**COMP 353/453 Project Phase 2**

Deliverables:

* [Revised] Description of the organization
* [Revised] ER diagram with min/max specifications
* [Revised] Constraints not in ER diagram
* [Revised] Relational Schema
* Queries with descriptions
* DML, DDL, SQL statements
* PostgreSQL Implementation

Assessment:

* Group status report

Phase Leader Michael Saban \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Phase Recorder Sathvik Maridasana Nagaraj \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Phase Checker Aaron Myrold \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Technical Advisor Crisi Patelis \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Introduction**

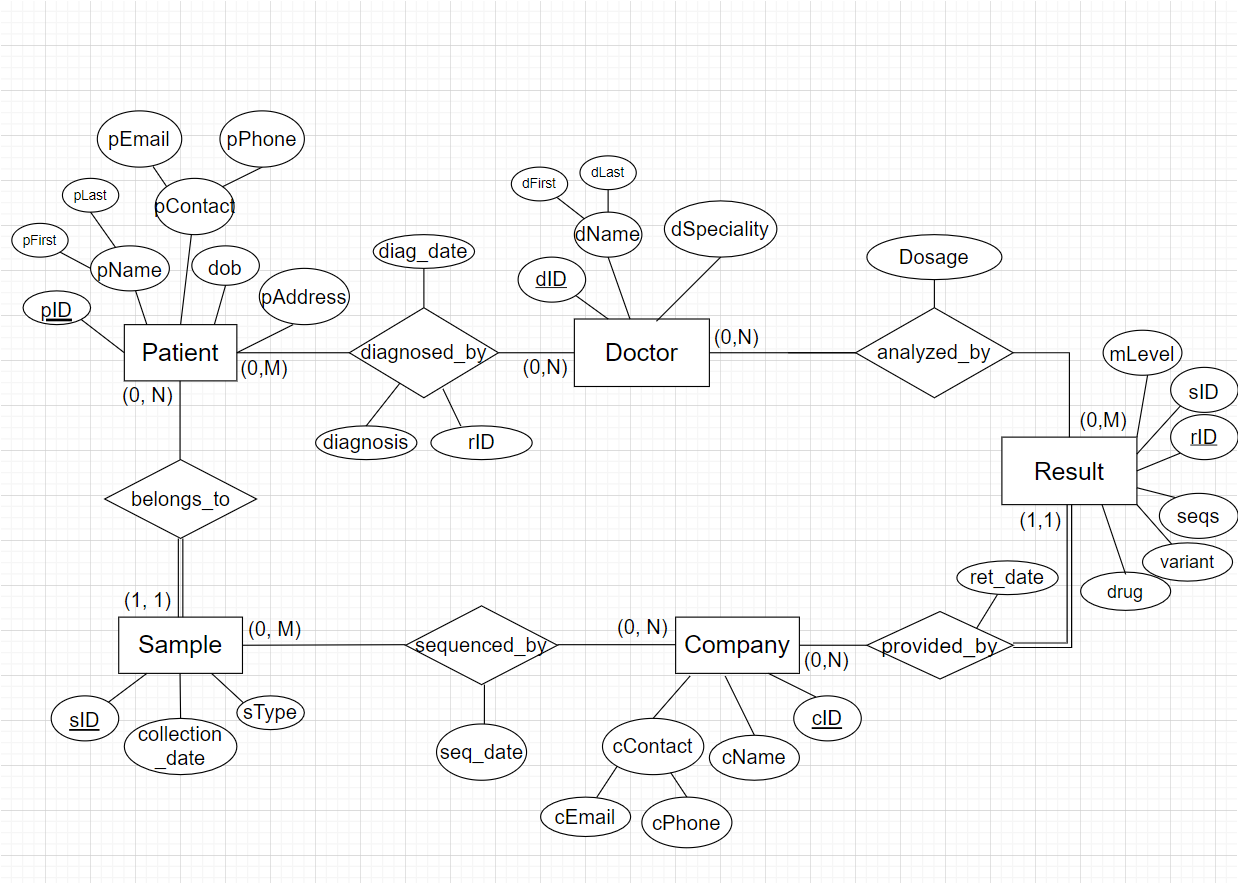
The database is based on a tool that Nick Miller has worked on called Flype. It consists of a relational database, in-house developed bioinformatics analysis pipelines, a connection framework to connect to external tools, and a web interface to provide easy and efficient access to the information stored in the database. The relational database stores things like patient information, sample and result information, result interpretation, and other information needed to provide quick and personalized healthcare.

