Chase Maughan

SQL Clinical Database for the Huntsman Cancer Institute BMP department

Who is The Huntsman Cancer Institute?

The Huntsman Cancer Institute (HCI) is a comprehensive cancer center located here at the University of Utah. HCI is dedicated to advancing research, patient care, and education in the field of cancer. The institute focuses on conducting cutting-edge cancer research to understand the causes of cancer, develop new treatments, and improve patient outcomes. Additionally, HCI provides state-of-the-art cancer care to patients, with a recent focus on personalized treatment plans. The institution is committed to educating healthcare professionals, scientists, and the community about cancer prevention, early detection, and treatment. The overarching goal of the Huntsman Cancer Institute is to make significant contributions to the prevention and cure of cancer through its integrated approach to research, patient care, and education.

What does BMP do for the HCI?

As part of the cancer research done Huntsman Cancer Institute and Hospital conduct an extensive range of experiments and projects are conducted. Many of these projects involve studying how cancer, and the various forms of treatment, affect the blood of patients. This Blood is drawn and delivered to the Biorepository and Molecular Pathology (BMP) blood processing department. Depending on what project the blood draw is assigned to determines which strict processing procedure will be followed and what information will be collected. Traditionally, most information will be stored on a physical document called a TAF. Before processing the TAF, BMP collects and identifies patient and clinical information from a variety of sources, such as EPIC, LabVantage, Core, nurses notes from the encounter, and comments made during processing. After processing, all data collected is uploaded to another online database, and the TAF is physically stored.

Database Requirements

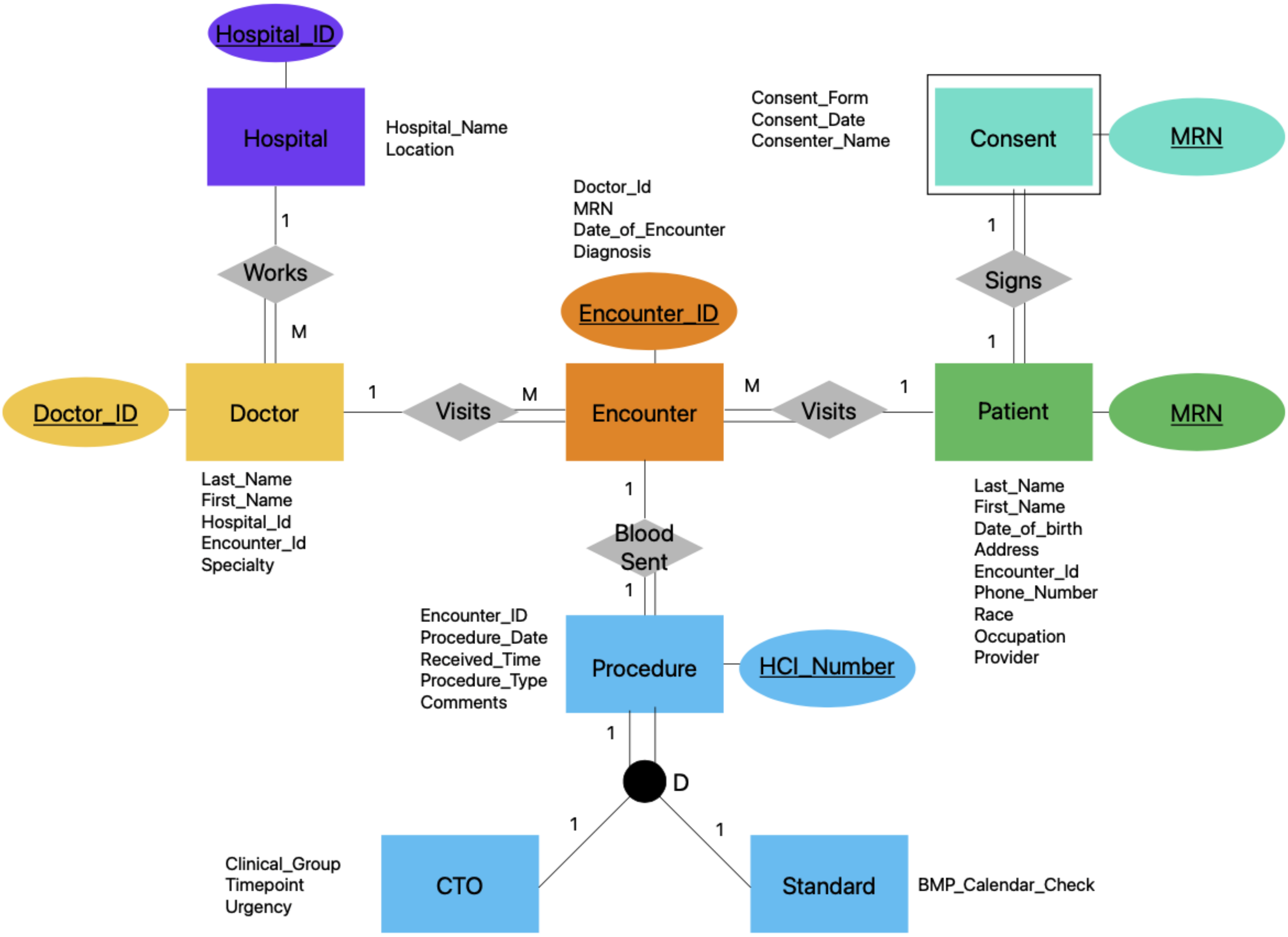
The purpose of implementing this new database within the BMP lab is to systematically organize and streamline the information needed for the various research projects. Simplifying the need for BMP personnel to access a variety of different sources for all necessary TAF information.

