer	Dose Number	Remaining Doses	Date of Dose	Administration Site	Next Scheduled Dose
String	String	Integer	Integer	(MM/DD/YYYY) - All integers	String (Selection)	(MM/DD/YYYY) - All integers
Static	Static	Static	Static	Static	Static	Static

Prescriptions and Medication

Currently Prescribed	Medication ID	Prescription Name	Dosage	Frequency	Date Prescribed	Refills Remaining	Prescribing Doctor
Boolean	Alphanumeric String	String	String	String (“Amount/Time Interval”)	Date (MM/DD/YYYY) - All integers	Integer	List of 3 Strings
Dynamic		Static	Static	Static	Static	Dynamic	Static

Allergies

Type	Severity	Additional Comments
String	String (Selection)	String
Dynamic	Dynamic	Dynamic

Medical Staff

ID	Name	Role
Alphanumeric String (16 characters)	List of 3 Strings	String (Selection)
Static	Dynamic	Static

Appendix B: 3NF Justification (Assignment 7)

Patient Information

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 PATIENTID	CHAR(16 BYTE)	Yes	(null)	1 (null)	
2 FULLNAME	VARCHAR2(50 BYTE)	Yes	(null)	2 (null)	
3 HEALTHCARDNUMBER	CHAR(10 BYTE)	Yes	(null)	3 (null)	
4 SEX	VARCHAR2(15 BYTE)	Yes	(null)	4 (null)	
5 GENDER	VARCHAR2(25 BYTE)	Yes	(null)	5 (null)	
6 DATEOFBIRTH	DATE	Yes	(null)	6 (null)	
7 DATEOFDEATH	DATE	Yes	(null)	7 (null)	
8 BLOODTYPE	VARCHAR2(5 BYTE)	Yes	(null)	8 (null)	
9 BILLINGINFORMATIONID	CHAR(10 BYTE)	Yes	(null)	9 (null)	
10 EMERGENCYCONTACTSID	CHAR(10 BYTE)	Yes	(null)	10 (null)	

1NF - All values are atomic

2NF - PatientID → {FullName, Sex, Gender, DoB, DoD, BloodType, BillingInformationID, EmergencyContactsID, HealthcardNumber}

3NF - No transitive dependencies in table

Patient Contact Information

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 PATIENTID	CHAR(16 BYTE)	Yes	(null)	1 (null)	
2 ADDRESS	VARCHAR2(255 B...)	Yes	(null)	2 (null)	
3 EMAIL	VARCHAR2(50 BYTE)	Yes	(null)	3 (null)	
4 PHONENUMBER	NUMBER	Yes	(null)	4 (null)	

1NF - All values are atomic

2NF - Patient ID → {Address, Email, Phone Number}

3NF - No transitive dependencies in table

Emergency Contact Information

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 EMERGENCYCONTACTSID	CHAR(10 BYTE)	Yes	(null)	1 (null)	
2 FULLNAME	VARCHAR2(50 BYTE)	Yes	(null)	2 (null)	
3 RELATION	VARCHAR2(50 BYTE)	Yes	(null)	3 (null)	
4 PHONENUMBER	NUMBER	Yes	(null)	4 (null)	
5 SECONDARYPHONENUMBER	NUMBER	Yes	(null)	5 (null)	
6 EMAIL	VARCHAR2(50 BYTE)	Yes	(null)	6 (null)	
7 PRIORITY	VARCHAR2(3 BYTE)	Yes	(null)	7 (null)	

1NF - All values are atomic

2NF - EmergencyContactsID → {FullName, PhoneNumber, Email}

- There are independent values, decomposition these tables into Emergency Contact and Emergency Relations
- Decomposition into Emergency_Contact & Emergency_Relations

Emergency Contact

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 EMERGENCYCONTACTSID	CHAR(10 BYTE)	No	(null)	1 (null)	
2 FULLNAME	VARCHAR2(50 BYTE)	Yes	(null)	2 (null)	
3 PHONENUMBER	NUMBER	Yes	(null)	3 (null)	
4 EMAIL	VARCHAR2(50 BYTE)	Yes	(null)	4 (null)	