1. **Description of the Organization**

The Healthcare database stores patient information along with their data related to test results such as Next Generation Sequencing (NGS) and Pharmacogenomic (PGX) testing. Maintaining a relational database that not only holds all patient information and results, but also interpretation of said results provides a quick and efficient way for healthcare providers to incorporate computational biology and genetic testing into a more personalized treatment for patients.

* Patients require genetic testing for a variety of illnesses. This involves collecting the correct samples and sending them off for sequencing. To efficiently track this, patients are given a unique patient ID along with storing their names, birth dates, and addresses. Patients may have multiple doctors working with them at one time or none at all.
* Over time, patients may require multiple different tests, so each sample includes a unique sample ID, a collection date, and information on the type of sample collected. Patients can have multiple samples taken, but each sample can only belong to one patient.
* A single collected sample may be sequenced by multiple companies, so contact information including the company name, email, and phone number are included along with a unique company ID. This allows tracking of communications with each company and when and where samples are sent. Every individual company may be sequencing multiple samples and the sequencing dates are stored in the relationship between the company and the sample.
* The results returned by the company are associated with the sample and patient IDs and include not only the sequencing results, but also identified and annotated gene variants of the sequenced samples with corresponding metabolism levels of a given drug based on those variants. A result can only belong to one company, but a company may send multiple results back. The sequencing return dates are stored in the relationship between the company and the result.
* While sequencing results contain annotation, a healthcare professional needs to analyze/interpret and relay that information to the patient’s electronic health record (EHR). Doctors’ names and specialties are stored under a unique ID and a diagnosis date is stored in the relationship between the doctor and the patient. Doctors can have multiple patients or none at all.

1. **ER Diagram**

****

1. **ER Diagram Uncaptured Constraints**

* Sequence strings in Results must be uppercase
* Address format: Street Number, followed by Direction (if applicable), then by Street Name(s)
  + No abbreviations such as St, Ave, N, S, etc.
  + Number, direction, and each name separated by a space
* Name format:
  + Patient and Doctor names: first name beginning with capital letter (pFirst/dFirst), last name beginning with a capital letter (pLast, dLast)
  + Company names:
* Date data types use the following format: YYYY-MM-DD
* Phone numbers use following format: XXX-XXX-XXXX
* ID format:
  + Patient IDs (pID): begin with “P” followed by three-digit number
  + Sample IDs (sID): begin with “S” followed by three-digit number
  + Result IDs (rID): begin with “R” followed by three-digit number
  + Doctor IDs (dID): begin with “D” followed by three-digit number
* Doctor Analysis
  + Dosage numerical values in mg
  + NULL represents incompatibility of drug with patient

1. **Relational Schema**

**5.1 Relational Schema with Referential Integrity**

Patient(pID,pFirst,pLast,pDOB,pAddress,pEmail,pPhone)

Sample(sID,collection\_date,sType,pID)

Foreign key (pID) references Patient (pID)

Sequenced\_By(cID,sID,seq\_date)

Foreign key (cID) references Company (cID)

Foreign key (sID) references Sample (sID)

Company(cID,cName,cEmail,cPhone)

provided\_by(cID,rID, ret\_date)

Foreign key (cID) references Company (cID)

Foreign key (rID) references Result (rID)

Result(rID,seqs,variant,drug,mlevel,sID)

Foreign key (sID) references Sample (sID)

analyzed\_by(rID,dID,dosage)

Foreign key (rID) references Result (rID)

Foreign key (dID) references Doctor (rID)

Doctor(dID,dFirst,dLast,dSpecialty)

diagnosed\_by(dID,pID,diag\_date,rID,diagnosis)

Foreign key (dID) references Doctor (dID)