The database is designed with a hierarchical structure, featuring key tables that play distinct roles in managing data. The Hospital Table serves to store where the doctor of the patient encounter works (therefore noting where the encounter took place). Each doctor can work at only a singular hospital, but a hospital is composed of many doctors. Collected in this table is information such as the Hospital_Id, Hospital Name, and the location of the hospital. The Doctor Table builds on the Hospital Table foundation by detailing all pertinent doctor information, such as the identifying Doctor_Id, last name, first name, and affiliations through Foreign Key references to the Hospital and Encounter tables. The Encounter Table notes all relevant information from a patient's visit to a doctor. The primary key being the Encounter_Id, the table also includes the date of encounter, and what the doctor's diagnosis is. The Patient Table, keyed with the patient's MRN number, captures comprehensive patient details, which can be linked back to from the encounters table as necessary. Further, the Consent Table records consent-related information tied to patients. The Procedures table notes all BMP projects and has two subtypes The CTO and Standard Tables, each with unique constraints, document procedural details based on Encounter_Id.

The meticulous design of these tables ensures the efficient organization, retrieval, and analysis of data, reflecting the BMP lab's commitment to advancing cancer research through systematic and structured information management.

Fig 1 - BMP ER Diagram

The conceptual framework for clinical data is visually represented in Figure 1, where entities are depicted by rectangles, relations by diamonds, and identifying attributes by ovals. Identifying attributes within entities are also specifically underlined for clarity. All other attributes are floating next to the entity rectangle. Optional relations are denoted by a double line, emphasizing the flexibility in their presence. Cardinality is clearly marked using '1' to signify a one-to-one relationship and 'M' to denote a many-to-many relationship. Additionally, the presence of a disjoint subcomponent relation is indicated by the letter 'd,' highlighting the existence of separate and distinct subcomponents within the overall framework.



