Histopathology Data Ingestion

Team: Gujjus
Jay Ghevariya (2020101070)
Tirth Motka (2020101036)
Urvish Pujara (2020101032)

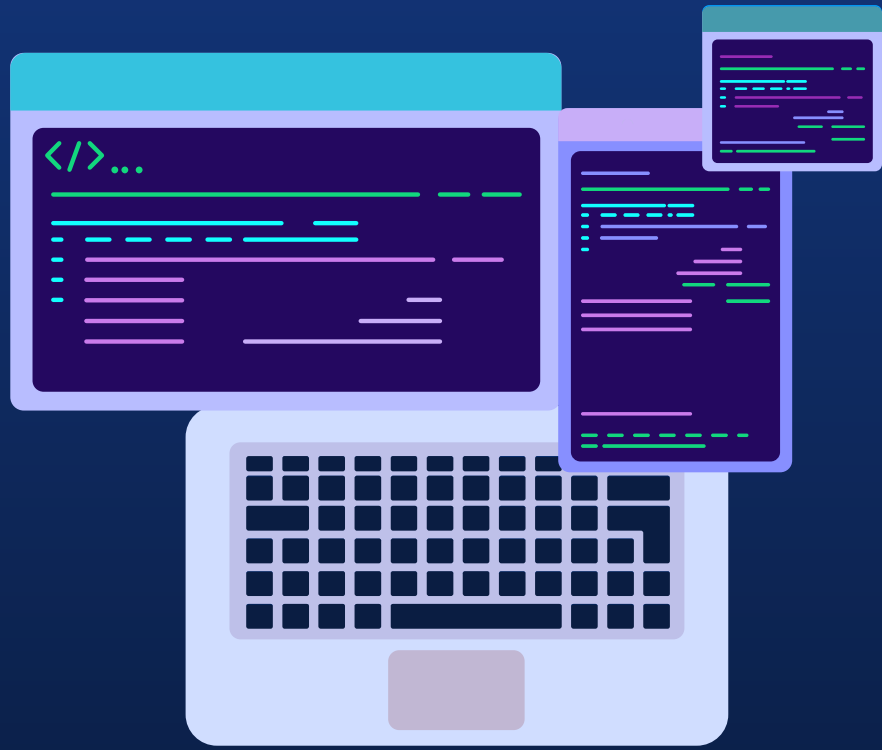


TABLE OF CONTENTS

Overview

- The goal of this project is to create an SQL schema from histopathology data provided in CSV and JSON formats, generate the data and migrate it into SQL tables. The project involves a combination of data analysis, database design, data cleaning, and data migration skills.

Requirements

- **SQL schema generation**: The project has to generate an SQL schema for the data provided in the CSV and JSON files. It will be static, meaning that the schema will not change as the data is updated.
- **Generating data**: The project requires us to generate dummy data for ingestion purpose.
- **Migration of data**: The project has to migrate the cleaned data into the SQL database. This involves inserting the data into the appropriate tables in the database, according to the SQL schema we created earlier.
- **Scalability**: The system should be scalable to accommodate large volumes of data.
- **Error handling**: The system should have mechanisms for handling and reporting errors that occur during data ingestion.
- **Logging**: The system should have mechanisms for logging the process of data ingestion, including any errors that may have occurred.