Foreign key (pID) references Patient (pID)

**5.2 Relational Table Details**

| **Table Name** | **Attribute** | **Description** |
| --- | --- | --- |
| Patient | pID | unique patient ID |
| pFirst | patient’s first name |
| pLast | patient’s last name |
| pDOB | patient’s date of birth |
| pAddress | patient’s address |
| pEmail | patient’s email address |
| pPhone | patient’s phone number |
| Sample | sID | unique sample ID |
| collection\_date | date of sample’s collection |
| sType | type of sample |
| pID | patient ID |
| Sequenced\_by | cID | unique company ID |
| sID | unique sample ID |
| sequence\_date | Date sample was sequenced |
| Company | cID | unique company ID |
| cName | company name |
| cEmail | company email address |
| cPhone | company phone number |
| Result | rID | result ID |
| seqs | sequences |
| variant | Gene variant determined from sequencing |
| drug | Drug needed by patient |
| mlevel | Metabolism level of drug based on gene variant |
| sID | Sample ID |
| provided\_by | cID | Company ID |
| rID | Result ID |
| ret\_date | Date sequenced sample was returned |
| Doctor | dID | unique doctor ID |
| dFirst | doctor’s first name |
| dLast | doctor’s last name |
| dSpecialty | doctor’s specialty |
| analyzed\_by | rID | Result ID |
| dID | Doctor ID |
| dosage | Recommended dosage of drug based on patient genes |
| Diagnosed\_by | dID | unique doctor ID |
| pID | unique patient ID |
| diag\_date | Date of diagnosis |
| rID | Result ID |
| diagnosis | Doctor diagnosis of patient |

1. **Queries**

| **Query Name** | **Descriptions** | **Output** | **Relations Accessed** |
| --- | --- | --- | --- |
| High Metabolizing Patients | Find information on patients who have high drug metabolism and the number of drugs they are prescribed. Group by the patient ID and order by the number of drugs taken. | * pID * pFirst * pLast * pPhone * Drug count | * Patient * Sample * Result |
| maxDrugDosage | For each drug, find the highest dosage given to patient(s). Return patient information including first and last name, as well as patient ID. Group by drug in subquery and by patient ID/drug in main query, then sort the results in patient ID ascending order. | * pID * pFirst * pLast * drug * maxDosage | * Patient * Sample * Result * Analyzed\_by |
| doctorSampleCount | For each doctor, determine the number of each sample type that they have requested | * dID * dFirst * dLast * sType * sCount | * Diagnosed\_by * Patient * sample |
| doctorUniqueVariants | For each doctor, determine the number of variants analyzed and display those having doctor’s data having analyzed count >= 2. Order by variant count. | * dID * dFirst * dLast * dspecialty * variant\_count | * Doctor * Analyzed\_by * result |

1. **DDL + DML + DQL (Including output screenshots)**

**DDL:**

DROP TABLE IF EXISTS Patient CASCADE;

DROP TABLE IF EXISTS Sample CASCADE;

DROP TABLE IF EXISTS Company CASCADE;

DROP TABLE IF EXISTS Result CASCADE;

DROP TABLE IF EXISTS Doctor CASCADE;

DROP TABLE IF EXISTS Sequenced\_by CASCADE;

DROP TABLE IF EXISTS provided\_by CASCADE;

DROP TABLE IF EXISTS analyzed\_by CASCADE;

DROP TABLE IF EXISTS Diagnosed\_by CASCADE;

CREATE TABLE Patient

(pID varchar (50) PRIMARY KEY,

pFirst varchar (50),

pLast varchar (50),

pDOB date,

pAddress varchar (50),

pEmail varchar (50),

pPhone varchar (50));

CREATE TABLE Sample

(sID varchar (50) PRIMARY KEY,

collection\_date date,

sType varchar (50),

pID varchar (50),

FOREIGN KEY (pID) REFERENCES Patient(pID));

CREATE TABLE Company

(cID varchar (50) PRIMARY KEY,

cName varchar (50),

cEmail varchar (50),

cPhone varchar (50));

