PROJECT TECHNICAL REPORT FOR MClinLIS

DBST 651

Professor Elkhodari

August 8, 2023

University of Maryland Global Campus

**Introduction**

This Database Design describes the design and implementation of a database that will be used in a clinical laboratory setting. The database will store information required to perform laboratory testing such as patient information, doctor information, tests ordered and their results, instrument information and reagent information. This LIMS system will be strictly for use by laboratory personnel at Molecular Clinical Laboratory. Any outside use will violate HIPPA laws and regulations. The lab will contract a software engineer to maintain and troubleshoot the database.

**Overview**

Molecular Clinical Laboratory is a new reference lab that is ready to start medical testing for patients. In a CLIA/CAP laboratory, it is mandatory that patient information is kept electronically as well as on the original paper document. To store this data electronically, MClinLIS will be created to hold proper documentation to meet CLIA/CAP standards. As different instruments are added, their system will be connected to MClinLIS via their personal IP address to input test results. Results can also be entered manually in case of connection errors.

**Literature Review**

“Design and implementation of a clinical laboratory information system in a low-resource setting”, a 2019 article published by Timothy M. Mtonga et al. in the *African Journal of Laboratory Medicine*, presents a case where a laboratory in a low-resource area needed a new information system that ran error free to support the laboratory testing process. This included computer-aided ordering of tests, barcode labeling and automated reporting of lab results. The new and improved database system helped medical technicians organize patient specimens to avoid losing specimen viability and keep track of tests that needed to be performed as well as the results once testing was complete. The designers left the system open-ended to allow other databases to be connected as needed. The pilot system was put in place for technicians to use. After feedback from the technicians, developers took their input to add additional features and the new LIS was officially implemented in the hospital laboratory.

During the use of the pilot database, an issue that occurred was the system shut down while technicians were entering information in the system via a tablet. To avoid this, this system can only be used on a desktop computer (no more than five years old) within the lab with updated software.

**Assumptions and Constraints**

**Assumptions**

While technicians will be able to enter an unlimited number of patients, each patient can only be entered once and test orders will be added to their same account if new tests are order at a later date than their initial visit.

**Constraints**

Molecular Clinical Laboratory currently only has a functional molecular department for testing. As the lab grows and gains new departments new test codes and specific departments will need to be added to the database when that time comes to sort orders for technicians so that the tests can be sent to the proper department such as hematology or microbiology.

**Risks**

When and if new departments are added in this laboratory, new entities and/or attributes may need to be added so that proper record keeping and resulting can be accessed.

**Design Decisions**

**Key Factors Influencing Data**

The design of this database began with an ERD model (attached in document). Five entities were created as the base structure of this database: 1) Patient entity which is the parent table and contains all patient information in order to create an order. 2) Doctor entity that orders said tests for patient(s). 3) Test entity will have the information about the test so that it can be performed and will later obtain the result to be released. 4) Instrument entity will receive the test order that is input by the user that will have test codes, patient info and type of specimen. 5) Reagent entity keeps track of reagent dates and amounts for inventory and inspection purposes. IT is in charge of monitoring the system and maintaining its security to keep compliance with the HIPAA laws.

**Functional Design Decisions**

The functional design of this database system will be on a server that allows authorized users to input, updated/edit and distribute information. Due to the labs small size, most information will be entered into the server manually with the exception of test results that will be input automatically from the instrument after it has been signed off by a technician. There will be a separate portal that is connected to the LIMS where doctors can use their specific login to receive the results of the patients they ordered tests for. Information can only be entered or updated by authorized laboratory personnel, doctors will only be able to view patient information, billing and tests results of their patients.

**Database Management System Decisions**

For the initial implementation of MClinLIS, it will hold the information from patients, doctors, instruments, tests and reagents. Separate servers that are connected but not a part of this specific database will be linked to the billing department, doctors’ portals and servers from the laboratory instruments. The broad design of this database system will allow for expansion in the near future if the lab decided to expand and create other testing departments within.

**Security and Privacy Design**

The security and privacy of this database will be the responsibility of the lab’s IT personnel. Lab technicians, lab supervisors and directors will have full access to the system. Accessioners will only have access to input/edit patient information, doctor information and tests ordered. Billing will only have access to what tests were ordered and by who for billing purposes as well as reagents to maintain proper inventory. Doctors will only have access to view patient information and their results from their office. All lab personnel will have a unique login provided by IT that will grant them proper access.

**Performance and Maintenance Decisions**

* To avoid a data overload, only one person will have access to a patient or doctor profile at a time. Once editing or updating is complete and changed in the system, it may be opened by another authorized user. The system will be updated every Sunday at 6:00 am. The lab is closed on the weekend, but due to the time needed for molecular testing it may run into Saturday.
* To reduce issues, once a patients information is input into the system it will produce a barcode containing the patients ID number, first name, last name and date of birth. This barcode can be scanned if the instruments fail to connect to the server properly.
* SLA: on call support from IT should be readily available during the work week.
* MClinLIS will be backed up daily at midnight. If an error were to be present and information of that day was unable to be backed up, lab personnel would have to re-enter information manually due to the low work volume. Adjustments could be made in the future if sample volume significantly increased.
* Patient records will be archived of no new testes has been ordered by the two-year mark of their last test. If an existing patient were to have a new test ordered after that two-year mark, the record would be transferred from the archives back to the main server.

**Statement of Work**

**Overview**

Molecular Clinical Laboratory is a new reference lab that does molecular testing for patients with possible immunodeficiencies and to test patients’ genes for their responses to prescription medications. These tests are known as CGX/Immunodeficiency panels and PGX. As of now they document everything manually, which is not as efficient as using a database. A database system will be created to help organize lab documentation for easier access by all laboratory personnel and better record-keeping.

**Objective**

In a laboratory setting, a database system is referred to as a laboratory information system (LIS). This system has the potential to hold all laboratory documentation from incoming orders, patient samples, patient results, inventory, TNPs and QA/QC.

**Project Scope**

This system is being designed with molecular reference laboratories in mind. The system can be expanded with other entities overtime. Documentation is extremely important in a laboratory in order to maintain CLIA/CAP/COLA accreditation and to ensure patients receive the correct results.