Non-Functional Requirements

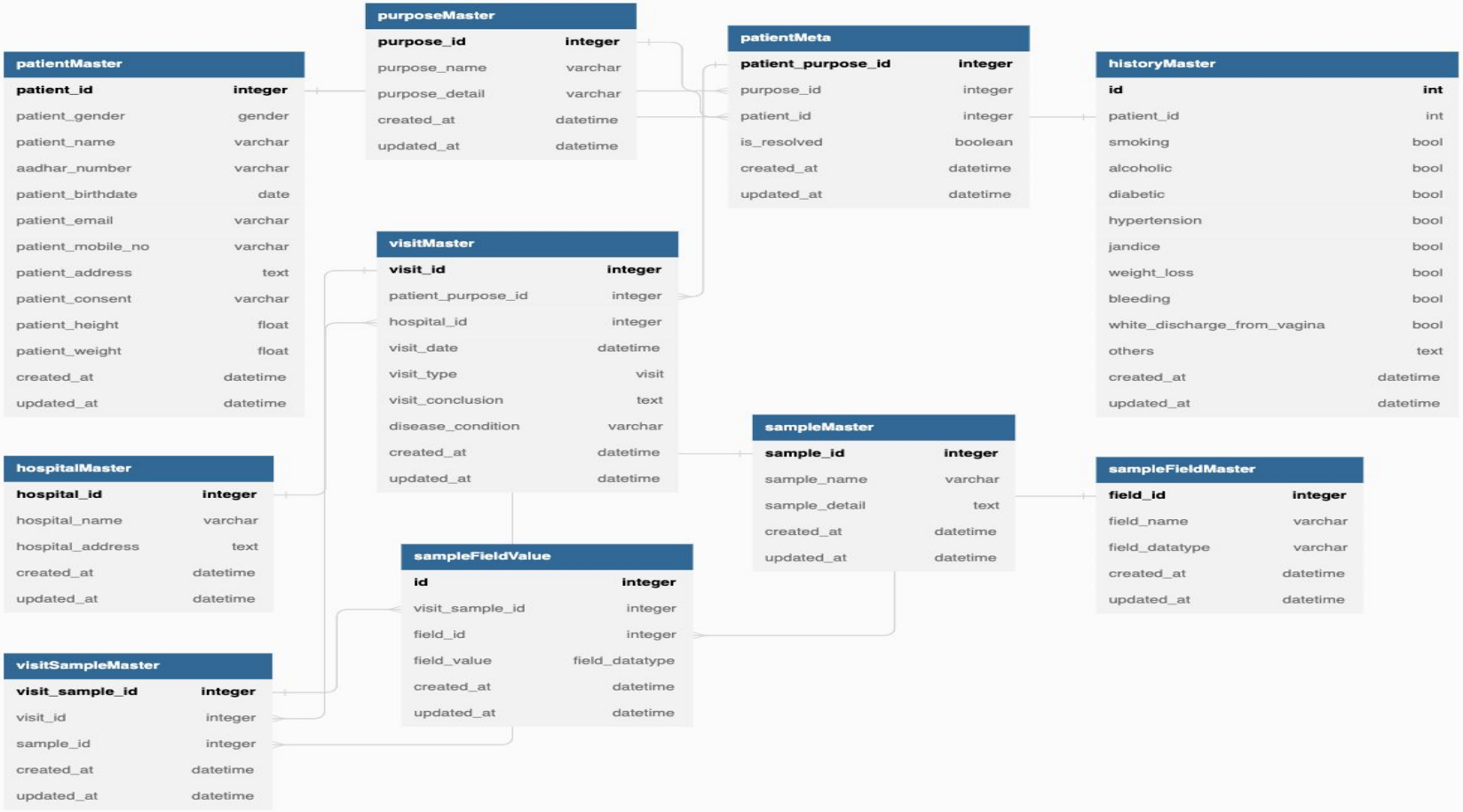
- **Scalability**: The system should be able to handle large volumes of data and scale horizontally as the data volumes grow.
- **Performance**: The system must be able to process and load data quickly and efficiently to minimize processing delays and ensure timely data delivery.
- **Usability**: The system should be user-friendly and easy to use, with clear documentation and error messages to help users troubleshoot issues.
- **Security**: We have implemented a security system to not let a sensitive data get compromised as per user requirements.

Technologies Used

- Javascript (Data Ingestion Scripts)
  - Node.js
  - ESLint (maintaining the code standard)
- MySQL (Database)
  - Sequelize (connect to a database and perform operations)