CREATE TABLE Result

(rID varchar (50) PRIMARY KEY,

seqs varchar (100),

variant varchar (50),

drug varchar (50),

mlevel varchar (50),

sID varchar(50),

FOREIGN KEY (sID) REFERENCES Sample(sID)

);

CREATE TABLE Doctor

(dID varchar (50) PRIMARY KEY,

dFirst varchar (50),

dLast varchar (50),

dSpecialty varchar (50));

CREATE TABLE Sequenced\_by

(cID varchar (50),

sID varchar (50),

sequence\_date date,

PRIMARY KEY(cID,sID),

FOREIGN KEY (cID) REFERENCES Company(cID),

FOREIGN KEY (sID) REFERENCES Sample(sID)

);

CREATE TABLE Provided\_by

(cID varchar (50),

rID varchar (50) PRIMARY KEY,

ret\_date date,

FOREIGN KEY (cID) REFERENCES Company(cID),

FOREIGN KEY (rID) REFERENCES Result(rID)

);

CREATE TABLE Analyzed\_by

(rID varchar (50),

dID varchar (50),

dosage numeric,

PRIMARY KEY(rID,dID),

FOREIGN KEY (rID) REFERENCES Result(rID),

FOREIGN KEY (dID) REFERENCES Doctor(dID)

);

CREATE TABLE Diagnosed\_by

(dID varchar (50),

pID varchar (50),

diag\_date date,

rID varchar(50),

diagnosis varchar(50),

PRIMARY KEY(dID,pID),

FOREIGN KEY (dID) REFERENCES Doctor(dID),

FOREIGN KEY (pID) REFERENCES Patient(pID),

FOREIGN KEY (rID) REFERENCES Result(rID)

);

**DML:**

INSERT INTO Patient(pID,pFirst,pLast,pDOB,pAddress,pEmail,pPhone)

VALUES

('P001','Bradley','Ostberg','1978-05-23','1433 Cherry Street, Denver, Colorado','bostberg@gmail.com','720-123-4567'),

('P002','Shirlee','Mould','1986-01-14','789 Elmwood Avenue, Austin, Texas','smould@yahoo.com','512-789-0123'),

('P003','Shania','Graves','1992-08-07','5279 Maple Drive, Seattle, Washington','sgraves@gmail.com','206-555-6789'),

('P004','Wisteria','Poole','1972-10-11','2218 Oak Street, New Orleans, Louisiana','wpoole@hotmail.com','504-234-5678'),

('P005','Sasha','Law','1999-03-29','4002 Pine Avenue, Sacramento, California','slaw@gmail.com','916-345-6789'),

('P006','Tahnee','Harlan','1983-06-09','9026 Cedar Lane, Indianapolis, Indiana','tharlan@yahoo.com','317-456-7890'),

('P007','Kae','Andrews','1979-12-01','6890 Birchwood Drive, Atlanta, Georgia','kandrews@yahoo.com','404-567-8901'),

('P008','Dora','Peyton','1996-04-12','1753 Spruce Street, Portland, Oregon','dpeyton@gmail.com','503-678-9012'),

('P009','Orrell','Scrivener','1988-09-03','3387 Aspen Court, Baltimore, Maryland','oscrivener@gmail.com','410-789-0123'),

('P010','Dorinda','Law','1976-11-25','620 Poplar Road, Kansas City, Missouri','dlaw@hotmail.com','816-234-5678')

;

INSERT INTO Sample(sID,collection\_date,sType,pID)

VALUES

('S001','2016-05-02','oral','P001'),

('S002','2016-05-02','blood','P001'),

('S003','2016-05-18','blood','P002'),

('S004','2019-08-21','urine','P003'),

('S005','2017-11-20','oral','P004'),

('S006','2017-12-13','blood','P004'),

('S007','2020-06-04','oral','P005'),

('S008','2020-03-02','oral','P006'),

('S009','2018-07-23','urine','P007'),

('S010','2018-07-23','blood','P007'),

('S011','2020-06-08','oral','P008'),

('S012','2020-07-19','oral','P009'),

('S013','2020-10-12','oral','P003'),

('S014','2021-01-06','oral','P010'),

('S015','2020-11-12','oral','P010')