Work within the scope

* Create database that stores inventory information
* Entity-Relationship models
* DDL scripts
* Comprehensive report
* Record log of temperatures needed for inventory storage and inspection

Work outside of scope

* Connection between client and lab for patient information and testing
* Test not performed (TNP) log for tests that cannot have a test performed and reasoning why
* QC/QA log for machine and reagent maintenance
* Billing

**Database Goals and Expectations**

The goal is to make access to laboratory documentation easier for staff and inspection personnel. The start of this database will allow staff to access information about inventory to avoid delayed testing and turn-around time to ensure faster results for clients. Temperature logs will also be added for daily documentation which can be easily accessed and reviewed by laboratory and inspection personnel.

**Database Benefits**

There are many benefits that will come from implementing this database. A lab generally requires a director, supervisor, lab technician/technologists, specimen processors etc. There are multiple hands handling documentation in the lab. This database will prevent documentation from being lost or misplaced and allows easier access to records for all personnel. This in turn will ensure faster processing and turn-around time that will keep business flowing.

**Project Hardware and Software**

*Hardware:* for this system a monitor, computer tower, keyboard, and mouse will be needed to use the database software that will be created.

*Software*: Oracle SQL Developer using Virtual Desktop Access via UMGC, ER Assistant for creating entities, attributes, and their relationships. The software being developed will be named MClinLIS which stands for Molecular Clinical Laboratory Information System.

The software will be able to be used on updated Windows and MacOS software’s.

**SQL Usage**

* Dates for patient date of birth, test ordered, inventory received, expiration and record log temperatures will be input as MM-DD-YYYY
* Amount of inventory will allow up to 3 digits
* SQL statements will be broken up per command to ensure readability
* SQL script comments will begin and end with “\*” or begin with “—”
* A name cannot be an SQL command

**Naming Conventions**

* First and last names must contain letters and no numbers
* Names of reagents and refrigerators/freezers will begin with a letter
* Consecutive symbols are not allowed
* Tables and the columns within cannot contain the same name

**Query Syntax**

* Full length keywords only. No abbreviations
* Spaces to be included around equal(=) signs
* Include new line before using keyword “AND”

**Conclusion**

This database system MClinLIS will be in place to help the laboratory operate with more efficiency and to reduce error. This database can expand on a need be basis. It will be user-friendly and help the business expand over time.

**Requirements Definition Document**

**Entity and Attribute Descriptions**

1. Patient
   1. Patient ID#: The accessioning number assigned to each new patient to comply with HIPPA
   2. First Name: First name of the patient
   3. Last Name: Last name of the patient
   4. DOB: Date of birth of the patient (MM-DD-YYYY)
   5. Sex: To determine if patient is male or female (M or F)
   6. Phone #: Phone number of patient to contact if needed (###-###-####)
   7. Specimen Received: The type of specimen sent to lab to be tested for patient (blood, buccal swab, urine, etc.)
   8. Test Code (FK): tests codes are entered along with other patient information so the employee and instrument can know what tests are needed by each patient.
2. Doctor
   1. NPI#: National provider identifier used to identify healthcare providers for healthcare claims
   2. First Name: First name of the doctor ordering the test
   3. Last Name: Last name of the doctor ordering the test
   4. Doctor’s Office: The name of the practice the doctor collected the specimen
   5. Phone #: Phone number of the doctor
   6. E-mail: E-mail of the doctor
   7. Billing ID: the number associated with the doctor so they can be billed for the tests ordered
3. Test Ordered
   1. Test Code: Unique code assigned to each test/ panel performed in the laboratory
   2. Test Name: Name of the test
   3. Date Test Received: The date the specimen was received by the laboratory
   4. Result of Test: Result of test performed
   5. Test Performed By: Initial of technologist that performed the test
   6. Patient ID# (FK): Test ordered will be associated with a specific patient each time
   7. NPI# (FK): the doctor that ordered test(s) for patient(s) will be associated with each test order in order to be performed
4. Instrument
   1. Instrument Name: Full name of the instrument
   2. Serial #: Serial number of the instrument
   3. Manufacturer: the manufacturer of the instrument
   4. Last Calibrated Date: The date the instrument was last calibrated to make sure it is in compliance with manufacturers recommendations to maintain test accuracy
   5. Tests Performed: The tests performed on each instrument
   6. Reagent Needed: Reagents needed for each instrument to perform test
   7. Test Code (FK): Each test code is associated with only on instrument. To know which instrument to perform test on
5. Reagents
   1. Reference #: Reference number associated with reagent
   2. Lot #: Lot number of reagent
   3. Name: Name of reagent
   4. EXP Date: Expiration date of reagent to prevent using expired reagents
   5. Amount Remaining: The amount of each reagent remaining prevent delayed testing

**Relationship and Cardinality Description**

|  |  |  |
| --- | --- | --- |
| **Relationship** | Cardinality | Rules |
| Doctor to Patient | 1:M | A Doctor can have zero to many patients to order tests for (optional). A patient must have one doctor to order their test (mandatory) |
| Patient to Test Ordered | 1:M | A patient (mandatory) must have one to many tests ordered at once to be input in database (mandatory). |
| Test Ordered to Instrument | M:1 | zero to many tests can be performed on an instrument at once (optional). An instrument must have at least one test to operate.(mandatory) |
| Instrument to Reagent | 1:M | An instrument must have one to many reagents to perform test(s) (mandatory). Reagents can be used on zero to many instruments (optional) |
| Test ordered to Doctor | M:1 | Zero to many test orders (optional) are sent to one doctor (mandatory) |
| Instrument to Test Ordered | 1:M | One instrument (mandatory) will result zero to many tests for tests ordered (optional). |

**Assumptions and Special Considerations**

There will be separate entities handled by HR and billing department that will not be included in this model that will be used for diagnostic testing orders only. The HR department will handle employee information; an employee will be able to initial test results once complete before releasing results to doctors. Billing ID associated with the Doctor table will be sent off to the billing department and handled there which will also provide test codes so proper billing can be done. Also amount remaining, reference # and reagent name on the Reagents table will be accessible to billing so proper inventory can be kept.