Implementation

- SQL Generic Schema Generation
  https://flaxen-mayonnaise-bdd.notion.site/Healthcare-Generic-Database-e6712ca6629045faa61c3a8fcf4adbd6
- Table Migration for MySQL Database
- Data Ingestion Scripts
  https://github.com/Patel-Technologies/Gujjus


purposeMaster
purpose_id integer
purpose_name varchar
purpose_detail varchar
created_at datetime
updated_at datetime
patientMaster
patient_id integer
patient_gender gender
patient_name varchar
aadhar_number varchar
patient_birthdate date
patient_email varchar
patient_mobile_no varchar
patient_address text
patient_consent varchar
patient_height float
patient_weight float
created_at datetime
updated_at datetime
patientMeta
patient_purpose_id integer
purpose_id integer
patient_id integer
is_resolved boolean
created_at datetime
updated_at datetime
historyMaster
id int
patient_id int
smoking bool
alcoholic bool
diabetic bool
hypertension bool
jandice bool
weight_loss bool
bleeding bool
white_discharge_from_vagina bool
others text
created_at datetime
updated_at datetime
visitMaster
visit_id integer
patient_purpose_id integer
hospital_id integer
visit_date datetime
visit_type visit
visit_conclusion text
disease_condition varchar
created_at datetime
updated_at datetime
sampleMaster
sample_id integer
sample_name varchar
sample_detail text
created_at datetime
updated_at datetime
sampleFieldMaster
field_id integer
field_name varchar
field_datatype varchar
created_at datetime
updated_at datetime
hospitalMaster
hospital_id integer
hospital_name varchar
hospital_address text
created_at datetime
updated_at datetime
sampleFieldValue
id integer
visit_sample_id integer
field_id integer
field_value field_datatype
created_at datetime
updated_at datetime
visitSampleMaster
visit_sample_id integer
visit_id integer
sample_id integer
created_at datetime
updated_at datetime

Code Structure and Refactoring

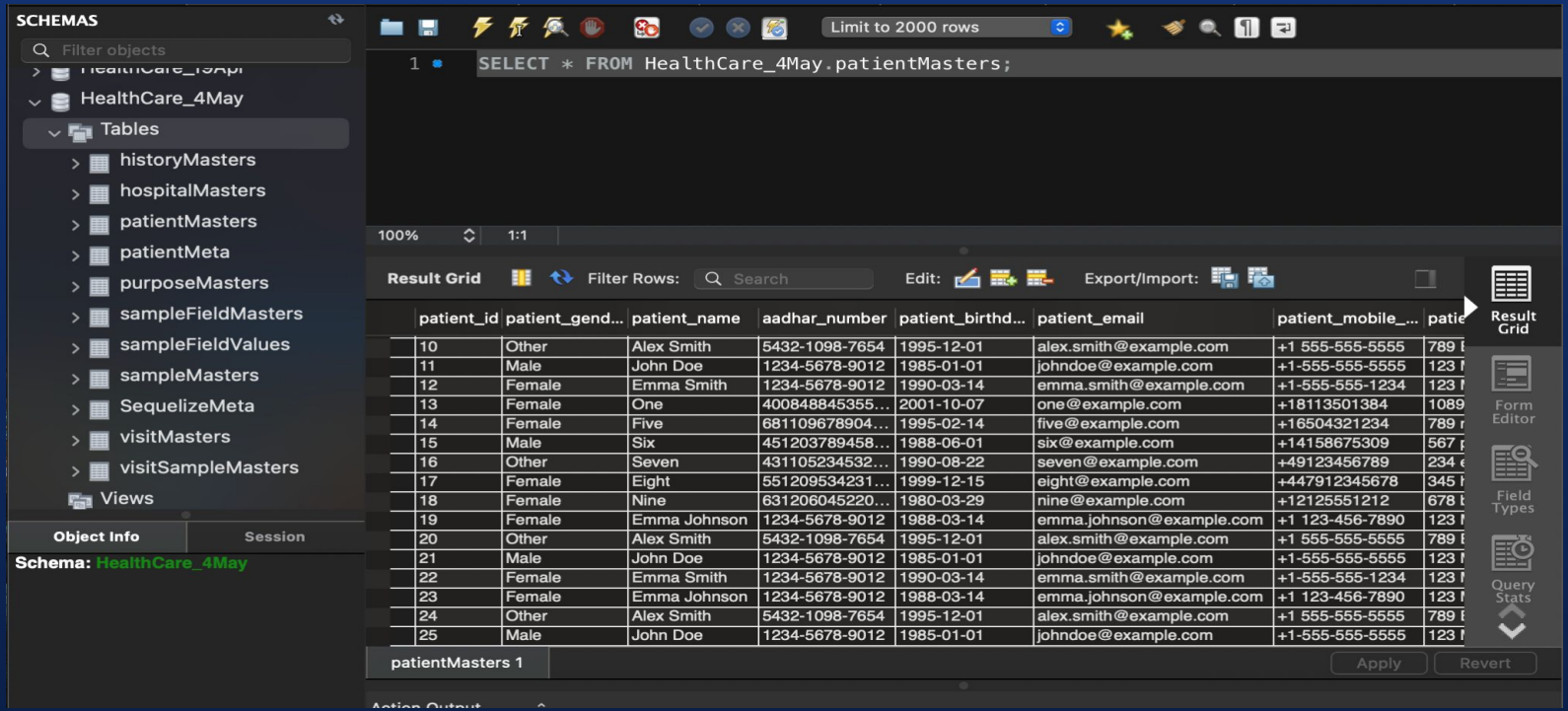
- Well Commented (every file is well commented such that new developer can easily understand whole code)
- Professional (variable name are well written, folder structure is neat)
- Reproducible (code is modular and break in small parts such that reuseable)