1NF - All values are atomic

2NF - EmergencyContactsID → {FullName, PhoneNumber, Email}

3NF - No transitive dependencies

Emergency Relations

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 EMERGENCYCONTACTSID	CHAR(10 BYTE)	Yes	(null)	1 (null)	
2 PATIENTID	CHAR(16 BYTE)	Yes	(null)	2 (null)	
3 RELATION	VARCHAR2(50 BYTE)	Yes	(null)	3 (null)	
4 SECONDARYPHONENUMBER	NUMBER	Yes	(null)	4 (null)	
5 PRIORITY	VARCHAR2(3 BYTE)	Yes	(null)	5 (null)	

1NF - All values are atomic

2NF - EmergencyContactsID, PatientID → {Relation, SecondaryPhoneNumber, Priority}

3NF - No transitive dependencies

- Partial Key: PatientID → SecondaryPhoneNumber

Move SecondaryPhoneNumber to EmergencyContact table to fix

Allergies

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 ALLERGIESID	CHAR(10 BYTE)	Yes	(null)	1 (null)	
2 TYPE	VARCHAR2(255 B...	Yes	(null)	2 (null)	
3 SEVERITY	VARCHAR2(50 BYTE)	Yes	(null)	3 (null)	
4 ADDITIONALCOMMENTS	VARCHAR2(255 B...	Yes	(null)	4 (null)	

1NF - All values are atomic

2NF - AllergiesID → Type

3NF - No transitive dependencies

Medical Staff

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 ID	CHAR(16 BYTE)	Yes	(null)	1 (null)	
2 FULLNAME	VARCHAR2(50 BYTE)	Yes	(null)	2 (null)	
3 ROLE	VARCHAR2(50 BYTE)	Yes	(null)	3 (null)	

1NF - All values are atomic

2NF - ID → {FullName, Role}

3NF - No transitive dependencies

Appointment Information

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 APPOINTMENT_ID	CHAR(10 BYTE)	Yes	(null)	1 (null)	
2 PATIENTID	CHAR(16 BYTE)	Yes	(null)	2 (null)	
3 DATEOFAPPOINTMENT	DATE	Yes	(null)	3 (null)	
4 REASONFORVISIT	VARCHAR2(255 BYTE)	Yes	(null)	4 (null)	
5 PHYSICIAN	VARCHAR2(50 BYTE)	Yes	(null)	5 (null)	
6 DIAGNOSIS	VARCHAR2(255 BYTE)	Yes	(null)	6 (null)	
7 TREATMENTPLAN	VARCHAR2(255 BYTE)	Yes	(null)	7 (null)	

1NF - All values are atomic

2NF - Appointment ID → {Patient ID, Date of Appointment, Reason for Visit, Physician, Diagnosis, Treatment Plan}

3NF - No transitive dependencies

Billing Information

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	BILLINGINFORMATIONID	CHAR	(10 BYTE)	Yes	(null)			1	(null)	
2	ADDRESS	VARCHAR2	(255 BYTE)	Yes	(null)			2	(null)	
3	CARDNUMBER	CHAR	(16 BYTE)	Yes	(null)			3	(null)	
4	BILLSID	CHAR	(10 BYTE)	Yes	(null)			4	(null)	
5	INSURANCECOVERAGEID	CHAR	(10 BYTE)	Yes	(null)			5	(null)	

1NF - All values are atomic

2NF - Billing Information ID → {Address, Card Number, Bills ID, Insurance Coverage ID}

3NF - No transitive dependencies

Billing History

1	BILLSID	CHAR	(10 BYTE)	Yes	(null)			1	(null)	
2	CHARGENAME	VARCHAR2	(50 BYTE)	Yes	(null)			2	(null)	
3	DATEOFCHARGE	DATE		Yes	(null)			3	(null)	
4	ORIGINALCHARGE	NUMBER	(7, 2)	Yes	(null)			4	(null)	
5	OUTSTANDINGPAYMENT	NUMBER	(7, 2)	Yes	(null)			5	(null)	
6	PAID	NUMBER	(7, 2)	Yes	(null)			6	(null)	

1NF - All values are atomic

2NF - Bills ID → {Charge Name, Date of Charge, Original Charge, Outstanding Payment, Paid}