Another special condition will be the instrument table. It **will not** contain 5 rows because the lab only has a total of 5 instruments since they only have a molecular department. The machines can only test for genomic variants, STDs and respiratory infections

**Detailed Database Design**

**Entity Relationship Diagram**

**A diagram of a test

Description automatically generated**

**DDL Source Code**

/\* DROP statements to clean up objects from previous run \*/

-- Triggers

DROP TRIGGER TRG\_Doctor;

DROP TRIGGER TRG\_Patient;

--Sequences

DROP SEQUENCE SEQ\_Patient\_patient\_id;

DROP SEQUENCE SEQ\_doctor\_billing\_id;

--Views

DROP VIEW PatientInfo;

DROP VIEW DoctorInfo;

DROP VIEW TestsOrdered;

--Indecies

DROP INDEX idx\_PT\_DOB;

DROP INDEX idx\_DR\_Last\_name;

DROP INDEX IDX\_Tests\_Ordered\_Patient\_ID\_FK;

DROP INDEX IDX\_Reagent\_Reagent\_Name;

DROP INDEX IDX\_Instrument\_Instrument\_Name;

--Tables

DROP TABLE Instrument;

DROP TABLE Reagents;

DROP TABLE Tests\_Ordered;

DROP TABLE Doctor;

DROP TABLE Patient;

CREATE TABLE Patient (

Patient\_ID VARCHAR2(20) NOT NULL,

First\_name VARCHAR2(30) NOT NULL,

Last\_name VARCHAR2(30) NOT NULL,

Dob DATE NOT NULL,

Sex CHAR(1) NOT NULL,

Phone\_Number VARCHAR2 (15) NOT NULL,

Specimen\_Received VARCHAR2 (20) NOT NULL,

CONSTRAINT PK\_Patient\_ID Primary Key(Patient\_ID)

);

CREATE TABLE Doctor (

NPI\_# INTEGER NOT NULL,

First\_name VARCHAR2(30) NOT NULL,

Last\_name VARCHAR2(30) NOT NULL,

Doctors\_Office VARCHAR2(100) NOT NULL,

Phone\_Number VARCHAR2 (15) NOT NULL,

Email VARCHAR2 (75) NOT NULL,

Billing\_ID INTEGER NOT NULL,

CONSTRAINT PK\_NPI\_# Primary Key(NPI\_#)

);

CREATE TABLE Tests\_Ordered(

Test\_Code INTEGER NOT NULL,

Test\_name VARCHAR2(30) NOT NULL,

Date\_Test\_Received DATE NOT NULL,

Result\_of\_Test VARCHAR2(100) NOT NULL,

Test\_Performed\_By VARCHAR2 (5) NOT NULL,

Patient\_ID VARCHAR2(100) NOT NULL,

NPI\_# INTEGER NOT NULL,

CONSTRAINT PK\_Test\_Code Primary Key(Test\_Code),

CONSTRAINT FK\_Tests\_Ordered\_Patient\_ID Foreign Key(Patient\_ID) REFERENCES Patient,

CONSTRAINT FK\_Tests\_Ordered\_NPI\_# Foreign Key(NPI\_#) REFERENCES Doctor

);

CREATE TABLE Reagents(

Reference\_Number INTEGER NOT NULL,

Lot\_Number INTEGER NOT NULL,

Reagent\_Name VARCHAR2(40) NOT NULL,

Expiration\_Date DATE NOT NULL,

Amount\_Remaining VARCHAR(30) NOT NULL,

CONSTRAINT PK\_Reference\_Number Primary Key(Reference\_Number)

);

CREATE TABLE Instrument(

Instrument\_Name VARCHAR2(100) NOT NULL,

Seriel\_Number INTEGER NOT NULL,

Manufacturer VARCHAR2(40) NOT NULL,

Date\_Last\_Calibrated DATE NOT NULL,

Test\_Performed\_By VARCHAR2 (5) NOT NULL,

Patient\_ID VARCHAR2(100) NOT NULL,

Lot\_Number INTEGER NOT NULL,

CONSTRAINT PK\_Seriel\_Number Primary Key(Seriel\_Number),

CONSTRAINT FK\_Insturment\_Patient\_ID Foreign Key(Patient\_ID) REFERENCES Patient,

CONSTRAINT FK\_Instrument\_Lot\_Number Foreign Key(Lot\_Number) REFERENCES Reagents

);

/\* Create indices for natural keys, foreign keys, and frequently-queried columns \*/

-- Patient

-- Natural Keys

CREATE INDEX idx\_PT\_DOB ON Patient (Dob);

-- Doctor

-- Natural Keys

CREATE INDEX idx\_DR\_Last\_name ON Doctor (Last\_name);

-- Tests\_ordered

-- Foreign Keys

CREATE INDEX IDX\_Tests\_Ordered\_Patient\_ID\_FK ON Tests\_Ordered (Patient\_id);

-- Reagents

-- Natural Keys

CREATE INDEX IDX\_Reagent\_Reagent\_Name ON Reagents (Reagent\_Name);

-- Instrument

-- Natural Keys

CREATE INDEX IDX\_Instrument\_Instrument\_Name ON Instrument (Instrument\_Name);

/\* Alter Tables by adding Audit Columns \*/

ALTER TABLE Patient ADD (

created\_by VARCHAR2(40),

date\_created DATE,

modified\_by VARCHAR2(40),

date\_modified DATE );

ALTER TABLE Doctor ADD (

created\_by VARCHAR2(40),

date\_created DATE,

modified\_by VARCHAR2(40),

date\_modified DATE );

ALTER TABLE Tests\_Ordered ADD (

created\_by VARCHAR2(40),

date\_created DATE,

modified\_by VARCHAR2(40),

date\_modified DATE );

ALTER TABLE Reagents ADD (

created\_by VARCHAR2(40),

date\_created DATE,

modified\_by VARCHAR2(40),

date\_modified DATE );

ALTER TABLE Instrument ADD (

created\_by VARCHAR2(40),

date\_created DATE,

modified\_by VARCHAR2(40),

date\_modified DATE );

/\* Create Views \*/

-- Business purpose: The PatientInfo view will be used primarily for rapidly fetching patient information

CREATE OR REPLACE VIEW PatientInfo AS

SELECT Patient\_id, First\_name, last\_name, Dob

FROM Patient;

