A green text on a black background

AI-generated content may be incorrect.

The BLOR Company: Lakeridge Community Health Centre Database Design Report

Group members:

* Brady Cousins
* Lucas Delvoie
* Omair Ahmad
* Robert Macklem

INFT2101 – Database development 1

Professor Kareem Draz

April 7, 2025

Table of Contents

[Introduction 3](#_Toc194938905)

[Case Study Background 3](#_Toc194938906)

[Initial Analysis & Preparation 3](#_Toc194938907)

[New Database System Proposal 3](#_Toc194938908)

[Discussion and Analysis 3](#_Toc194938909)

[Implementation Strategy 3](#_Toc194938910)

[Final Remarks and Conclusion 4](#_Toc194938911)

# Introduction

## Mission Statement:

We are laser-focused on promoting, protecting and improving the lifelong health of everyone in the greater Oshawa area through care, reliance, precision and health equity.

## Who We Are:

The BLOR Company has been turning data management into effective, secure, reliable, and optimized solutions for leading organizations in various industries.

We do a lot more than build and deploy a database in the strict sense: we are the architects who lead you to data-driven success. The basis of our competitive advantage is a commitment to excellence, innovative solutions, and collaborative relationships with clients. Our team is composed of highly professional database experts with extensive experience.

One common driving force unites them: to create solutions that truly matter in patient care.

Our portfolio contains numerous successful projects ranging from small clinics to large networks of hospitals, which serve as a testament to our capability in translating complicated requirements into sophisticated, efficient database systems.

BLOR renders the intricacies of data understandable, informative, and thus adoptable for creative decisions. We manage to do so by focusing on security, ease of user interaction, and transparency while providing results within the pre-accorded timeline.

# Case Study Background

Text

# Initial Analysis & Preparation

## Issues in the Current System

There are at least two clear issues we saw in the current system: batch processing (delay in data synchronization and no access to live data updates) and unincorporated data (such as laboratory results).

In a dynamic and high-pressure environment like a hospital, waiting for systems to update to get actionable data and business intelligence is not acceptable. We set out to change that as our number one goal. The second priority, incorporating missing datapoints, would be essential for centralizing the data in the hospital and making it accessible across formats and platforms, so you find the data you are looking for where you are looking for it.

## Team Formation and Approach

When we formed the BLOR Company, we connected and organized over the course discussion board and in-class. We had an early conversation about our strengths and weaknesses, deciding early Lucas was a star presentation-designer, Omair could handle the scripting, and Robert and Brady we strong in the textual and graphical design elements.

Rather than splitting each deliverable four ways, we played into our strengths. Earlier on, design was needed and Robert and Brady got started (with plenty of contribution from Omair as well!) while Lucas started early planning the presentation. We continued prioritizing each other’s strengths throughout the duration of the course: Lucas’s work culminated in a strong presentation plan, and Omair picked up much of the scripting work for the fourth deliverable. Working together to balance load, we were able to stay ahead of deadlines and complete multiple components asynchronously.

## Initial Redesign Plan

Our initial plan was to create a database that acted as a **hub** and treated the software and business needs at the hospital as the **spokes**. By centralizing the data aspect of the system, all other systems could synchronize reliably and meet the needs of LRCH. We wanted to move laboratory results, treatments, prescriptions and other records into the database and link them to core records such as patients and physicians. This would also enable clearer linking of billing items and cost centres, simplifying the billing and accounting side of the system.

# New Database System Proposal

## Our Data Flow Diagram (DFD)

## Attached is our mapping of LRCH’s current flow of data across the organization. When we designed our DFD, it was important show that **data only flows one way at LCHR**. Our aim was to change that.

## It shows patients and staff having to push data into the system, which pushed data to third-party systems (such as vendor systems), and pushed it through the current system to wait for batch processing. Only then does it come back around to feed into the patients and staff, informing and feeding back into the exist system as a constant flowing loop – always rotating around in one direction.

## The purpose of this design was to highlight the need for change, and for a push-pull of data across the organization to allow for flexibility and patient safety.