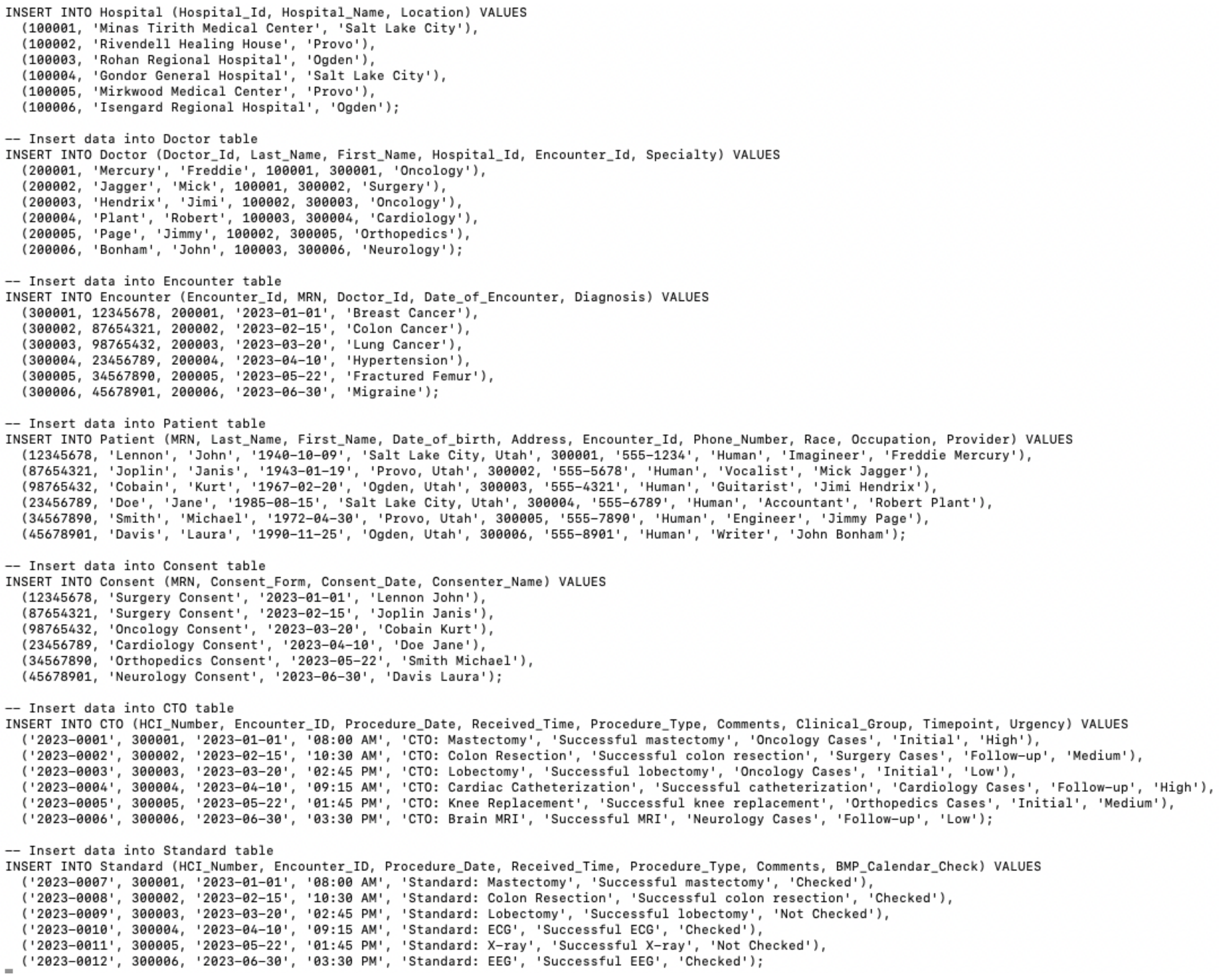
Database Design Language with Table and Attribute Descriptions.