-- Business purpose: The DoctorInfo view will be used to fetch information about doctors

CREATE OR REPLACE VIEW DoctorInfo AS

SELECT NPI\_#, first\_name, last\_name

FROM Doctor;

-- Business purpose: The TestsOrdered view will be used to populate a list patients that needs said test

CREATE OR REPLACE VIEW TestsOrdered AS

SELECT test\_code, test\_name, Patient\_ID

FROM Tests\_ordered;

/\* Create Sequences \*/

CREATE SEQUENCE SEQ\_Patient\_patient\_id

INCREMENT BY 1

START WITH 1

NOMAXVALUE

MINVALUE 0

NOCACHE;

CREATE SEQUENCE SEQ\_Doctor\_billing\_id

INCREMENT BY 1

START WITH 0

NOMAXVALUE

MINVALUE 0

NOCACHE;

/\* Create Triggers \*/

-- Business purpose: The TRG\_Patient trigger automatically assigns a sequential patient ID to a newly-inserted row in the Patient table

create or replace TRIGGER TRG\_Patient

BEFORE INSERT ON Patient

FOR EACH ROW

BEGIN

:NEW.Patient\_id := SEQ\_PATIENT\_PATIENT\_ID.nextval;

END;

/

-- Business purpose: The TRG\_Doctor trigger automatically assigns a sequential billing ID to a newly-inserted row in the doctor table

CREATE TRIGGER TRG\_Doctor

BEFORE INSERT ON Doctor

FOR EACH ROW

BEGIN

:NEW.Billing\_id := SEQ\_DOCTOR\_BILLING\_ID.nextval;

END;

/

**DML Source Code**

**--**Inserting patient information for 10 different patients

INSERT INTO patient(patient\_id, first\_name, last\_name, dob, sex, phone\_number, specimen\_received)

VALUES(SEQ\_PATIENT\_PATIENT\_ID.nextval, 'Aldair', 'Seac', to\_date('1995-05-15', 'YYYY-MM-DD'), 'M', '3059793585', 'Buccal')

INSERT INTO patient(patient\_id, first\_name, last\_name, dob, sex, phone\_number, specimen\_received)

VALUES(SEQ\_PATIENT\_PATIENT\_ID.nextval, 'Adonis', 'Seac', to\_date('2022-05-15', 'YYYY-MM-DD'), 'M', '3059793585', 'Blood');

INSERT INTO patient(patient\_id, first\_name, last\_name, dob, sex, phone\_number, specimen\_received)

VALUES(SEQ\_PATIENT\_PATIENT\_ID.nextval, 'Chuck', 'Davis', to\_date('1935-12-21', 'YYYY-MM-DD'), 'M', '2405689458', 'Blood');

INSERT INTO patient(patient\_id, first\_name, last\_name, dob, sex, phone\_number, specimen\_received)

VALUES(SEQ\_PATIENT\_PATIENT\_ID.nextval, 'Mike', 'Lows', to\_date('1991-03-25', 'YYYY-MM-DD'), 'M', '2569854857', 'Tissue');

INSERT INTO patient(patient\_id, first\_name, last\_name, dob, sex, phone\_number, specimen\_received)

VALUES(SEQ\_PATIENT\_PATIENT\_ID.nextval, 'Sally', 'Kane', to\_date('1975-05-18', 'YYYY-MM-DD'), 'F', '5248993585', 'Buccal');

INSERT INTO patient(patient\_id, first\_name, last\_name, dob, sex, phone\_number, specimen\_received)

VALUES(SEQ\_PATIENT\_PATIENT\_ID.nextval, 'Mollie', 'Mae', to\_date('1984-08-17', 'YYYY-MM-DD'), 'F', '3058647525', 'Tissue');

INSERT INTO patient(patient\_id, first\_name, last\_name, dob, sex, phone\_number, specimen\_received)

VALUES(SEQ\_PATIENT\_PATIENT\_ID.nextval, 'Dory', 'Haze', to\_date('1957-11-15', 'YYYY-MM-DD'), 'F', '2406793585', 'Buccal');

INSERT INTO patient(patient\_id, first\_name, last\_name, dob, sex, phone\_number, specimen\_received)

VALUES(SEQ\_PATIENT\_PATIENT\_ID.nextval, 'Tyler', 'Carter', to\_date('1973-12-02', 'YYYY-MM-DD'), 'M', '2458793105', 'Buccal');

INSERT INTO patient(patient\_id, first\_name, last\_name, dob, sex, phone\_number, specimen\_received)

VALUES(SEQ\_PATIENT\_PATIENT\_ID.nextval, 'Don', 'Scov', to\_date('1995-05-24', 'YYYY-MM-DD'), 'M', '3052468512', 'Blood');

INSERT INTO patient(patient\_id, first\_name, last\_name, dob, sex, phone\_number, specimen\_received)

VALUES(SEQ\_PATIENT\_PATIENT\_ID.nextval, 'Melvin', 'Joseph', to\_date('1945-02-11', 'YYYY-MM-DD'), 'M', '5689425735', 'Buccal')

;

--Inserting doctors information for 10 different doctors

INSERT INTO doctor(npi\_#, first\_name, last\_name, doctors\_office, phone\_number, email, billing\_id)

VALUES(246859, 'Sara', 'Carpenter', 'Med Pros', '3095689458', 'scarpenter@mp.com', seq\_doctor\_billing\_id.nextval);

INSERT INTO doctor(npi\_#, first\_name, last\_name, doctors\_office, phone\_number, email, billing\_id)

VALUES(257896, 'Carol', 'Weathers', 'Happy Nursing Home', '3094586870', 'cweathers@hnh.com', seq\_doctor\_billing\_id.nextval);

INSERT INTO doctor(npi\_#, first\_name, last\_name, doctors\_office, phone\_number, email, billing\_id)

VALUES(587468, 'Jean', 'Joseph', 'Happy Nursing Home', '3094586870', 'jjoseph@hnh.com', seq\_doctor\_billing\_id.nextval);

INSERT INTO doctor(npi\_#, first\_name, last\_name, doctors\_office, phone\_number, email, billing\_id)