## Our 3NF Normalized Relational Schema

Our schema was simplified by our design process: by this point we had identified all the entities in the existing system (including the missing datapoints, like laboratory results). Once converted into tables and fields were identified, creating the associated bridge tables for normalization was a straightforward task. When you inspect our schema in the appendices, you will see care was taken to ensure that normalization would be maintained. We additionally sought feedback from the client, particularly about physician and staff ID data, and updated the schema accordingly to match their business requirements.

## Our Entity Relationship Diagram (ERD)

Schema in-hand, creating the ERD was a simple design task. So when we went to putting the tables and attributes into diagram form we focused on providing a clear visual representation of the relationships: a column of all the actors (people and vendors) in the system in the centre, focusing on patients in the centre of the entire diagram.

We branched off various systems from there: health services to the right and down, vendor details above, addresses and location details left, and below that in the bottom left we clustered billing and accounting details.

This design made a clear, readable, and easily navigable document to build our plan off of going forward.

## Our SQL Scripts

Text

# Discussion and Analysis

## Resolving Current Issues

Our proposal resolves the current issues in the system (batch processing and missing datapoints) by establishing a live database server that can be queried in real-time and by adding missing datapoints to the schema.

Use of SQL (via MySQL or PostgreSQL, for example) can easily be configured as a live server, accessible at anytime over the network. Additionally, you will find in our schema representation of multiple pieces of information that are not currently captured (or not captured well) in the current schema. The laboratory results records are the best example of this.

## Following Best Practices

We followed best practices at each stage of our design. We collaborated frequently, sought feedback, and allowed a waterfall methodology to transpire for our design process. This meant we started small with analysis, then scaled up our design at each subsequent stage for each deliveriable.

# Implementation Strategy

Migrating existing infrastructure to the proposed system, if done carefully, would not result in data loss and should result in minimal downtime, if any.

Since the current system uses batch processing, we would plan to schedule tests and deployments during hours where the system is already vulnerable to downtime due to processing overhead. At these times, interference with LRCH operations would be minimal.

Data loss prevention will be ensured by phasing and auditing of migration activities. We would begin with a test migration, in which the live data (post batch-processing) is untampered, but read-into our system. Afterwards we will perform an audit against your live data to find gaps and losses. We will repeat this process until we have certainty of a lossless migration.

Once a lossless migration has been confirmed, our team will begin testing operational impacts in a development environment. The development environment we build will reflect the live environment you have now, and will ensure we see the challenges and gaps that might occur for a live migration.

Only once a lossless migration is confirmed to have no operational impact to the health and safety of hospital patrons and staff, will we plan and execute a live migration into a new production environment.

# Final Remarks and Conclusion

Going forward with our proposal, the next steps would be be looking to collaborate with LRCH vendor partners to ensure a seamless and successful deployment. It would be after discussions with all stakeholders we could begin to start planning a timeline towards our eventual go-live date.

We at The BLOR Company are extremely grateful for this opportunity to develop and showcase our design expertise, and thank the LRCH and DC teams for all of their support.

# Appendix A: Data Flow Diagram (DFD)

A diagram of a diagram

AI-generated content may be incorrect.

# Appendix B: As-Is Contextual Diagram

A diagram of a system

AI-generated content may be incorrect.