The Tables below were all generated using a SQLITE3 Database.

```
CREATE TABLE Hospital (
    Hospital_Id INTEGER PRIMARY KEY,
    Hospital_Name TEXT,
    Location TEXT
    -- Hospital_Id: Unique identifier for each hospital.
    -- Hospital_Name: The name of the hospital.
    -- Location: The location or city of the hospital.
);

CREATE TABLE Doctor (
    Doctor_Id INTEGER PRIMARY KEY,
    Last_Name TEXT,
    First_Name TEXT,
    Hospital_Id INTEGER REFERENCES Hospital(Hospital_Id),
    Encounter_Id INTEGER REFERENCES Encounter(Encounter_Id),
    Specialty TEXT
    -- Doctor_Id: Unique identifier for each doctor.
    -- Last_Name: Last name of the doctor.
    -- First_Name: First name of the doctor.
    -- Hospital_Id: Foreign key referencing the hospital where the doctor works.
    -- Encounter_Id: Foreign key referencing a specific encounter associated with the doctor.
    -- Specialty: The medical specialty of the doctor.
);

CREATE TABLE Encounter (
    Encounter_Id INTEGER PRIMARY KEY,
    Doctor_Id INTEGER REFERENCES Doctor(Doctor_Id),
    MRN INTEGER REFERENCES Patient(MRN),
    Date_of_Encounter DATE,
    Diagnosis TEXT
    -- Encounter_Id: Unique identifier for each encounter.
    -- Doctor_Id: Foreign key referencing the doctor associated with the encounter.
    -- MRN: Foreign key referencing the patient involved in the encounter.
    -- Date_of_Encounter: The date when the encounter occurred.
    -- Diagnosis: The medical diagnosis associated with the encounter.
);


CREATE TABLE Patient (
    MRN INTEGER PRIMARY KEY,
    Last_Name TEXT,
    First_Name TEXT,
    Date_of_birth DATE,
    Address TEXT,
    Encounter_Id INTEGER REFERENCES Encounter(Encounter_Id),
    Phone_Number TEXT,
    Race TEXT,
    Occupation TEXT,
    Provider TEXT
    -- MRN: Unique identifier for each patient.
    -- Last_Name: Last name of the patient.
    -- First_Name: First name of the patient.
    -- Date_of_birth: Date of birth of the patient.
    -- Address: Address of the patient.
    -- Encounter_Id: Foreign key referencing a specific encounter associated with the patient.
    -- Phone_Number: Phone number of the patient.
    -- Race: Race or ethnicity of the patient.
    -- Occupation: Occupation of the patient.
    -- Provider: The medical provider of the patient.
);

CREATE TABLE Consent (
    MRN INTEGER PRIMARY KEY REFERENCES Patient(MRN),
    Consent_Form TEXT,
    Consent_Date DATE,
    Consenter_Name TEXT
    -- MRN: Foreign key referencing the consented patient.
    -- Consent_Form: The type of consent form.
    -- Consent_Date: The date when the consent was given.
    -- Consenter_Name: The name of the person giving consent.
);

CREATE TABLE Procedure (
    HCI_Number TEXT PRIMARY KEY,
    Encounter_ID INTEGER REFERENCES Encounter(Encounter_Id),
    Procedure_Date DATE,
    Received_Time TEXT,
    Procedure_Type TEXT,
    Comments TEXT
    -- HCI_Number: Unique identifier for each procedure.
    -- Encounter_ID: Foreign key referencing the encounter associated with the procedure.
    -- Procedure_Date: The date when the procedure occurred.
    -- Received_Time: The time when the procedure was received.
    -- Procedure_Type: The type of procedure.
    -- Comments: Additional comments or details about the procedure.
);

CREATE TABLE CTO (
    HCI_Number TEXT PRIMARY KEY REFERENCES Procedure(HCI_Number),
    Clinical_Group TEXT,
    Timepoint TEXT,
    Urgency TEXT
    -- HCI_Number: Foreign key referencing the CTO procedure.
    -- Clinical_Group: The clinical group associated with the CTO procedure.
    -- Timepoint: The timepoint of the CTO procedure.
    -- Urgency: The urgency level of the CTO procedure.
);

CREATE TABLE Standard (
    HCI_Number TEXT PRIMARY KEY REFERENCES Procedure(HCI_Number),
    BMP_Calendar_Check TEXT
    -- HCI_Number: Foreign key referencing the standard procedure.
    -- BMP_Calendar_Check: BMP calendar check status for the standard procedure.
);
```

Filling the Database with information.