;

INSERT INTO Result(rID,seqs,Variant,drug,mlevel,sID) VALUES

('R001','ATGGTCTTACTTGGTCTTGCAGAAGCAGGGTATGGAACAGTCCCTTTGTCTTCC','CYP2C19\*1/\*17','Clopidogrel','rapid','S001'),

('R002','GTCCTGCTCGCGCGCTCGCGCGCGCGCGCGCGCTGCGCGCTGCGCGCGCGCGC','CYP2C19\*1/\*17','Voriconazole','rapid','S002'),

('R003','CGGAGTGACACGTCTTGAACTGTGATGTTGTGTCTTCAGTTTCCGAGAAGGGC','CYP2C19\*1/\*1','Voriconazole','normal','S003'),

('R004','TGCTGCCAACTTGGAGGCGCAGCGCGAGCGCGCGCGCGCGCGCGCGCGCGCGC','CYP2B6\*1/\*4','Efavirenz','rapid','S004'),

('R005','GAGGGGGATGTTGGAGCTGCGGCGTTGCCTCTGGGGTTCTAGGTGTTTTGCTG','CYP2B6\*6/\*6','Efavirenz','poor','S005'),

('R006','GGAGCGTGCGCTTGCGCGCGAGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGC','CYP2C19\*17/\*17','Celecoxib','ultrarapid','S006'),

('R007','ACGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCG','CYP2C19\*17/\*17','Voriconazole','ultrarapid','S007'),

('R008','GATTTGGTTGGGGAGTTGCTGAGGCAGAAGGCTGGCCAGTGTTCTCTGATTTA','CYP2C19\*2/\*2','Clopidogrel','poor','S008'),

('R009','GCTCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCG','CYP2C19\*1/\*17','Voriconazole','rapid','S009'),

('R010','TGGTACGTGTGAGTCCAGGGTCCAGGATAGGCGTCTCCATCCCTGTGATGGG','CYP2C19\*17/\*17','Clopidogrel','ultrarapid','S010'),

('R011','ATCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGC','CYP2B6\*1/\*4','Efavirenz','rapid','S011'),

('R012','CGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGC','CYP2B6\*1/\*1','Efavirenz','normal','S012'),

('R013','GGAGCTTTGGGAGGAAGCCAGGAAGAGTGCTCAGAGCTGGGAGGTGTTGTGC','CYP2C19\*1/\*17','Clopidogrel','rapid','S013'),

('R014','CTCTCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGCGC','CYP2C19\*1/\*1','Clopidogrel','normal','S014'),

('R015','GTGAGTGTGAGTCTGGAGAGGATGAGGACAGGGAAGAGGGACGGGAGGGAGC','CYP2C19\*1/\*1','Voriconazole','normal','S015')

;

INSERT INTO Company(cID,cName,cEmail,cPhone) VALUES

('C001','Medtronic','info@medtronic.com','692-245-0958'),

('C002','Novartis','contact.center@novartis.com','789-574-8531'),

('C003','Max Health','info.mh@max.com','727-382-0333')

;

INSERT INTO Sequenced\_By(cID,sID,sequence\_date) VALUES

('C001','S001','2016-05-14'),

('C001','S002','2016-05-14'),

('C001','S003','2016-06-01'),

('C002','S004','2019-08-30'),

('C002','S005','2017-12-07'),

('C001','S006','2017-12-29'),

('C003','S007','2020-06-21'),

('C003','S008','2020-03-22'),

('C001','S009','2018-08-03'),

('C001','S010','2018-08-03'),

('C002','S011','2020-06-27'),

('C002','S012','2020-07-29'),

('C001','S013','2020-10-25'),

('C003','S014','2021-01-17'),

('C003','S015','2020-11-30')

;

INSERT INTO provided\_by(cID,rID, ret\_date) VALUES

('C001','R001','2016-05-16'),

('C001','R002','2016-05-16'),

('C001','R003','2016-06-04'),

('C002','R004','2019-09-02'),

('C002','R005','2017-12-10'),

('C001','R006','2018-01-02'),

('C003','R007','2020-06-23'),

('C003','R008','2020-03-25'),

('C001','R009','2018-08-05'),

('C001','R010','2018-08-05'),

('C002','R011','2020-06-29'),

('C002','R012','2020-08-01'),

('C001','R013','2020-10-27'),

('C003','R014','2021-01-19'),

('C003','R015','2020-12-03')