# Appendix C: Requirements Documentation

# Functional Requirements

* The System must send Notifications to physicians
* The System must check the room availability.
* The System must update the room availability.
* The system should check, confirm, and update room availability based on the patient throughput (the patient being assigned and discharging from the room)
* The System must assign room to patients.
* The System must generate and send room utilization report to hospital administrator
* The System must request specific lab tests.
* The System must store patient data with their assigned room.
* The System should collect patient information
* The System must store gathered information in the database
* The System must validate the patient ID
* The System must flag the potential duplicates
* The System must confirm verification and store it in the database
* The System must determine room availability
* The System must Assign Available Room
* The System must Record Room Assignment
* The System must Create New Patient Record
* The System must Update Existing Patient Record
* The System must Communicate Room Assignment
* The System must verify the patient payment information
* The System must receive the payment.
* The System must process the payment information.
* The System must generate a payment confirmation receipt and send an email with payment confirmation details to the patient.
* The System must calculate the medical cost.
* The System must validate the medical cost data.
* The System must generate the bill.
* The System must send payment information to accounting system.
* The System must send receipt information to the accounting system.
* The System must collect all transactions related to patients from central database.
* The System must validate the transaction data.
* The System must create and manage invoices.
* The System must generate a revenue report based on transaction information.
* The System must send revenue report to hospital administrator
* The System must receive the patient’s condition summary.
* The System must request the lab test information.
* The System must conduct the lab test.
* The System must send the lab test reports to the room.
* The System must give prescribed medication to a patient.
* The System must check and update the medicine availability/inventory.
* The System must send reports to the room utilization system.
* The System must accept prescriptions from physician.
* The System must send a medicine report to the room.
* The System must review the treatment.
* The System must generate the medical report.
* The System must send the medical report to the physician system.
* The System must gather the physician information.
* The System must validate the physician information.
* The System should check the physician’s availability.
* The System should update the physician’s status.
* The System must assign physician to a patient.
* The System must receive the physician’s data.
* The System must request the patient’s condition.
* The System must send the patient’s condition information to the pharmacy system.
* The System must Store Patient Data to database
* The System must allow retrieval of data from the database
* The System must keep track of patient data (patients' admission, readmission, discharge, etc.)
* The System will keep track of the doctor’s availability
* The System should data backup and recovery: the system should automatically backup data to a secure location on a predetermined schedule.
* Support the updating and maintenance of patient prescriptions.
* Support the scheduling and tracking of physician appointments.
* Provide an accessible interface and data store for laboratory results.
* Real-time updating and display of patient information.
* Real-time updating and display of physician details and patients.
* Real-time updating and display of room utilization.
* Real-time financial tracking with breakdowns by cost centre.
* Automatically generate daily & weekly reports, as required (Referring Physician Reports, Revenue Reports, Room Utilization Reports, and others)
* Automatically generate patient billing information.

# Non-Functional Requirements

Performance:

* Real-Time Responsiveness - provide information in real-time for critical operations like patient registration and inquiries.
* Query Performance: queries should execute efficiently and return results quickly.
* Transaction Throughput: able to handle a large volume of transactions concurrently, especially during peak hours in a hospital.

Scalability:

* Scalability to 200 Beds: accommodate the planned expansion to 200 beds and the associated increase in patients, physicians, and data volume.
* Scalability for Future Growth: scalable to handle future growth beyond the initial 200 beds as the city and region grow.
* The system architecture should support easy scaling, allowing for future expansion to accommodate additional facilities or departments.
* Horizontal Scalability: Adding more servers (risk of integration problems).
* Vertical Scalability: Upgrading existing servers (risk of losing data).

Availability and Reliability:

* High Availability: ensure continuous operation and minimal downtime.
* Reliability: reliable and stable, minimizing errors and data loss.
* Data Backup and Recovery: protect against data loss due to system failures or disasters.
* Backups will be automated weekly and stored securely with a rollback option for up to three months in case of system errors.
* The system will be backed up every four months.
* The system should remain operational 24/7 to support continuous hospital operations.
* The system interface should comply with accessibility standards (e.g., WCAG 2.1) to support users with disabilities.
* The system must be compatible with future Windows OS updates and maintain support for legacy Windows versions as long as they are commonly used in hospitals.
* ability to run on different platforms or environments

Security:

* Data Security: protect sensitive patient data from unauthorized access, modification, or disclosure. Implement appropriate security measures (access control, encryption, audit trails).
* Role-Based Access Control: ensure that users only have access to the data and functionalities necessary for their roles (physicians, nurses, administrators, etc.).