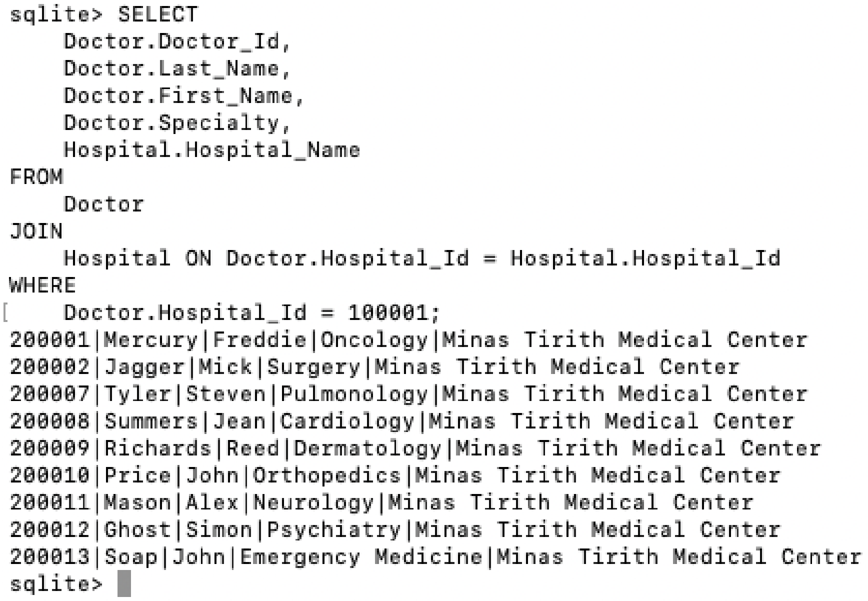
For privacy concerns real patient data was not used, instead identically formatted fake patient information was simulated using AI. Including Hospitals, Doctors, DIagnoses, etc. *Note: This is not all the information added to the tables.*



```
INSERT INTO Hospital (Hospital_Id, Hospital_Name, Location) VALUES
  (100001, 'Minas Tirith Medical Center', 'Salt Lake City'),
  (100002, 'Rivendell Healing House', 'Provo'),
  (100003, 'Rohan Regional Hospital', 'Ogden'),
  (100004, 'Gondor General Hospital', 'Salt Lake City'),
  (100005, 'Mirkwood Medical Center', 'Provo'),
  (100006, 'Isengard Regional Hospital', 'Ogden');

-- Insert data into Doctor table
INSERT INTO Doctor (Doctor_Id, Last_Name, First_Name, Hospital_Id, Encounter_Id, Specialty) VALUES
  (200001, 'Mercury', 'Freddie', 100001, 300001, 'Oncology'),
  (200002, 'Jagger', 'Mick', 100001, 300002, 'Surgery'),
  (200003, 'Hendrix', 'Jimi', 100002, 300003, 'Oncology'),
  (200004, 'Plant', 'Robert', 100003, 300004, 'Cardiology'),
  (200005, 'Page', 'Jimmy', 100002, 300005, 'Orthopedics'),
  (200006, 'Bonham', 'John', 100003, 300006, 'Neurology');

-- Insert data into Encounter table
INSERT INTO Encounter (Encounter_Id, MRN, Doctor_Id, Date_of_Encounter, Diagnosis) VALUES
  (300001, 12345678, 200001, '2023-01-01', 'Breast Cancer'),
  (300002, 87654321, 200002, '2023-02-15', 'Colon Cancer'),
  (300003, 98765432, 200003, '2023-03-20', 'Lung Cancer'),
  (300004, 23456789, 200004, '2023-04-10', 'Hypertension'),
  (300005, 34567890, 200005, '2023-05-22', 'Fractured Femur'),
  (300006, 45678901, 200006, '2023-06-30', 'Migraine');

-- Insert data into Patient table
INSERT INTO Patient (MRN, Last_Name, First_Name, Date_of_birth, Address, Encounter_Id, Phone_Number, Race, Occupation, Provider) VALUES
  (12345678, 'Lennon', 'John', '1940-10-09', 'Salt Lake City, Utah', 300001, '555-1234', 'Human', 'Imagineer', 'Freddie Mercury'),
  (87654321, 'Joplin', 'Janis', '1943-01-19', 'Provo, Utah', 300002, '555-5678', 'Human', 'Vocalist', 'Mick Jagger'),
  (98765432, 'Cobain', 'Kurt', '1967-02-20', 'Ogden, Utah', 300003, '555-4321', 'Human', 'Guitarist', 'Jimi Hendrix'),
  (23456789, 'Doe', 'Jane', '1985-08-15', 'Salt Lake City, Utah', 300004, '555-6789', 'Human', 'Accountant', 'Robert Plant'),
  (34567890, 'Smith', 'Michael', '1972-04-30', 'Provo, Utah', 300005, '555-7890', 'Human', 'Engineer', 'Jimmy Page'),