Challenges and Solution

- There are 3 health care Teams so we had to work together and have to make generic schema such that all team can work on single schema.
  Solution: we have meet with everyone and understand other 2's requirement and then we based on everyone’s requirement we are come up with this schema.

- Problem: There may be varying number of sample collected for patient visit and also vary fields in each sample. so we have to design our data model accordingly such that it is easily extendable and scalable.
  Solution: Database schema

Result


SCHEMAS
Filter objects
HealthCare_4May
Tables
historyMasters
hospitalMasters
patientMasters
patientMeta
purposeMasters
sampleFieldMasters
sampleFieldValues
sampleMasters
SequelizeMeta
visitMasters
visitSampleMasters
Views
Object Info
Session
Schema: HealthCare_4May
Limit to 2000 rows
1 SELECT * FROM HealthCare_4May.patientMasters;
100%
1:1
Result Grid
Filter Rows:
Search
Edit:
Export/Import:
patient_id | patient_gend... | patient_name | aadhar_number | patient_birthd... | patient_email | patient_mobile_... | patie
10 | Other | Alex Smith | 5432-1098-7654 | 1995-12-01 | alex.smith@example.com | +1 555-555-5555 | 789
11 | Male | John Doe | 1234-5678-9012 | 1985-01-01 | johndoe@example.com | +1-555-555-5555 | 123
12 | Female | Emma Smith | 1234-5678-9012 | 1990-03-14 | emma.smith@example.com | +1-555-555-1234 | 123
13 | Female | One | 400848845355... | 2001-10-07 | one@example.com | +18113501384 | 1089
14 | Female | Five | 681109678904... | 1995-02-14 | five@example.com | +16504321234 | 789
15 | Male | Six | 451203789458... | 1988-06-01 | six@example.com | +14158675309 | 567
16 | Other | Seven | 431105234532... | 1990-08-22 | seven@example.com | +49123456789 | 234
17 | Female | Eight | 551209534231... | 1999-12-15 | eight@example.com | +447912345678 | 345
18 | Female | Nine | 631206045220... | 1980-03-29 | nine@example.com | +12125551212 | 678
19 | Female | Emma Johnson | 1234-5678-9012 | 1988-03-14 | emma.johnson@example.com | +1 123-456-7890 | 123
20 | Other | Alex Smith | 5432-1098-7654 | 1995-12-01 | alex.smith@example.com | +1 555-555-5555 | 789
21 | Male | John Doe | 1234-5678-9012 | 1985-01-01 | johndoe@example.com | +1-555-555-5555 | 123
22 | Female | Emma Smith | 1234-5678-9012 | 1990-03-14 | emma.smith@example.com | +1-555-555-1234 | 123
23 | Female | Emma Johnson | 1234-5678-9012 | 1988-03-14 | emma.johnson@example.com | +1 123-456-7890 | 123
24 | Other | Alex Smith | 5432-1098-7654 | 1995-12-01 | alex.smith@example.com | +1 555-555-5555 | 789
25 | Male | John Doe | 1234-5678-9012 | 1985-01-01 | johndoe@example.com | +1-555-555-5555 | 123
patientMasters 1
Apply
Revert
Result Grid
Form Editor
Field Types
Query Stats