Usability:

* User-Friendly Interface: intuitive interface for all user groups (physicians, nurses, administrative staff) with varying levels of technical skills.
* Easy to Learn and Use: minimize training time and maximize user adoption.
* Efficient Workflow Support: streamline hospital workflows and processes.
* Provide comprehensive training materials and user manuals for staff to ensure effective system usage

Maintainability:

* Easy to Maintain: designed to be easily maintained and updated.
* Modularity: facilitate easier maintenance and future enhancements.
* Well-Documented: database schema, code, user manuals to aid in maintenance and support.

Data Integrity and Consistency:

* Data Accuracy: minimize data entry errors through validation rules and data constraints.

# Appendix D: 3NF Relational Schema

|  |  |
| --- | --- |
| PERSONS | (**PERSON\_ID, FNAME, LNAME, ADDRESS\_ID 🡪 ADDRESSES**, **PHONE**, EMAIL) |
| ADDRESSES | (**ADDRESS\_ID, STREET\_NUM, STREET\_NAME,** LINETWO, **CITY, POSTAL\_CODE 🡪 POSTAL\_CODES**) |
| PROVINCES | (**PROVINCE\_CODE, PROVINCE\_NAME**) |
| COUNTRIES | (**COUNTRY\_CODE, COUNTRY\_NAME**) |
| POSTAL\_CODES | (**POSTAL\_CODE, PROVINCE\_CODE 🡪 PROVINCES**) |
| COUNTRY\_PROVINCES | (**PROVINCE\_CODE 🡪 PROVINCES, COUNTRY\_CODE 🡪 COUNTRIES**) |
| ROOMS | (**ROOM\_NUM, ROOM\_TYPE**) |
| BEDS | (**BED\_ID**, **ROOM\_NUM, EXTENSION**) |
| PATIENTS | (**PATIENT\_ID, PERSON\_ID, HCN, SEX, ROOM\_NUM 🡪 ROOMS, BED, FINANCIAL\_STATUS**) |
| PHYSICIANS | (**PHYSICIAN\_ID, PERSON\_ID, STAFF\_ID, SPECIALTY**) |
| STAFF | (**STAFF\_ID, PERSON\_ID**) |
| VENDOR | (**VENDOR\_ID, ADDRESS\_ID 🡪 ADDRESSES, PHONE**, EMAIL) |
| VENDOR\_REPS | (**VENDOR\_ID, PERSON\_ID 🡪 PERSONS**) |
| STAYS | (**STAY\_ID**, **PATIENT\_ID 🡪 PATIENTS, BED, DATE\_ADMITTED,** DATE\_DISCHARGED**)** |
| PERSCRIPTIONS | (**PERSCRIPTION\_ID, PHYSICIAN\_ID 🡪 PHYSICIANS, PATIENT\_ID 🡪 PATIENTS, DRUG\_SKU, DOSE, QTY**) |
| TREATMENTS | (**TREAMENT\_ID, PATIENT\_ID 🡪 PATIENTS, PHYSICIAN\_ID 🡪 PHYSICIANS, DETAILS**) |
| TREATMENT\_PERSCRIPTIONS | (**TREATMENT\_ID, PERSCRIPTION\_ID**) |
| APPOINTMENTS | (**PHYSICIAN\_ID, PATIENT\_ID, DATE\_TIME**, NOTES) |
| APPLIED\_TREATMENTS | (**APPOINTMENT\_ID, TREAMENT\_ID**) |
| LABORATORY\_RESULTS | (**RESULTS\_ID, PATIENT\_ID 🡪 PATIENTS, PHYSICIAN\_ID 🡪 PHYSICIANS,** |
| COST\_CENTRES | (**COST\_CENTRE, NAME**) |
| BILLING\_CODE | (**BILLING\_CODE**, **DESCRIPTION, COST\_CENTRE, COST**) |
| INVOICES | (**INVOICE\_NUM**, **PATIENT\_ID 🡪 PATIENTS, INVOICE\_DATE, PAID)** |
| INVOICE\_ITEMS | (**INVOICE\_NUM, BILLING\_CODE**) |

# Appendix E: Entity Relationship Diagram (ERD)

A computer diagram of a computer

AI-generated content may be incorrect.

# Appendix F: SQL Scripts

(insert)

# Appendix G: PowerPoint Slides