3NF - No transitive dependencies

Insurance Information

1	INSURANCECOVERAGEID	CHAR	(10 BYTE)	Yes	(null)			1	(null)	
2	INSURANCEPROVIDER	VARCHAR2	(50 BYTE)	Yes	(null)			2	(null)	
3	POLICYNUMBER	VARCHAR2	(20 BYTE)	Yes	(null)			3	(null)	
4	COVERAGETYPE	VARCHAR2	(100 BYTE)	Yes	(null)			4	(null)	
5	CURRENTCOVERAGE	NUMBER		Yes	(null)			5	(null)	

1NF - All values are atomic

2NF - Insurance Coverage ID → {Insurance Provider, Policy Number, Coverage Type, Current Coverage}

3NF - No transitive dependencies

Hypothetical Transitive Dependency:

billingInformation (billingInformationID, address, cardNumber, insuranceCoverageID, policyNumber)

Transitive Dependency - billingInformationID → insuranceCoverageID → policyNumber

Fix - Decompose into two tables:

billingInformation (billingInformationID, address, cardNumber, insuranceCoverageID)

insuranceInformation(insuranceCoverageID, policyNumber)

Medical History

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 PATIENTID	CHAR(16 BYTE)	Yes	(null)	1 (null)	
2 ALLERGIESID	CHAR(10 BYTE)	Yes	(null)	2 (null)	
3 FAMILYHISTORY	VARCHAR2(255 BYTE)	Yes	(null)	3 (null)	
4 PREVIOUSOPERATIONSID	CHAR(10 BYTE)	Yes	(null)	4 (null)	
5 PRESCRIPTIONSID	CHAR(10 BYTE)	Yes	(null)	5 (null)	
6 IMMUNIZATIONSID	CHAR(10 BYTE)	Yes	(null)	6 (null)	
7 COMMENTS	VARCHAR2(255 BYTE)	Yes	(null)	7 (null)	

1NF - All values are atomic

2NF - Patient ID → {Allergies ID, Family History, Previous Operations ID, Prescriptions ID, Immunizations ID, Comments}

3NF - No transitive dependencies in table

Previous Operations

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 PREVIOUSOPERATIONSID	CHAR(10 BYTE)	Yes	(null)	1 (null)	
2 OPERATIONDATE	DATE	Yes	(null)	2 (null)	
3 SURGERYTYPE	VARCHAR2(50 BYTE)	Yes	(null)	3 (null)	
4 INVASIVENESS	VARCHAR2(30 BYTE)	Yes	(null)	4 (null)	
5 DURATIONOFSTAY	VARCHAR2(50 BYTE)	Yes	(null)	5 (null)	
6 DATEOFRELEASE	DATE	Yes	(null)	6 (null)	
7 ATTENDINGPHYSICIAN	VARCHAR2(50 BYTE)	Yes	(null)	7 (null)	
8 COMPLICATIONS	VARCHAR2(255 BYTE)	Yes	(null)	8 (null)	
9 NOTES	VARCHAR2(255 BYTE)	Yes	(null)	9 (null)	

1NF - All values are atomic

2NF - Previous Operations ID → {Operation Date, Surgery Type, Invasiveness, Duration of Stay, Date of Release, Attending Physician, Complications, Notes}

3NF - No transitive dependencies in table

Immunization Information

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 IMMUNIZATIONSID	CHAR(10 BYTE)	Yes	(null)	1 (null)	
2 VACCINENAME	VARCHAR2(255 BYTE)	Yes	(null)	2 (null)	
3 VACCINEMANUFACTURER	VARCHAR2(255 BYTE)	Yes	(null)	3 (null)	
4 DOSAGENUMBER	VARCHAR2(25 BYTE)	Yes	(null)	4 (null)	
5 REMAININGDOSAGES	NUMBER	Yes	(null)	5 (null)	
6 DATEOFDOSAGE	DATE	Yes	(null)	6 (null)	
7 ADMINISTRATIONROUTE	VARCHAR2(50 BYTE)	Yes	(null)	7 (null)	
8 NEXTDOSAGE	DATE	Yes	(null)	8 (null)	

1NF - All values are atomic

2NF - Immunizations ID → {Vaccine Name, Vaccine Manufacturer, Dosage Number, Remaining Dosages, Date of Dosage, Administration Route, Next Dosage}

