**USE CASE STUDY FINAL REPORT**

**Group No: 12**

**Title: Clinical Trial Data Management of Breast Cancer Treatment**

**Student Name: Yongchao Mou, Murari Guragain**

1. **Introduction**

In the clinical trial of a small molecule targeting deadly breast cancer, thousands of patients’ health information records over the time after drug administration can help clinicians and scientists to determine the clinical drug efficacy in rescuing patients and improving the overall survival rate of patients. The clinical trial can be performed in different clinical sites in different countries by recruiting thousands of patients based on the guidelines of the clinical trial plan. In the clinical trial, clinicians gave patients the testing drug with different dosages at single dose injection or repeated dose injection at different time points, the data from those treatments is used to evaluate the biosafety of the testing compound and to calculate the pharmacokinetics, pharmacodynamics, and toxicology in patients' population. During the treatment, clinicians and scientists need to record the patients’ clinic-pathological features include age, site of breast with cancer, histologic types, histologic grade, pathological tumor size, pathological lymph node status, prognostic stage, and metastasis. By analyzing the pathological data, clinicians and scientists will compare the compound-treated group with the placebo-treated group and conclude whether the testing compound significantly improves patients’ survival rate and rescue breast cancer.

**Theory for clinical trial data management of cancer treatment:**

When the clinical trial starts, doctors will recruit patients based on the standards and guidelines that were included in the clinical trial plan during Investigational New Drug (IND) application. The recruited patients’ health and pathological data is recorded during the office visit before and after drug administration in the clinical trial. Using the data, the drug biosafety, pharmacokinetics, pharmacodynamics, and drug efficacy can be evaluated and compared between different dosages groups and the placebo group.

In our project, the recruited patients in the double-blind study clinical trial will have their unique ID number and will be divided into different groups: (1) placebo, low-dose, medium-dose, and high-dose group; (2) one single injection and repeated injection. In the whole study, the patients need to visit the office for routine health check and data collection. The physician needs to record the details of patient's health data, study starting date, office visiting date, ending date, drug injection date, and dose, etc.

The patient's health record include sex, race, age, weight, smoking status, alcohol using, current drug in use, blood pressure, cancer tissue size, biomarkers expression in blood samples, histological grade of breast cancer, cancer cells distribution during the treatment, and pathological lymph node status. For the patients that died in the clinical trial, a thorough investigation should be conducted and stored in the circumstances surrounding the death and report the incident to regulatory authorities if necessary.

**Other requirements:**

In this project, entities include patients, doctors, doctors’ offices, visits, clinical treatments, and lab results.

1. In the clinical trial, a **patient** can have one to multiple **doctors**; a patient must have at least one doctor in charge of the treatment, but doctors can take care of zero to multiple patients.
2. The **patient** can visit one to multiple **doctor’s offices** during the clinical trial; there are multiple doctor offices, the patients may visit different offices because of some other diseases including breast cancer. The doctor’s offices can host zero to multiple patients.
3. In the clinical trial, the **patient** can have one to multiple **visits** to the doctor’s office.
4. In the clinical trial, the **patient** can only take one **clinical treatment**. Although there are several clinical treatments in the clinical trial, the patient can only take one clinical design. The clinical treatment plans are offered to one to multiple patients in this clinical trial.
5. A **Doctor** can only work in one **doctor’s office** in the clinical trial and the **doctor’s office** can have one to multiple **doctors**.
6. A **doctor** can offer one to multiple **clinical treatments** to patients and the **clinical treatments** can be offered by one to multiple **doctors**.
7. A **doctor’s office** can give one to multiple **clinical treatments** to patients, and the **clinical treatments** can be provided by one to multiple **doctors’ offices**.
8. The **visits** of patients can have one to multiple of **doctors**, who will be responsible for the visits. The doctors can be responsible for zero to multiple visits from patients. The visits can happen in one to multiple **doctor’s offices**, and the doctor's offices can have zero to multiple visits.
9. A **patient** can have one to multiple **lab results**, one **doctor** can charge zero to multiple lab results, the **doctor’s office** can have zero to multiple lab results, and **clinical treatments** can have one to multiple lab results.