Result

```
2023-05-06T05:16:29.396Z - INFO - Logger initialized  with log level debug and logging to both console and file at logs/healthc
2023-05-06T05:16:30.046Z - DEBUG - Starting application
2023-05-06T05:16:30.048Z - DEBUG - Data loaded successfully
2023-05-06T05:16:30.049Z - INFO - Json to SQL script called for data length: 4
2023-05-06T05:16:30.124Z - DEBUG - Data for patient 356 added to database patientMaster
2023-05-06T05:16:30.134Z - DEBUG - Data for patient 356 added to database historyMaster
2023-05-06T05:16:30.143Z - DEBUG - Data for purpose 694, patient 356 added to database purposeMaster
2023-05-06T05:16:30.150Z - DEBUG - Data for purpose 694, patient 356 added to database purposeMeta
2023-05-06T05:16:30.162Z - DEBUG - Data for hospital 1 added to database hospitalMaster
2023-05-06T05:16:30.170Z - DEBUG - Data for visit [object Object], patient 356 added to database visitMaster
2023-05-06T05:16:30.173Z - DEBUG - Data for purpose 695, patient 356 added to database purposeMaster
2023-05-06T05:16:30.178Z - DEBUG - Data for purpose 695, patient 356 added to database purposeMeta
2023-05-06T05:16:30.182Z - DEBUG - Data for hospital 2 added to database hospitalMaster
2023-05-06T05:16:30.187Z - DEBUG - Data for visit [object Object], patient 356 added to database visitMaster
2023-05-06T05:16:30.191Z - DEBUG - Data for sample 1 added to database sampleMaster
2023-05-06T05:16:30.212Z - DEBUG - Data for hospital 2 added to database hospitalMaster
2023-05-06T05:16:30.217Z - DEBUG - Data for visit [object Object], patient 356 added to database visitMaster
2023-05-06T05:16:30.220Z - DEBUG - Data for sample 2 added to database sampleMaster
2023-05-06T05:16:30.228Z - INFO - Data Ingested Successfully.
2023-05-06T05:16:30.232Z - DEBUG - Data for patient 357 added to database patientMaster
2023-05-06T05:16:30.236Z - DEBUG - Data for patient 357 added to database historyMaster
2023-05-06T05:16:30.239Z - DEBUG - Data for purpose 696, patient 357 added to database purposeMaster
2023-05-06T05:16:30.242Z - DEBUG - Data for purpose 696, patient 357 added to database purposeMeta
2023-05-06T05:16:30.245Z - DEBUG - Data for hospital 2 added to database hospitalMaster
2023-05-06T05:16:30.248Z - DEBUG - Data for visit [object Object], patient 357 added to database visitMaster
2023-05-06T05:16:30.251Z - DEBUG - Data for sample 3 added to database sampleMaster
2023-05-06T05:16:30.269Z - DEBUG - Data for sample 1 added to database sampleMaster
2023-05-06T05:16:30.284Z - DEBUG - Data for hospital 2 added to database hospitalMaster
2023-05-06T05:16:30.287Z - DEBUG - Data for visit [object Object], patient 357 added to database visitMaster
2023-05-06T05:16:30.289Z - DEBUG - Data for sample 3 added to database sampleMaster
2023-05-06T05:16:30.303Z - DEBUG - Data for sample 1 added to database sampleMaster
2023-05-06T05:16:30.320Z - DEBUG - Data for purpose 697, patient 357 added to database purposeMaster
```

Contribution

- Jay Ghevariya
  - Building Schema
  - Json to MySQL data ingestion
- Urvish Pujara
  - Documentation
  - Table Migration
  - Backend Architecture
- Tirth Motka
  - Csv to MySQL data ingestion
  - Encryption-decryption and Hashing for data security

THANKS