VALUES(358965, 'Francisco', 'Ortiz', 'Flash Care', '3098650023', 'fortiz@fc.com', seq\_doctor\_billing\_id.nextval);

INSERT INTO doctor(npi\_#, first\_name, last\_name, doctors\_office, phone\_number, email, billing\_id)

VALUES(134898, 'Margret', 'Baker', 'Med Pros', '3095689458', 'mbaker@mp.com', seq\_doctor\_billing\_id.nextval);

INSERT INTO doctor(npi\_#, first\_name, last\_name, doctors\_office, phone\_number, email, billing\_id)

VALUES(152894, 'Michael', 'Smith', 'Med Pros', '3095689458', 'msmith@mp.com', seq\_doctor\_billing\_id.nextval);

INSERT INTO doctor(npi\_#, first\_name, last\_name, doctors\_office, phone\_number, email, billing\_id)

VALUES(345897, 'Alan', 'Cole', 'Med Pros', '3095689458', 'acole@mp.com', seq\_doctor\_billing\_id.nextval);

INSERT INTO doctor(npi\_#, first\_name, last\_name, doctors\_office, phone\_number, email, billing\_id)

VALUES(001579, 'Roger', 'Holden', 'Flash Care', '3098650023', 'rholden@fc.com', seq\_doctor\_billing\_id.nextval);

INSERT INTO doctor(npi\_#, first\_name, last\_name, doctors\_office, phone\_number, email, billing\_id)

VALUES(340897, 'Sergio', 'Rodriguez', 'Flash Care', '3098650023', 'srodriguez@fc.com', seq\_doctor\_billing\_id.nextval);

INSERT INTO doctor(npi\_#, first\_name, last\_name, doctors\_office, phone\_number, email, billing\_id)

VALUES(468730, 'Nicole', 'Jackson', 'Flash Care', '3098650023', 'njackson@fc.com', seq\_doctor\_billing\_id.nextval);

--Inserting 10 different tests orders into the system

INSERT INTO tests\_ordered(test\_code, test\_name, date\_test\_received, result\_of\_test, test\_performed\_by, patient\_id, npi\_#)

VALUES(2019, 'COVID', to\_date('2023-08-06', 'YYYY-MM-DD'), 'NEGATIVE', 'AD', '2', 246859);

INSERT INTO tests\_ordered(test\_code, test\_name, date\_test\_received, result\_of\_test, test\_performed\_by, patient\_id, npi\_#)

VALUES(2018, 'COVID', to\_date('2023-08-06', 'YYYY-MM-DD'), 'POSITIVE', 'AD', '10', 246859);

INSERT INTO tests\_ordered(test\_code, test\_name, date\_test\_received, result\_of\_test, test\_performed\_by, patient\_id, npi\_#)

VALUES(2017, 'COVID', to\_date('2023-08-06', 'YYYY-MM-DD'), 'NEGATIVE', 'AD', '14', 246859);

INSERT INTO tests\_ordered(test\_code, test\_name, date\_test\_received, result\_of\_test, test\_performed\_by, patient\_id, npi\_#)

VALUES(069, 'STD Panel', to\_date('2023-08-05', 'YYYY-MM-DD'), 'POSITIVE', 'DS', '6', 1579);

INSERT INTO tests\_ordered(test\_code, test\_name, date\_test\_received, result\_of\_test, test\_performed\_by, patient\_id, npi\_#)

VALUES(609, 'STD Panel', to\_date('2023-08-05', 'YYYY-MM-DD'), 'NEGATIVE', 'DS', '16', 1579);

INSERT INTO tests\_ordered(test\_code, test\_name, date\_test\_received, result\_of\_test, test\_performed\_by, patient\_id, npi\_#)

VALUES(0828, 'CGX', to\_date('2023-08-03', 'YYYY-MM-DD'), 'PATHOGENIC', 'BJ', '20', 587468);

INSERT INTO tests\_ordered(test\_code, test\_name, date\_test\_received, result\_of\_test, test\_performed\_by, patient\_id, npi\_#)

VALUES(8028, 'CGX', to\_date('2023-08-03', 'YYYY-MM-DD'), 'NON PATHOGENIC', 'BJ', '8', 587468);

INSERT INTO tests\_ordered(test\_code, test\_name, date\_test\_received, result\_of\_test, test\_performed\_by, patient\_id, npi\_#)

VALUES(8208, 'CGX', to\_date('2023-08-03', 'YYYY-MM-DD'), 'NON PATHOGENIC', 'BJ', '12', 587468);

INSERT INTO tests\_ordered(test\_code, test\_name, date\_test\_received, result\_of\_test, test\_performed\_by, patient\_id, npi\_#)

VALUES(0515, 'PGX', to\_date('2023-08-04', 'YYYY-MM-DD'), 'NOT DETECTED', 'BJ', '4', 340897);

INSERT INTO tests\_ordered(test\_code, test\_name, date\_test\_received, result\_of\_test, test\_performed\_by, patient\_id, npi\_#)

VALUES(5015, 'PGX', to\_date('2023-08-04', 'YYYY-MM-DD'), 'NOT DETECTED', 'BJ', '18', 340897);

**--**Inserting the reagents that will be kept in inventory in the lab

INSERT INTO reagents(reference\_number, lot\_number, reagent\_name, expiration\_date, amount\_remaining)

VALUES(245, 4865, 'Ethanol', to\_date('2025-06-15', 'YYYY-MM-DD'), 10);

INSERT INTO reagents(reference\_number, lot\_number, reagent\_name, expiration\_date, amount\_remaining)

VALUES(265, 4968, 'Isopropanol', to\_date('2026-08-24', 'YYYY-MM-DD'), 9);

INSERT INTO reagents(reference\_number, lot\_number, reagent\_name, expiration\_date, amount\_remaining)

VALUES(813, 9846, 'Working Solution', to\_date('2024-07-02', 'YYYY-MM-DD'), 3);

INSERT INTO reagents(reference\_number, lot\_number, reagent\_name, expiration\_date, amount\_remaining)

VALUES(454, 4084, 'Buffer', to\_date('2024-11-15', 'YYYY-MM-DD'), 6);

INSERT INTO reagents(reference\_number, lot\_number, reagent\_name, expiration\_date, amount\_remaining)

