ITIS 6120

Project 2 Report

Kelsey Kornegay

Will Lapinel

Stephanie Karp

Matt Barksdale

**Submitted: 05/03/2024**

# Introduction:

## Project Overview

The purpose of this project is to demonstrate competence in the science of database design. The project requires selecting a specialty in the medical field and designing a database that is custom tailored to meet the needs of that field. In addition to designing the database, SQL scripts were written to implement the database and populate it with sample data as well as add useful functions and queries. UML and ER diagrams were also drawn to give a visual representation of the database structure.

## Database Design

This database is designed to support the operations and administration of a day-to-day ophthalmology practice. It provides for the management and storage of data regarding patients, healthcare providers, visits, clinical care, and operational management including appointments, room allocation, supplies, and billing. It aims to enhance the efficiency of patient care delivery, support administrative functions, and improve data management practices within the practice.

## Updated Portion

Updates to the database in this version include users with different privileges, an audit table and an example of a trigger that would make entries to the audit table, 5 stored procedures, a Python command-line application interface, 2 views, and 2 indexes.

# Functional Requirements:

There are 12 entities/tables in this database (see create\_tables.sql):

* **Appointments**: This table stores all of the information regarding the appointments of each facility; the id of the appointment, the patient’s id, the doctor/provider’s id, the date of the appointment, and the time of the appointment.
* **Audit Table (NEW)**: This table stores all of the information regarding audits, which is a chronological record of changes made to data within a system. It captures details such as what was changed, when it was changed, who made the change, and the old and new values.
* **Equipment**: This table stores which facility it is in, where it is located inside the facility, when it was purchased, and if it is in working condition.
* **Facilities**: This table stores all of the information regarding eye care clinics within the Springfield area; the facility number, the name, and where it is located. Includes facility id, facility name, and address.
* **Inventory**: This table stores all of the information regarding the inventory of products used by the facilities; its id, its name, how many there are, its reorder status, and when it was last ordered. Includes inventory id, item name, quantity, reorder level, and last ordered date.
* **Invoices**: This table stores all of the information regarding the invoices; its id, its visit id ( later used in medical records), the total fee, the amount the customer needs to pay, the amount the customer has paid, the amount insurance needs to pay and the amount insurance has paid, and the date the invoice was issued.
* **Medical Records**: This table stores all of the information regarding the patient medical records of the facilities; the id of the medial record, the patient’s id, the provider/doctor’s id, the id of the visit, the id of the facility, the date it was discharged, the symptoms the patient is facing, the doctor’s diagnosis, and the doctor’s treatment plan.
* **Patients**: This table stores all of the information regarding the patients table and their personal information; the patient's id, their full name, their date of birth, their address, their gender, the insurance they use, and their phone number.
* **Prescriptions**: This table stores all of the information regarding the prescriptions based on the patients records; their prescription id, their medical record id, which medication they use, the dosage required, and the instructions to take the prescription.
* **Providers**: This table stores all of the information regarding the doctors/providers; the id of the provider, their full name, their specialty, whether it be optometrist or ophthalmologist, their phone number, and the number of the room they use.
* **Staff**: This table stores all of the information regarding the people employed/staff; their id, their full name, their job/role, their phone number and the schedule that they work.
* **Vision Tests**: This table stores all of the information regarding the types of vision tests; the id of the test, the id of the medical record, the type of test done, the date the test was done, and the result of the test.
* **Visits**: This table stores all of the information regarding the patient visits; the id of the patient’s visit, the patient’s id, the id of the doctor used for the visit, which facility it was based on the facility id, the date of the visit, and the time of the visit.

# Entity Relationship:

## ER Diagram (arrows indicate new entities)



## UML Diagram

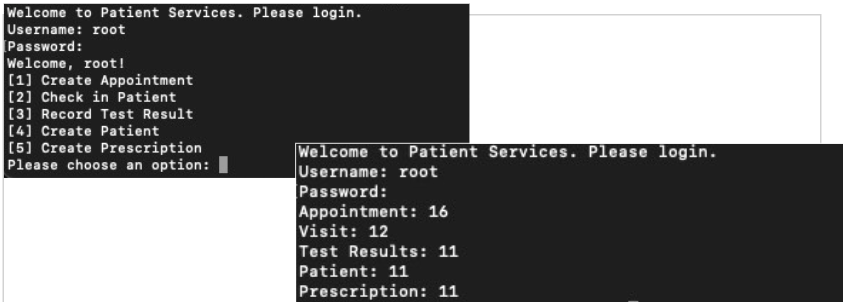


# Proof of BCNF:

Boyce-Codd Normal Form (BCNF) requires that each table has a primary key (or a composite primary key in the case of linked tables), and all attributes are fully functionally dependent on their respective primary key. There are no partial dependencies or transitive dependencies that would violate the rules of BCNF; therefore, all tables are in BCNF. Table Information:

# Application Programming Interface (API) Implementation:

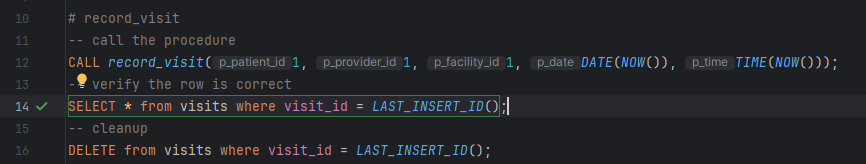
A Python command-line-interface is provided for interacting with the database indirectly. (See api.py)

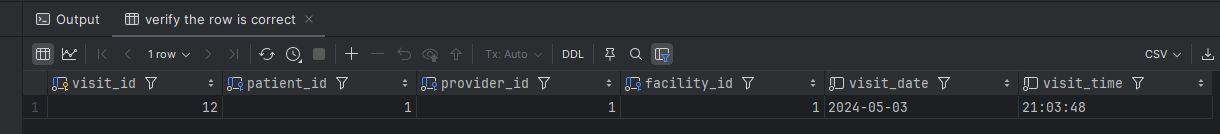


## Stored Procedures

There are five stored procedures (see create\_stored\_procedures.sql):

* **make\_appt**: for making appointments
* **record\_visit**: for recording patient visits
* **record\_test\_results**: for recording results of a vision test or eye exam
* **create\_patient**: for creating a new entry in the patients table
* **create\_prescription**: for creating an entry in the prescriptions table

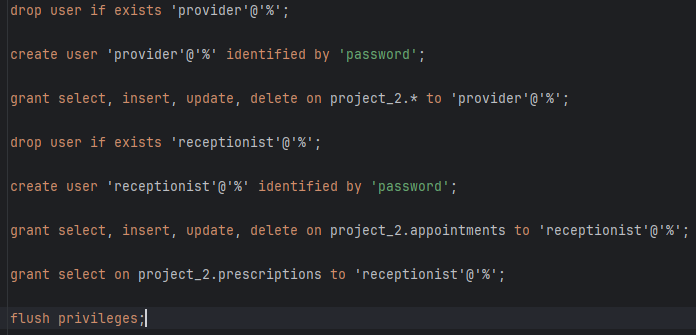




## User Authentication

There are two new users (see create\_users.sql):

* **provider**: more privileged role, with password
* **receptionist**: less privileged role, with password



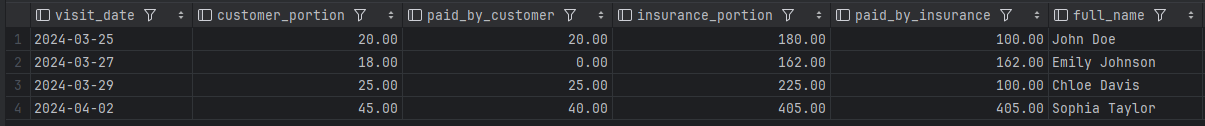
## 

## Views

There are 2 new views (see create\_views.sql):

* **outstanding\_invoices**: shows all unpaid invoices
* **upcoming\_appointments**: shows essential information from various tables for all upcoming appointments

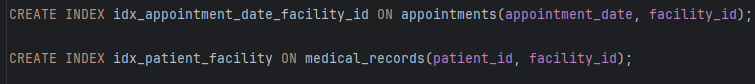




## Indexes

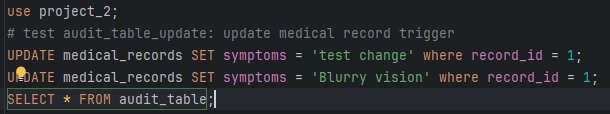
There are 2 new indexes (see create\_indexes.sql):

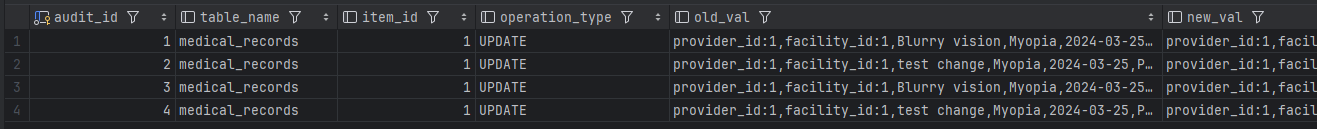
* **idx\_appointment\_date\_facility\_id**: index on date and facililty ID for faster lookup of appointments for a given facility
* **idx\_patient\_facility**: index on patient and facility for faster lookup of records for a given patient and facility



## Audit Trails

There is a trigger example that would be implemented on every SQL command, which creates an entry in the audit table, showing such essential information as the data being changed as well as the user making the changes.





# Tools Used:

* MySQL Workbench
* LucidChart (UML Diagram)
* DataGrip
* Google Docs, Google Slides

# References:

All of the information used in this report came from the instructional materials provided on the ITIS 6120 Spring 2024 Canvas page.