;

INSERT INTO Doctor(dID,dFirst,dLast,dSpecialty) VALUES

('D001','Vincent','Bird','Internal Medicine'),

('D002','Andrea','Staford','Family Medicine'),

('D003','Dennis','Franklin','Pathology'),

('D004','Marc','Ruiz','Cardiology'),

('D005','Halima','Kim','Medical Genetics'),

('D006','Jackson','Cannon','Family Medicine'),

('D007','Isaiah','Huffman','Internal Medicine')

;

INSERT INTO analyzed\_by(rID,dID,dosage) VALUES

('R001', 'D003', '300'),

('R002', 'D004', '100'),

('R003', 'D005', '75'),

('R004', 'D001', '600'),

('R005', 'D005', NULL),

('R006', 'D006', '65'),

('R007', 'D004', '65'),

('R008', 'D007', NULL),

('R009', 'D007', '350'),

('R010', 'D003', '65'),

('R011', 'D006', '50'),

('R012', 'D003', '200'),

('R013', 'D002', '65'),

('R014', 'D003', '75'),

('R015', 'D001', '350')

;

INSERT INTO Diagnosed\_by(dID,pID,diag\_date,rID,diagnosis) VALUES

('D003','P001','2016-05-20','R001','Peripheral vascular disease'),

('D004','P001','2016-05-20','R002','Aspergillosis'),

('D005','P002','2016-06-07','R003','Candida esophagitis'),

('D001','P003','2019-09-06','R004','HIV'),

('D005','P004','2018-01-05','R005','HIV'),

('D006','P004','2018-01-05','R006','Osteoarthritis'),

('D004','P005','2020-06-26','R007','Pyelonephritis'),

('D007','P006','2020-03-29','R008','Angina'),

('D007','P007','2018-08-08','R009','Gastroenteritis'),

('D003','P007','2018-08-08','R010','Coronary artery disease'),

('D006','P008','2020-07-03','R011','HIV'),

('D003','P009','2020-08-04','R012','HIV'),

('D002','P001','2020-11-01','R013','Angina'),

('D003','P010','2021-01-22','R014','Myocardial infarction'),

('D001','P010','2020-12-05','R015','Cystitis')

;

**DQL:**

**High Metabolizing Patients**

SELECT Patient.pID,pFirst,pLast,pPhone,count(drug) as drugCount

FROM Patient,Sample,Result

WHERE Patient.pID = Sample.pID AND

Sample.sID = Result.sID AND

(mlevel = 'rapid' OR mlevel = 'ultrarapid')

GROUP BY Patient.pID

ORDER BY count(drug) DESC

;

**Output1:**

"pid","pfirst","plast","pPhone","drugCount"

"P001","Bradley","Ostberg","720-123-4567",2

"P003","Shania","Graves","206-555-6789",2

"P007","Kae","Andrews","404-567-8901",2

"P004","Wisteria","Poole","504-234-5678",1

"P005","Sasha","Law","916-345-6789",1

"P008","Dora","Peyton","503-678-9012",1

**maxDrugDosages**

SELECT Patient.pID, Patient.pFirst, Patient.pLast, Result.drug, max(Analyzed\_by.dosage) as maxDosage

FROM Patient, Sample, Result, Analyzed\_by

WHERE Patient.pID = Sample.pID AND

Sample.sID = Result.sID AND

Result.rID = Analyzed\_by.rID AND

(drug,dosage) IN (SELECT drug, max(dosage)

FROM Analyzed\_by,Result

WHERE Result.rID = Analyzed\_by.rID

GROUP BY drug)

GROUP BY Patient.pID, Result.drug

ORDER BY Patient.pID ASC

;

**Output2:**

"*pID*","*pFirst*","*pLast*","*drug*","*maxDosage*"