VALUES(065, 7593, 'PK', to\_date('2026-12-15', 'YYYY-MM-DD'), 3);

INSERT INTO reagents(reference\_number, lot\_number, reagent\_name, expiration\_date, amount\_remaining)

VALUES(496, 0484, 'Binding Beads', to\_date('2023-12-31', 'YYYY-MM-DD'), 12);

INSERT INTO reagents(reference\_number, lot\_number, reagent\_name, expiration\_date, amount\_remaining)

VALUES(068, 3496, 'Bleach', to\_date('2030-06-15', 'YYYY-MM-DD'), 5);

INSERT INTO reagents(reference\_number, lot\_number, reagent\_name, expiration\_date, amount\_remaining)

VALUES(646, 9449, 'Elution', to\_date('2026-03-20', 'YYYY-MM-DD'), 7);

INSERT INTO reagents(reference\_number, lot\_number, reagent\_name, expiration\_date, amount\_remaining)

VALUES(568, 7621, 'Solution Cartridge', to\_date('2024-07-25', 'YYYY-MM-DD'), 8);

INSERT INTO reagents(reference\_number, lot\_number, reagent\_name, expiration\_date, amount\_remaining)

VALUES(001, 0123, 'Molecular Grade Water', to\_date('2028-05-15', 'YYYY-MM-DD'), 20)

--Only 5 instruments since the lab is only molecular

INSERT INTO Instrument(Instrument\_Name, Seriel\_Number, Manufacturer, Date\_Last\_Calibrated, Test\_Performed\_By, Patient\_ID, Lot\_Number)

VALUES('Ion Chef', 5864, 'Thermo Fisher', to\_date('2023-05-15', 'YYYY-MM-DD'), 'AD', '2', '245');

INSERT INTO Instrument(Instrument\_Name, Seriel\_Number, Manufacturer, Date\_Last\_Calibrated, Test\_Performed\_By, Patient\_ID, Lot\_Number)

VALUES('Ion Studio S5', 5879, 'Thermo Fisher', to\_date('2023-05-15', 'YYYY-MM-DD'), 'BJ', '12', '245');

INSERT INTO Instrument(Instrument\_Name, Seriel\_Number, Manufacturer, Date\_Last\_Calibrated, Test\_Performed\_By, Patient\_ID, Lot\_Number)

VALUES('King Fisher',2484, 'Thermo Fisher', to\_date('2023-05-15', 'YYYY-MM-DD'), 'AD', '16', '65');

INSERT INTO Instrument(Instrument\_Name, Seriel\_Number, Manufacturer, Date\_Last\_Calibrated, Test\_Performed\_By, Patient\_ID, Lot\_Number)

VALUES('Quant Studio 12', 2340, 'Thermo Fisher', to\_date('2023-05-15', 'YYYY-MM-DD'), 'BJ', '10', '265');

INSERT INTO Instrument(Instrument\_Name, Seriel\_Number, Manufacturer, Date\_Last\_Calibrated, Test\_Performed\_By, Patient\_ID, Lot\_Number)

VALUES('Qubit Flex', 1860, 'Beckman', to\_date('2023-05-15', 'YYYY-MM-DD'), 'AD', '8', '646');

**QUERY Source Code**

**--Q1 showing all columns and rows from patient table**

SELECT \*

FROM Patient

**--Q2 Showing Patient ID, first name, last name, DOB and specimen type from patient table**

SELECT Patient\_id, first\_name, last\_name, DOB, specimen\_received

FROM Patient

**--Q3 Showing a view, in this case doctors NPI#, first and last name.**

SELECT \* FROM DoctorInfo

**--Q4 Joining two tables Patient and Tests Ordered in full**

SELECT \* FROM Patient

JOIN tests\_ordered ON patient.patient\_id=tests\_ordered.patient\_id

**--Q5 Showing patient id, first name and last name of patient and put in alphabetical order by last name**

SELECT Patient\_ID, First\_name, Last\_name

FROM patient

ORDER BY last\_name

**--Q6 Joining 3 tables. Patient ID and specimen received from patient table. Test result from tests ordered table and billing ID from doctor table**

SELECT patient.Patient\_ID, patient.Specimen\_received, tests\_ordered.result\_of\_test, doctor.billing\_id

FROM patient

JOIN tests\_ordered ON patient.patient\_id=tests\_ordered.patient\_id

JOIN doctor ON doctor.npi\_#=tests\_ordered.npi\_#

**--Q7 Joining 3 tables. Patient ID and specimen received from patient table. Test result from tests ordered table and billing ID from doctor table where the specimen is Buccal**

SELECT patient.Patient\_ID, patient.Specimen\_received, tests\_ordered.result\_of\_test, doctor.billing\_id

FROM patient

JOIN tests\_ordered ON patient.patient\_id=tests\_ordered.patient\_id

JOIN doctor ON doctor.npi\_#=tests\_ordered.npi\_#

WHERE patient.specimen\_received='Buccal'

**--Q8 Grouping patient and tests ordered table by # of specimens received for each category**

SELECT patient.specimen\_received, COUNT(\*) as "SPECIMEN"

FROM patient

JOIN tests\_ordered ON patient.patient\_id=tests\_ordered.patient\_id

GROUP BY patient.specimen\_received

HAVING count(\*)>1

**--Q9 Using keyword IN to show patient id, last name and specimens of Buccal**

SELECT patient\_id, last\_name, specimen\_received

FROM patient

WHERE specimen\_received IN('Buccal')

**--Q10 Showing length of first name in patient table**

SELECT last\_name, length(last\_name)

FROM patient

**--Q11 Deleting a row from tests ordered table**

DELETE FROM tests\_ordered

WHERE patient\_id=6

**--Q12 Updating a specimen received**

UPDATE patient

SET specimen\_received='Blood'

WHERE patient\_ID= '2'

**--Q13 Number of patients that have blood specimen**

SELECT COUNT(\*) AS "Number of Patient"

FROM patient

WHERE specimen\_received='Blood'

**--Q14 Number of patients with same last name**

SELECT COUNT(\*) AS "Number of Patient"

FROM patient

WHERE last\_name='Seac'