  (45678901, 'Davis', 'Laura', '1990-11-25', 'Ogden, Utah', 300006, '555-8901', 'Human', 'Writer', 'John Bonham');

-- Insert data into Consent table
INSERT INTO Consent (MRN, Consent_Form, Consent_Date, Consenter_Name) VALUES
  (12345678, 'Surgery Consent', '2023-01-01', 'Lennon John'),
  (87654321, 'Surgery Consent', '2023-02-15', 'Joplin Janis'),
  (98765432, 'Oncology Consent', '2023-03-20', 'Cobain Kurt'),
  (23456789, 'Cardiology Consent', '2023-04-10', 'Doe Jane'),
  (34567890, 'Orthopedics Consent', '2023-05-22', 'Smith Michael'),
  (45678901, 'Neurology Consent', '2023-06-30', 'Davis Laura');

-- Insert data into CTO table
INSERT INTO CTO (HCI_Number, Encounter_ID, Procedure_Date, Received_Time, Procedure_Type, Comments, Clinical_Group, Timepoint, Urgency) VALUES
  ('2023-0001', 300001, '2023-01-01', '08:00 AM', 'CTO: Mastectomy', 'Successful mastectomy', 'Oncology Cases', 'Initial', 'High'),
  ('2023-0002', 300002, '2023-02-15', '10:30 AM', 'CTO: Colon Resection', 'Successful colon resection', 'Surgery Cases', 'Follow-up', 'Medium'),
  ('2023-0003', 300003, '2023-03-20', '02:45 PM', 'CTO: Lobectomy', 'Successful lobectomy', 'Oncology Cases', 'Initial', 'Low'),
  ('2023-0004', 300004, '2023-04-10', '09:15 AM', 'CTO: Cardiac Catheterization', 'Successful catheterization', 'Cardiology Cases', 'Follow-up', 'High'),
  ('2023-0005', 300005, '2023-05-22', '01:45 PM', 'CTO: Knee Replacement', 'Successful knee replacement', 'Orthopedics Cases', 'Initial', 'Medium'),
  ('2023-0006', 300006, '2023-06-30', '03:30 PM', 'CTO: Brain MRI', 'Successful MRI', 'Neurology Cases', 'Follow-up', 'Low');

-- Insert data into Standard table
INSERT INTO Standard (HCI_Number, Encounter_ID, Procedure_Date, Received_Time, Procedure_Type, Comments, BMP_Calendar_Check) VALUES
  ('2023-0007', 300001, '2023-01-01', '08:00 AM', 'Standard: Mastectomy', 'Successful mastectomy', 'Checked'),
  ('2023-0008', 300002, '2023-02-15', '10:30 AM', 'Standard: Colon Resection', 'Successful colon resection', 'Checked'),
  ('2023-0009', 300003, '2023-03-20', '02:45 PM', 'Standard: Lobectomy', 'Successful lobectomy', 'Not Checked'),
  ('2023-0010', 300004, '2023-04-10', '09:15 AM', 'Standard: ECG', 'Successful ECG', 'Checked'),
  ('2023-0011', 300005, '2023-05-22', '01:45 PM', 'Standard: X-ray', 'Successful X-ray', 'Not Checked'),
  ('2023-0012', 300006, '2023-06-30', '03:30 PM', 'Standard: EEG', 'Successful EEG', 'Checked');
```