3NF - No transitive dependencies in table

Prescriptions and Medication

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 PRESCRIPTIONSID	CHAR(10 BYTE)	Yes	(null)	1 (null)	
2 CURRENTLYPRESCRIBED	NUMBER(1,0)	Yes	0	2 (null)	
3 MEDICATIONID	CHAR(8 BYTE)	Yes	(null)	3 (null)	
4 PRESCRIPTIONNAME	VARCHAR2(100 BYTE)	Yes	(null)	4 (null)	
5 DOSAGE	VARCHAR2(50 BYTE)	Yes	(null)	5 (null)	
6 FREQUENCY	NUMBER	Yes	(null)	6 (null)	
7 DATEPRESCRIBED	DATE	Yes	(null)	7 (null)	
8 REFILLS_REMAINING	NUMBER	Yes	(null)	8 (null)	
9 PRESCRIBINGPHYSICIAN	VARCHAR2(50 BYTE)	Yes	(null)	9 (null)	

1NF - All values are atomic

2NF - Prescriptions ID → {Currently Prescribed, Medication ID, Prescription Name, Dosage, Frequency, Date Prescribed, Refills Remaining, Prescribing Physician}

3NF - No transitive dependencies in table

Appendix C: Hypothetical Bernstein's Algorithm Application and Hypotential BCNF Decomposition (Assignment 8)

Hypothetical Bernstein's Algorithm Application:

billingInformation (billingInformationID, address, cardNumber, insuranceCoverageID, policyNumber)

Step 1: Functional Dependencies

billingInformationID → address, cardNumber, insuranceCoverageID

insuranceCoverageID → policyNumber

Step 2a: Identify & Remove Redundancies

FD = {billingInformationID → address, billingInformationID → cardNumber, billingInformationID → insuranceCoverageID, insuranceCoverageID → policyNumber}

billingInformationID → address: billingInformationID+ = {billingInformationID, address} not redundant

billingInformationID → cardNumber: billingInformationID+ = {billingInformationID, cardNumber} not redundant

billingInformationID → insuranceCoverageID: billingInformationID+ = {billingInformationID, insuranceCoverageID} not redundant

insuranceCoverageID → policyNumber: insuranceCoverageID+ = {insuranceCoverageID, policyNumber} not redundant

Step 2a: Identify & Remove Partial Dependencies

No composite key; no potential partial dependencies

Step 3: Identify Keys

billingInformationID: LHS only - part of key

address, cardNumber, policyNumber: RHS only - not part of key

billingInformationID+ = {billingInformationID, address, cardNumber, insuranceCoverageID}

- Not key: missing policyNumber

{billingInformationID, insuranceCoverageID}+ = {billingInformationID, address, cardNumber, insuranceCoverageID, policyNumber}

- Not key: missing policyNumber

Step 4: Make Table

billingInformationID → address, cardNumber, insuranceCoverageID

- R1 (billingInformationID, address, cardNumber, insuranceCoverageID)

insuranceCoverageID → policyNumber

- R2 (insuranceCoverageID, policyNumber)

R1 (billingInformationID, address, cardNumber, insuranceCoverageID)

- with FD: billingInformationID → address, cardNumber, insuranceCoverageID

R2 (insuranceCoverageID, policyNumber)

- with FD: insuranceCoverageID → policyNumber

Hypothetical BCNF Decomposition:

Suppose we have the following variation of Patient Information where the functional dependencies are:

Patient Information

FD = PatientID → {BillingInformationID, EmergencyContactsID, HealthcardNumber}

FullName → {Sex, Gender, DoB, DoD, BloodType}

This violates BCNF, where for each FD $X \rightarrow Y$, X is a super key. In simpler terms, X should determine all of the columns in the rest of the table. As we can see there are two keys determining different sets of the column. To fix this we can decompose the table into 2 different ones.

Patient Health Info

FD = PatientID \rightarrow {BillingInformationID, EmergencyContactsID, HealthcardNumber, FullName}

Patient General Info

FD = FullName \rightarrow {Sex, Gender, DoB, DoD, BloodType}

These two tables now both satisfy BCNF, as they follow the above rule.