**--Q15 Number of doctors that work at a specific office**

SELECT COUNT(\*) AS "Number of Doctors"

FROM doctor

WHERE doctors\_office='Med Pros'

**--Q16 Amount of reagents remaining less than 7**

SELECT Reagent\_name, reference\_number, amount\_remaining

FROM reagents

WHERE (amount\_remaining<7)

**--Q17 Patient information and doctors office**

SELECT patient.first\_name, patient.last\_name, patient.specimen\_received, doctor.npi\_#, doctor.doctors\_office

FROM patient

JOIN tests\_ordered ON patient.patient\_id=tests\_ordered.patient\_id

JOIN doctor ON doctor.npi\_#=tests\_ordered.npi\_#

**--Q18 Show patients with area code (305)**

SELECT first\_name, last\_name, phone\_number

FROM patient

WHERE SUBSTR(phone\_number,1,3)='305'

**DDL Source Code Output**

Trigger TRG\_DOCTOR dropped.

Trigger TRG\_PATIENT dropped.

Sequence SEQ\_PATIENT\_PATIENT\_ID dropped.

Sequence SEQ\_DOCTOR\_BILLING\_ID dropped.

View PATIENTINFO dropped.

View DOCTORINFO dropped.

View TESTSORDERED dropped.

Index IDX\_PT\_DOB dropped.

Index IDX\_DR\_LAST\_NAME dropped.

Index IDX\_TESTS\_ORDERED\_PATIENT\_ID\_FK dropped.

Index IDX\_REAGENT\_REAGENT\_NAME dropped.

Index IDX\_INSTRUMENT\_INSTRUMENT\_NAME dropped.

Table INSTRUMENT dropped.

Table REAGENTS dropped.

Table TESTS\_ORDERED dropped.

Table DOCTOR dropped.

Table PATIENT dropped.

Table PATIENT created.

Table DOCTOR created.

Table TESTS\_ORDERED created.

Table REAGENTS created.

Table INSTRUMENT created.

Index IDX\_PT\_DOB created.

Index IDX\_DR\_LAST\_NAME created.

Index IDX\_TESTS\_ORDERED\_PATIENT\_ID\_FK created.

Index IDX\_REAGENT\_REAGENT\_NAME created.

Index IDX\_INSTRUMENT\_INSTRUMENT\_NAME created.

Table PATIENT altered.

Table DOCTOR altered.

Table TESTS\_ORDERED altered.

Table REAGENTS altered.

Table INSTRUMENT altered.

View PATIENTINFO created.

View DOCTORINFO created.

View TESTSORDERED created.

Sequence SEQ\_PATIENT\_PATIENT\_ID created.

Sequence SEQ\_DOCTOR\_BILLING\_ID created.

Trigger TRG\_PATIENT compiled

Trigger TRG\_DOCTOR compiled

**DML Output**

--Patient Table

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

--Doctor Table

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

--Test Ordered Table

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

--Reagent Table

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

--Instrument Table

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

**QUERY Script Output**

**--Q1**

PATIENT\_ID FIRST\_NAME LAST\_NAME DOB S PHONE\_NUMBER SPECIMEN\_RECEIVED CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI

-------------------- ------------------------------ ------------------------------ --------- - --------------- -------------------- ---------------------------------------- --------- ---------------------------------------- ---------

2 Aldair Seac 15-MAY-95 M 3059793585 Buccal

4 Adonis Seac 15-MAY-22 M 3059793585 Blood

6 Chuck Davis 21-DEC-35 M 2405689458 Blood

8 Mike Lows 25-MAR-91 M 2569854857 Tissue

10 Sally Kane 18-MAY-75 F 5248993585 Buccal

12 Mollie Mae 17-AUG-84 F 3058647525 Tissue

14 Dory Haze 15-NOV-57 F 2406793585 Buccal

16 Tyler Carter 02-DEC-73 M 2458793105 Buccal

18 Don Scov 24-MAY-95 M 3052468512 Blood

20 Melvin Joseph 11-FEB-45 M 5689425735 Buccal

10 rows selected.

--**Q2**

PATIENT\_ID FIRST\_NAME LAST\_NAME DOB SPECIMEN\_RECEIVED

-------------------- ------------------------------ ------------------------------ --------- --------------------

2 Aldair Seac 15-MAY-95 Buccal

4 Adonis Seac 15-MAY-22 Blood

6 Chuck Davis 21-DEC-35 Blood

8 Mike Lows 25-MAR-91 Tissue

10 Sally Kane 18-MAY-75 Buccal

12 Mollie Mae 17-AUG-84 Tissue

14 Dory Haze 15-NOV-57 Buccal

16 Tyler Carter 02-DEC-73 Buccal

18 Don Scov 24-MAY-95 Blood

20 Melvin Joseph 11-FEB-45 Buccal

10 rows selected.

**--Q3**

NPI\_# FIRST\_NAME LAST\_NAME

---------- ------------------------------ ------------------------------

246859 Sara Carpenter

257896 Carol Weathers

587468 Jean Joseph

358965 Francisco Ortiz

134898 Margret Baker

152894 Michael Smith

345897 Alan Cole

1579 Roger Holden

340897 Sergio Rodriguez

468730 Nicole Jackson

10 rows selected.

**--Q4**

PATIENT\_ID FIRST\_NAME LAST\_NAME DOB S PHONE\_NUMBER SPECIMEN\_RECEIVED CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI TEST\_CODE TEST\_NAME DATE\_TEST RESULT\_OF\_TEST TEST\_ PATIENT\_ID NPI\_# CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI

-------------------- ------------------------------ ------------------------------ --------- - --------------- -------------------- ---------------------------------------- --------- ---------------------------------------- --------- ---------- ------------------------------ --------- ---------------------------------------------------------------------------------------------------- ----- ---------------------------------------------------------------------------------------------------- ---------- ---------------------------------------- --------- ---------------------------------------- ---------