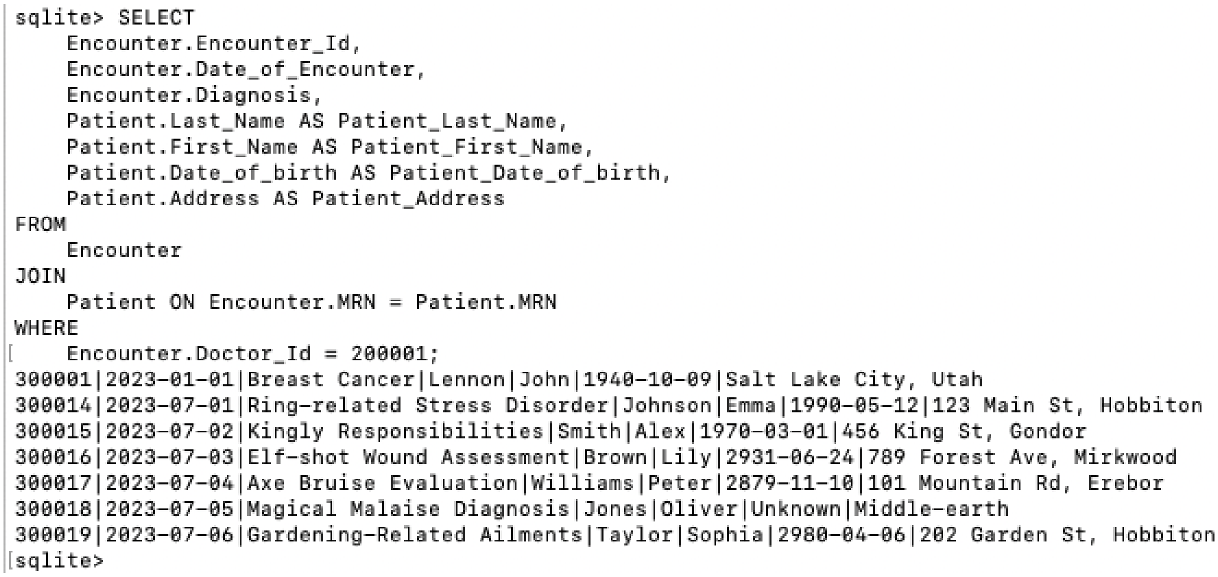
Common query commands examples

Query all doctors at a given hospital



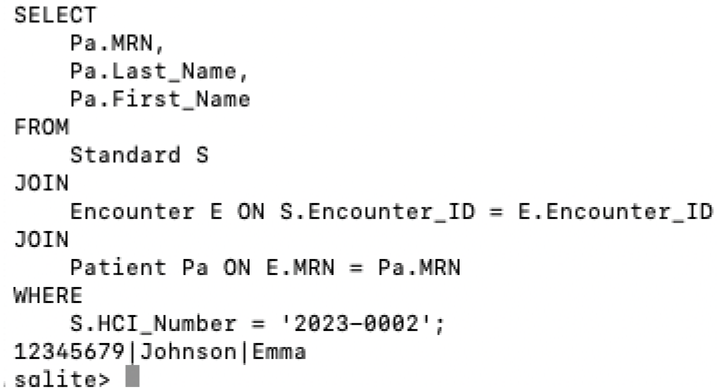
```
sqlite> SELECT
    Doctor.Doctor_Id,
    Doctor.Last_Name,
    Doctor.First_Name,
    Doctor.Specialty,
    Hospital.Hospital_Name
FROM
    Doctor
JOIN
    Hospital ON Doctor.Hospital_Id = Hospital.Hospital_Id
WHERE
[    Doctor.Hospital_Id = 100001;
200001|Mercury|Freddie|Oncology|Minas Tirith Medical Center
200002|Jagger|Mick|Surgery|Minas Tirith Medical Center
200007|Tyler|Steven|Pulmonology|Minas Tirith Medical Center
200008|Summers|Jean|Cardiology|Minas Tirith Medical Center
200009|Richards|Reed|Dermatology|Minas Tirith Medical Center
200010|Price|John|Orthopedics|Minas Tirith Medical Center
200011|Mason|Alex|Neurology|Minas Tirith Medical Center
200012|Ghost|Simon|Psychiatry|Minas Tirith Medical Center
200013|Soap|John|Emergency Medicine|Minas Tirith Medical Center
sqlite>
```