"P001","Bradley","Ostberg","Clopidogrel",300

"P003","Shania","Graves","Efavirenz",600

"P004","Wisteria","Poole","Celecoxib",65

"P007","Kae","Andrews","Voriconazole",350

"P010","Dorinda","Law","Voriconazole",350

**doctorSampleCount**

SELECT doctor.did, dfirst, dlast, sample.stype, count(stype) as sampleNum

FROM doctor, diagnosed\_by, patient, sample

WHERE doctor.did = diagnosed\_by.did AND

diagnosed\_by.pid = patient.pid AND

patient.pid = sample.pid

GROUP BY doctor.did, sample.stype

ORDER BY doctor.did ASC;

**Output3:**

"did" "dfirst" "dlast" "stype" "samplenum"

"D001" "Vincent" "Bird" "oral" 3

"D001" "Vincent" "Bird" "urine" 1

"D002" "Andrea" "Staford" "blood" 1

"D002" "Andrea" "Staford" "oral" 1

"D003" "Dennis" "Franklin" "blood" 2

"D003" "Dennis" "Franklin" "oral" 4

"D003" "Dennis" "Franklin" "urine" 1

"D004" "Marc" "Ruiz" "blood" 1

"D004" "Marc" "Ruiz" "oral" 2

"D005" "Halima" "Kim" "blood" 2

"D005" "Halima" "Kim" "oral" 1

"D006" "Jackson" "Cannon" "blood" 1

"D006" "Jackson" "Cannon" "oral" 2

"D007" "Isaiah" "Huffman" "blood" 1

"D007" "Isaiah" "Huffman" "oral" 1

"D007" "Isaiah" "Huffman" "urine" 1

**#doctorUniqueVariants**

SELECT d.dID, d.dFirst, d.dLast, d.dSpecialty, COUNT(DISTINCT r.variant) AS variant\_count

FROM doctor d, Analyzed\_by A, Result r

WHERE d.dID = A.dID

AND r.rID = A.rID

GROUP BY d.dID

HAVING COUNT(DISTINCT r.variant) >= 2

ORDER BY variant\_count DESC;

**OUTPUT4 :**

"did" "dfirst" "dlast" "dspecialty" "variant\_count"

"D003" "Dennis" "Franklin" "Pathology" 4

"D004" "Marc" "Ruiz" "Cardiology" 2

"D007" "Isaiah" "Huffman" "Internal Medicine" 2

"D001" "Vincent" "Bird" "Internal Medicine" 2

"D005" "Halima" "Kim" "Medical Genetics" 2

"D006" "Jackson" "Cannon" "Family Medicine" 2

**Group Status Report**

**Phase 2**

Dates and attendance of team meetings:

Tuesday, March 21st 11:30am - 12:45pm All group members

Progress overview as of March 21st:

The Phase 2 deliverables have been completed. Our Pharmacogenomic Healthcare Database has been updated with tables and data and is complete. Below are the responsibilities of each team member for the creation of tables, queries, and revisions to our previously outlined model.

Contributions of group members:

Phase Leader: Michael Saban

* Description of the Organization revisions
* Uncaptured constraints - Doctor Analysis
* Relational schema revisions
* ER Diagram revisions
* QUERY IMPLEMENTATION: highMetabolizingPatients
* TABLE DML: Patient
* TABLE DML: Sample
* TABLE DML: Results
* DDL Revisions and updates
* Group status report

Phase Recorder: Sathvik Maridasana Nagaraj

* Outline for analyzed\_by DML
* DML TABLE: Company
* QUERY IMPLEMENTATION: doctorUniqueVariants

Phase Checker: Aaron Myrold

* DDL Draft
* QUERY IMPLEMENTATION: doctorSampleCount

Technical Advisor: Crisi Patelis

* Reached out to Prof. Silva for data review
* QUERY IMPLEMENTATION: maxDrugDosage
* TABLE DML: Sequenced\_by
* TABLE DML: provided\_by
* TABLE DML: analyzed\_by
* TABLE DML: Diagnosed\_by
* TABLE DML: Doctor
* Group status report
* Uncaptured constraints revisions - Format specifications