A **patient** will haveseveral attributes, including ID number, age, name, address, health records, pathological data, study starting date, office visiting date, ending date, drug injection date, and dose.

The **doctors** will have several attributes, including ID number, name, age, office address, specialization, phone Number, and email.

The **doctor's offices** will have several attributes, including office ID, name, address, zip code.

The **clinical treatments** include attributes: medication ID, medication name, dosage, start date, end date.

The **visits** include visit ID, patient ID, doctor’s ID, visit date, visit type, procedures conducted, and other visit records.

The **lab results** include lab result ID, patient ID, test name, test date, doctor’s name, the doctor’s office ID, and results data.

1. **Conceptual Data Modeling**

EER and UML diagrams:

A computer screen shot of a computer

Description automatically generatedSeveral many white ovals with black text

Description automatically generated with medium confidence

EER diagram UML diagram

**III. Mapping Conceptual Model to Relational Model**

**Relation Model:**

Primary Kay: Underlined Foreign Key: *Italicized*

Hospital (Hospital ID, Name, Phone, Country, City, Address, *Company ID*)

-Hospital ID: Primary key, Not Null.

-Company ID: Foreign key refers to Company ID in Sponsor, Not Null.

Sponsor (Company ID, Company Name, Address)

-Company ID: Primary key, Not Null.

Treatment (LOT ID, date, time, Treatment Categories, *Company ID*)

-LOT ID: Primary key, Not Null.

-Company ID: Foreign key refers to Company ID in Sponsor, Not Null.

LabTest (Test ID, date, TestName, Processed test data, Original test data, TestPhase, *LotID*)

-TestID: Primary key, Not Null.

-LotID: Foreign key refers to LOT ID in Treatment, Not null.

Patient (PatientID, DOB, Phone, Name, Email, Address, Gender)

-PatientID: Primary key, Not Null.

Physician (Physician ID, Physician Name, PhysicianEmail, *Hospital ID, Company ID*)

-Physician ID: Primary key, Not Null.

-Company ID: Foreign key refers to Company ID in Sponsor, Not Null.

-Hospital ID: Foreign key refers to Hospital ID in Hospital, Not Null.

Patient\_Physician (*Physician ID*, *Patient ID*)

-Physician ID: Foreign key refers to Physician ID in Physician, Not Null.

-PatientID: Foreign key refers to Patient ID in Patient, Not Null.

LOG (*Patient ID*, *Physician ID*, *LOT ID*)

-Physician ID: Foreign key refers to Physician ID in Physician, Not Null.

-PatientID: Foreign key refers to Patient ID in Patient, Not Null.

-LOT ID: Foreign key refers to LOT ID in Treatment, Not null.

Nurse (NurseID, Nurse Name, NurseEmail, Nurse Phone, *Hospital ID*)

-Nurse ID : Primary key, Not Null.

-Hospital ID: Foreign key refers to Hospital ID in Hospital, Not Null.

Nurse\_Patient (*NurseID*, *PatientID*)

-Nurse ID: Foreign key refers to Nurse ID in Nurse, Not Null.

-PatientID: Foreign key refers to Patient ID in Patient, Not Null.

Physician\_Sponsor (*Physician ID*, *Company ID*)

-Physician ID: Foreign key refers to Physician ID in Physician, Not Null.

-Company ID: Foreign key refers to Company ID in Sponsor, Not Null.

Drug\_Usage\_Surgery\_History (RecordID, Date, RecordName, Duration, Dose, RecordCategory)

- RecordID: Primary key, Not Null

Patient\_ Drug\_Usage\_Surgery\_History (*PatientID*, *RecordID*)

-PatientID: Foreign key refers to Patient ID in Patient, Not Null.

- RecordID: Foreign key refers to Record ID in Drug\_Usage\_Surgery\_History, Not Null.

OfficeVisit (Office\_ID, VisitDate, VisitTime, Reason, Visit Categories)

-Office\_ID: Primary key, Not Null.

