Samuel Baird
Anthony Tatnyo
Project Group: C01

Project Step 4 (Portfolio)

Project title: Sensory Laboratory Information Management Database

Web URL: http://flux.qa.cs.uregina.ca:5111/

Executive Summary:

The Sensory Laboratory Information Management Database project started to improve the management of patient data, testing program, and test results, replacing a cumbersome spreadsheet system. The project underwent several changes from its initial version to its final implementation, driven by feedback received throughout the development process.

Project Scope and Objectives:

The Sensory Laboratory Information Management Database project initial to improve the management of patient data, testing program, and test results, replacing a cumbersome spreadsheet system. The project underwent several changes from its initial version to its final implementation, driven by feedback received throughout the development process.

Project Outcomes:

- Sensory Laboratory Information Management Database, an otherwise identic employee information and an Sensory Lab Test Entry. They supported the creation of employees, also as a primary key for Tests and modifying the ENR_id, the fk fields in entity one primary key and table for convenience, with one attribute, ENR_id in Sensory.
- Three Entity Relationship Diagrams(ERD). The initial data flow diagram functionality in the ERD, the final ERD, and the Sensory Database ERD.
- Removing UNION ALL operator, plus SAC, ICE Report the ordering, combining where filtering and sorting. Implement, we re-structured the Sensory to remove Indent user experience.
- Enhancing Sample ID Purpose, Sample Type, Sample Date, to added Patient and Sample Type information in the Sample ID singleton to provide more context to users when creating a new sample.
- Data Cleaning. We substituted a long-winded date formatting in the Test page. We decided to use the ISO standard date format (YYYY-MM-DD) for consistency.

Project Outline:

Sensory Laboratory tests 20,000 samples for substance abusers per year. Currently, a cumbersome system of spreadsheets is used to record patient data, testing program, and test results. While the laboratory has given its staff the system for many years, the volume of testing has original since the COVID-19 pandemic, and the spreadsheets has become slow and complicated to use. Additionally, a serious mistake occurred at least a set of test results were misplaced and can call off, causing test results to be reported incorrectly to dozens of patients. For these reasons, Laboratory administrators want to develop a more robust system for managing this important patient data.

Laboratory administrators have asked the developers to organize testing data into a database with an administrative web-based interface. This database will keep a record of all Patients and their Samples, and will keep a record of all Employees who have performed Tests on Samples in the Sensory Laboratory Program. The database will also keep a record of Employees and which Tests are assigned to each employee. Tests will be organized by SampleTypes which are related to a set of Reagents used for each type of test. To keep track of reagents, the database will track the units, available for Reagents.

Database Outline:

- **Patients**: records all patients registered in the system so that a result or Tests can be assigned to a specific patient.
  - patient_id: int auto_increment, unique not null PK
  - first_name: varchar not null
  - last_name: varchar not null
  - date_of_birth: date not null
  - email: varchar not null
  - Relationships: 1-M relationship to Samples, with patient_id as a foreign key in Samples
- **Samples**: records all samples collected from Patients.
  - sample_id: int auto_increment, unique not null PK
  - patient_id: int not null FK
  - sample_type: int not null FK
  - sample_date: date not null
  - Relationships:
    - M: 1 relationship to Patients, with patient_id as a foreign key in Samples
    - 1-M relationship to Tests, with sample_id as a foreign key in Tests
- **SampleTypes**: categorizes different types of samples the laboratory is allowed to collect
  - sample_type_id: int auto_increment, unique, not null PK
  - description: varchar, not null
  - Relationship: 1-M relationship to Samples, with sample_type_id as a foreign key in Samples
- **Tests**: records all tests run on Samples and the results.
  - test_id: int auto_increment, unique, not null PK
  - employee_id: int, not null FK
  - sample_id: int, not null FK
  - test_type_id: varchar, not null FK
  - result: varchar
  - test_date: date
  - Relationships:
    - M: 1 relationship to Samples, with sample_id as a foreign key in Tests
    - M: 1 relationship to Employees, with test_id as a foreign key in Tests
    - M: 1 relationship to TestTypes, with test_type_id as a foreign key in Tests
- **Employees**: records all employees conducting testing.
  - employee_id: int auto_increment, unique, not null, PK
  - first_name: varchar, not null
  - last_name: varchar, not null
  - email: varchar, not null
  - Relationships: 1-M relationship to Tests, with employee_id as a foreign key in the Tests table
- **TestTypes**: categorizes Tests by the analyses being tested for and the method used to do the test.
  - test_type_id: varchar, unique, not null PK,
  - analyte: varchar, not null
  - method: varchar, not null
  - Relationships:
    - M:M relationship to Reagents, with test_type_id as a foreign key in the Reagents table
    - 1-M relationship to Tests, with test_type_id as a foreign key in Tests
- **Reagents**: Stores types of reagents and how many available, and how much of each reagent is in inventory.
  - reagent_id: int, unique, auto increment, unique, not null PK,
  - reagent_name: varchar,
  - units_available: int, not null, default 0
  - Relationship: M:M relationship to TestTypes with reagent_id as a foreign key in the TestTypesReagents intersection table.

ER Diagram:

Screen1 updates:

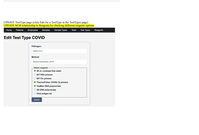
CREATE READ UPDATE DELETE Patients page

UPDATE a Patient page (click Edit on a Patient in the Patients page)

CREATE READ UPDATE DELETE Employees page

UPDATE an Employee page (click Edit on an Employee in the Employees page)

CREATE READ UPDATE DELETE TestTypes page



UPDATE a Sample page (click Edit on a Sample in the Samples page)

CREATE READ UPDATE DELETE TestTypes page

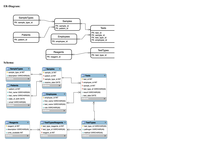
CREATE READ UPDATE DELETE TestTypes page

UPDATE a Test page (click Edit on a Test in the Tests page)

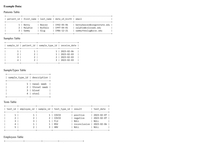
CREATE READ UPDATE DELETE TestTypes page

UPDATE TestTypes page (click Edit for a TestType in the TestTypes page)

CREATE READ UPDATE DELETE TestTypesReagents page



UPDATE TestTypesReagents (click Edit for a Reagent in the Reagents page)



- [x] TaqMan DNA polymerase
- [ ] Q5 DNA polymerase
- [ ] Viral antigen kit

Submit