Query all patient encounters for a particular doctor



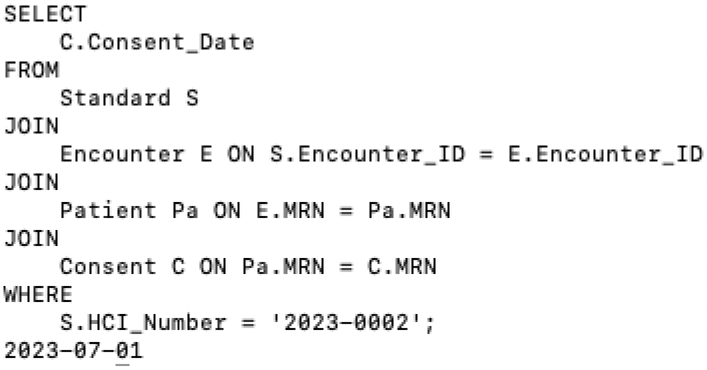
```
sqlite> SELECT
    Encounter.Encounter_Id,
    Encounter.Date_of_Encounter,
    Encounter.Diagnosis,
    Patient.Last_Name AS Patient_Last_Name,
    Patient.First_Name AS Patient_First_Name,
    Patient.Date_of_birth AS Patient_Date_of_birth,
    Patient.Address AS Patient_Address
FROM
    Encounter
JOIN
    Patient ON Encounter.MRN = Patient.MRN
WHERE
[    Encounter.Doctor_Id = 200001;
300001|2023-01-01|Breast Cancer|Lennon|John|1940-10-09|Salt Lake City, Utah
300014|2023-07-01|Ring-related Stress Disorder|Johnson|Emma|1990-05-12|123 Main St, Hobbiton
300015|2023-07-02|Kingly Responsibilities|Smith|Alex|1970-03-01|456 King St, Gondor
300016|2023-07-03|Elf-shot Wound Assessment|Brown|Lily|2931-06-24|789 Forest Ave, Mirkwood
300017|2023-07-04|Axe Bruise Evaluation|Williams|Peter|2879-11-10|101 Mountain Rd, Erebor
300018|2023-07-05|Magical Malaise Diagnosis|Jones|Oliver|Unknown|Middle-earth
300019|2023-07-06|Gardening-Related Ailments|Taylor|Sophia|2980-04-06|202 Garden St, Hobbiton
[sqlite>
```

Get a Patient name and MRN for a particular HCI Number



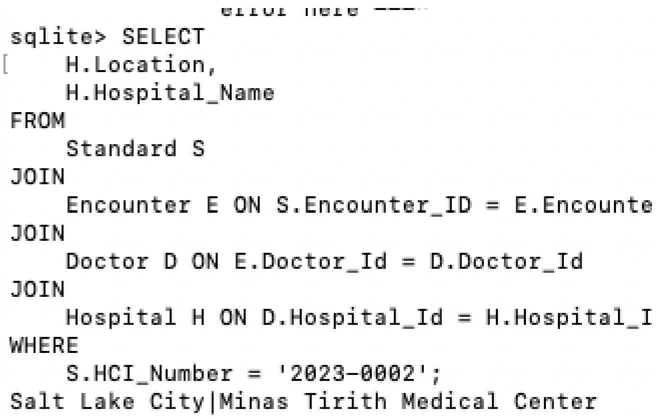
```
SELECT
    Pa.MRN,
    Pa.Last_Name,
    Pa.First_Name
FROM
    Standard S
JOIN
    Encounter E ON S.Encounter_ID = E.Encounter_ID
JOIN
    Patient Pa ON E.MRN = Pa.MRN
WHERE
    S.HCI_Number = '2023-0002';
12345679|Johnson|Emma
sqlite>
```

Get a Consent Date for a particular HCI Number



```
SELECT
    C.Consent_Date
FROM
    Standard S
JOIN
    Encounter E ON S.Encounter_ID = E.Encounter_ID
JOIN
    Patient Pa ON E.MRN = Pa.MRN
JOIN
    Consent C ON Pa.MRN = C.MRN
WHERE
    S.HCI_Number = '2023-0002';
2023-07-01
```

Get Hospital location and name for a particular HCI Number



```
               error here ---
sqlite> SELECT
    H.Location,
    H.Hospital_Name
FROM
    Standard S
JOIN
    Encounter E ON S.Encounter_ID = E.Encounte
JOIN
    Doctor D ON E.Doctor_Id = D.Doctor_Id
JOIN
    Hospital H ON D.Hospital_Id = H.Hospital_I
WHERE
    S.HCI_Number = '2023-0002';
Salt Lake City|Minas Tirith Medical Center
```

Discussion

Going forward I believe that two worries need to be addressed:

1) **Usability in lab**

   During the development of the database it became evident that although it could serve a realistic purpose for BMP, My colleagues lack experience with SQL, which means that further development should be focused on connecting the database to tools like R or Python for broader usability.

2) **Application in real world**

   The use of fictitious data during development offered advantages by allowing for unrestricted design and interaction with the database. Although, I don't know how well the database would transition to a real life lab environment. My main worries here are privacy regulation laws, bugs and errors that would occur from real world scenarios, and how it would fair with adopting unknown specifics needed down the road.