Patient\_OfficeVisit (*Patient ID*, *Office ID*)

-PatientID: Foreign key refers to Patient ID in Patient, Not Null.

-Office\_ID: Foreign key refers to Office ID in OfficeVisit, Not Null.

OfficeVisit\_LabTest (*Office ID, Test ID*)

-Office\_ID: Foreign key refers to Office ID in OfficeVisit, Not Null.

-TestID: Foreign key refers to TestID in LabTest, Not Null.

A screenshot of a computer code

Description automatically generated**IV. Implementation of Relation Model via MySQL and NoSQL**

**MySQL Implementation:**

Here is the screenshot of

DDLcommand to create Database.

1. A screenshot of a computer

   Description automatically generated**The following query is to COUNT**

**DOSE = 60 in the drug\_use\_surgery\_history Table.**

Select count (\*) from clinical\_database.

**A screenshot of a computer

Description automatically generated**drug\_usage\_surgery\_history where dose =60;

1. **The following query is to INSERT**

**a dataset in the drug\_use\_surgery\_history Table.**

Select \* from clinical\_database.drug\_usage\_surgery\_history;

insert into clinical\_database.drug\_usage\_surgery\_history

(RECORDID, RECORDDATE, RECORDNAME,

DURATION, DOSE, RECORDCATAGORY)

VALUES (11111, ‘2024-07-13’, ‘SONY’, 24, 22, ‘PAIN’);

1. **A screenshot of a computer

   Description automatically generatedThe following DELETE query can be used to**

**delete dataset from the Table drug\_use\_surgery\_history.**

Select \* from

clinical\_database.drug\_usage\_surgery\_history;

delete \* from

clinical\_database.drug\_usage\_surgery\_history

where recordid = 111111;

**4**. **To find the Nurse information, who take care of PatientID= '116041', '149949', '790792'**

**For patientID=149949 and 790792, multiple nurses took care of them. One nurse took care of PatientID= 116041 patient**.

A screenshot of a computer

Description automatically generatedSelect \* from nurse join nursepatent on

nurse.nurseid = nursepatent.nurseid

where nursepatent.patientid in

(‘116440’, ‘149949’, ‘790792’);

**5**.**To find what drugs and how many doses the patientID = '116041', '149949', '790792' had in the record. Who took the most of doses?**

A screenshot of a computer

Description automatically generatedSelect \* from drug\_usage\_surgery\_history

join patient\_drug\_usage\_history

on drug\_usage\_surgery\_history.recordid =

patient\_drug\_surgery\_history.recordid

where patient\_drug\_usage\_history.patientid in

(‘116441’, ‘149949’, ‘790792’);

6. **The following query is to retrieve the patient information, who had the maximum dose in the record**.

Select p.\* from patient p join patient\_drug\_usage\_history pdsh on p.patientid = pdsh.patientid

join drug\_usage\_surgery\_history d on pdsh.recordid = d.recordid where d.dose = (select max(d1.dose) from drug\_usage\_surgery\_history d1 where d1.recordid);

A screenshot of a phone

Description automatically generated

A screenshot of a phone

Description automatically generated7. **Simple Query the patientname, and patientaddress of all patients having the string camp as part of their address**.

Select patientname, patientaddress

from patient where patientaddress

A screenshot of a computer

Description automatically generatedlike ‘%Camp%’;

8. **This query retrieves lotid and treatmentcatagories**

**all from treatments in descending order**.

select lotid, treatmentcatagories,

date from treatments

A screenshot of a computer

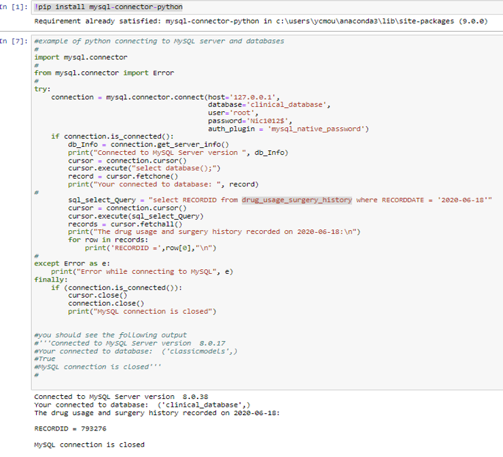
Description automatically generatedorder by date desc;