2 Aldair Seac 15-MAY-95 M 3059793585 Buccal 2019 COVID 06-AUG-23 NEGATIVE AD 2 246859

4 Adonis Seac 15-MAY-22 M 3059793585 Blood 515 PGX 04-AUG-23 NOT DETECTED BJ 4 340897

6 Chuck Davis 21-DEC-35 M 2405689458 Blood 69 STD Panel 05-AUG-23 POSITIVE DS 6 1579

8 Mike Lows 25-MAR-91 M 2569854857 Tissue 8028 CGX 03-AUG-23 NON PATHOGENIC BJ 8 587468

10 Sally Kane 18-MAY-75 F 5248993585 Buccal 2018 COVID 06-AUG-23 POSITIVE AD 10 246859

12 Mollie Mae 17-AUG-84 F 3058647525 Tissue 8208 CGX 03-AUG-23 NON PATHOGENIC BJ 12 587468

14 Dory Haze 15-NOV-57 F 2406793585 Buccal 2017 COVID 06-AUG-23 NEGATIVE AD 14 246859

16 Tyler Carter 02-DEC-73 M 2458793105 Buccal 609 STD Panel 05-AUG-23 NEGATIVE DS 16 1579

18 Don Scov 24-MAY-95 M 3052468512 Blood 5015 PGX 04-AUG-23 NOT DETECTED BJ 18 340897

20 Melvin Joseph 11-FEB-45 M 5689425735 Buccal 828 CGX 03-AUG-23 PATHOGENIC BJ 20 587468

10 rows selected.

**Q5**

PATIENT\_ID FIRST\_NAME LAST\_NAME

-------------------- ------------------------------ ------------------------------

16 Tyler Carter

6 Chuck Davis

14 Dory Haze

20 Melvin Joseph

10 Sally Kane

8 Mike Lows

12 Mollie Mae

18 Don Scov

4 Adonis Seac

2 Aldair Seac

10 rows selected.

**Q6**

PATIENT\_ID SPECIMEN\_RECEIVED RESULT\_OF\_TEST BILLING\_ID

-------------------- -------------------- ---------------------------------------------------------------------------------------------------- ----------

2 Buccal NEGATIVE 1

10 Buccal POSITIVE 1

14 Buccal NEGATIVE 1

8 Tissue NON PATHOGENIC 5

12 Tissue NON PATHOGENIC 5

20 Buccal PATHOGENIC 5

6 Blood POSITIVE 15

16 Buccal NEGATIVE 15

4 Blood NOT DETECTED 17

18 Blood NOT DETECTED 17

10 rows selected.

**Q7**

PATIENT\_ID SPECIMEN\_RECEIVED RESULT\_OF\_TEST BILLING\_ID

-------------------- -------------------- ---------------------------------------------------------------------------------------------------- ----------

2 Buccal NEGATIVE 1

10 Buccal POSITIVE 1

14 Buccal NEGATIVE 1

20 Buccal PATHOGENIC 5

16 Buccal NEGATIVE 15

**Q8**

SPECIMEN\_RECEIVED SPECIMEN

-------------------- ----------

Blood 3

Buccal 5

Tissue 2

**Q9**

PATIENT\_ID LAST\_NAME SPECIMEN\_RECEIVED

-------------------- ------------------------------ --------------------

2 Seac Buccal

10 Kane Buccal

14 Haze Buccal

16 Carter Buccal

20 Joseph Buccal

**Q10**

LAST\_NAME LENGTH(LAST\_NAME)

------------------------------ -----------------

Seac 4

Seac 4

Davis 5

Lows 4

Kane 4

Mae 3

Haze 4

Carter 6

Scov 4

Joseph 6

10 rows selected.

**Q11**

1 row deleted.

**Q12**

1 rows updated

**Q13**

Number of Patient

-----------------

3

**Q14**

Number of Patient

-----------------

2

**Q15**

Number of Doctors

-----------------

4

**Q16**

REAGENT\_NAME REFERENCE\_NUMBER AMOUNT\_REMAINING

---------------------------------------- ---------------- ------------------------------

Working Solution 813 3

Buffer 454 6

PK 65 3

Bleach 68 5

**Q17**

FIRST\_NAME LAST\_NAME SPECIMEN\_RECEIVED NPI\_# DOCTORS\_OFFICE

------------------------------ ------------------------------ -------------------- ---------- ----------------------------------------------------------------------------------------------------

Aldair Seac Buccal 246859 Med Pros

Sally Kane Buccal 246859 Med Pros

Dory Haze Buccal 246859 Med Pros

Mike Lows Tissue 587468 Happy Nursing Home

Mollie Mae Tissue 587468 Happy Nursing Home

Melvin Joseph Buccal 587468 Happy Nursing Home

Chuck Davis Blood 1579 Flash Care

Tyler Carter Buccal 1579 Flash Care

Adonis Seac Blood 340897 Flash Care

Don Scov Blood 340897 Flash Care

10 rows selected.

**Q18**

FIRST\_NAME LAST\_NAME PHONE\_NUMBER

------------------------------ ------------------------------ ---------------

Aldair Seac 3059793585

Adonis Seac 3059793585

Mollie Mae 3058647525

Don Scov 3052468512

**DATA Administration and Monitoring**

**Roles and Responsibilities**

The IT personnel much be proficient in cybersecurity measures to keep patient information confidential. Laboratory technical supervisor will be responsible for database management within the laboratory.

**Database Management System Configuration**

Database Configuration is outside of the scope of scripting this LIMS.

**Database Support Software**

Oracle SQL Developer

**Security and Privacy**

All patients and doctors have numerical identifiers to protect information. Laboratory personnel do not have access to this information except the technicians/technologists performing the tests. Billing will have access to the doctors information for billing purposes only.

**Data Transfer Requirements**

Database must be ran on Windows 10 or higher. MacOS will be allowed in the future if needed.

**Data Formats**

Files or reports exported from the database will be in pdf format and can be converted to other formats outside of the database.

**Backup and Recovery**

Database system will be backed up everyday at midnight automatically. If an error were to occur before the backup, information entered that day would need to be reentered manually.

**References**

Mtonga, T. M., Choonara, F. E., Espino, J. U., Kachaje, C., Kapundi, K., Mengezi, T. E., Mumba, S. L., & Douglas, G. P. (2019). Design and implementation of a clinical laboratory information system in a low-resource setting. *African journal of laboratory medicine*, *8*(1), 841. <https://doi.org/10.4102/ajlm.v8i1.841>