9**.This query retrieves hospitalname**

**and hospitalcity from the hospital.**

Select hospitalcity, hospitalname

from hospital group by hospitalcity;



Here is the process to access the MySQL database using

Python. Here is the code and screenshot that we used Jupyter

Notebook to access my “clinical\_database”. Using the

python, we retrieved the RECORDID that was recorded

on ‘2020-06-18’.in the table ‘drug\_usage\_surgery\_history’.

Here is the code and screenshot that

**A screenshot of a computer program

Description automatically generated**shows we can retrieve database by Python.

Then, we tried whether we could

plot the patient gender in a

bar graph using Python.

The following is the code

and the result screenshot.

**A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generatedHere is the result:**

**The next, we want to use a pie chart to show the percentage of different doses ('0-10', '10-20','20-30', '30-40','40-50','50-60','60-70','70-80','80-90','90-100') in drug\_usage\_surgery\_history table.**

A screenshot of a computer

Description automatically generatedPie chart Result**:**

**A screenshot of a graph

Description automatically generated**

In the following case, we hope to count the number of patients from different countries, there is not a column of patientcountry, but we can count the number of patients by link several tables, including physiciansponsor, phy\_pat, hospitcal by JOINING. Here is the Python code and the scatter results on the number of patients from different countries.

****A graph with blue dots and white lines

Description automatically generatedScatter results

NoSQL Implementation:

**First, we inserted the data in the MongoDB database by importing the same data that we used for MySQL. Below is the screenshot that we generated to show that the data has been imported successfully. One collection is “drug\_usage-surgery\_history” and the other is the collection of “nurse”.**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

If we just want to **show only the “NURSEID”**

**and “NURSENAME”,**

**and sort the table in**

**ascending order**.

Here is the result.

**To aggregate and group the number of patients that a nurse took care of, then sort the total number of patients in descending order. Below is the query and the screenshot of the query.**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generateddb.getCollection('nursepatient').aggregate( [ {

$group: { \_id: '$NURSEID',

totalPatients: { $sum: 1 }} },

{ $sort: { totalPatients: -1 } }],

{ maxTimeMS: 60000, allowDiskUse: true });

**If we want to write a query without using a map-reduce pipeline to get the total dose for all recordeds and average dose of records in the collection of drug\_usage\_surgery\_history.**

The total dose: 5162, and average dose: 51.62. Here is the query:

A screenshot of a computer

Description automatically generateddb.getCollection('drug\_usage\_surgery\_history'

).aggregate([{ $group: {\_id: null, totalDose:

{ $sum: '$DOSE' }, totalCount:

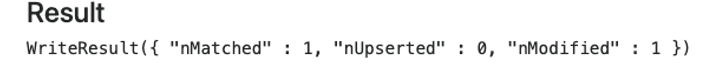
{ $sum: 1 } }, { $project: { \_id: 0,

averageDose: {$divide: ['$totalDose',

'$totalCount']} } } ], { maxTimeMS:

60000, allowDiskUse: true });

**Here are two screenshots to show an update query which updates hospitalname to LexingClinic**.

db.hospitals.update({“\_id”: ‘12557171435’},

{$set:{“hospitalname”:”LexingClinic”}});



Db.hospitals.find(“\_id”:’12557171435’});

**VII. Summary and Recommendation**

The Clinical Trial Data Management of Breast Cancer Treatment designed on MySQL is a relational database that can be used to track the patient’s health data. It will help the company to track and analyze the drug’s efficacy by collecting and analyzing all the relevant information. In this report, we used Python to retrieve the data and used several graphs to visualize the patient’s information.

The data for the clinical database management has been generated randomly, therefore the data quality needs to be improved in this case.