

Spring 2020: Guillaume Faddoul

ISYS 864: Data Management for Analytics.

Final Report - Synthetic Patient Data.

Team :

- **Aditya Tamhankar**
- **Hema Vivekanandan**
- **Siddhant Navaratna**

Introduction:

Name: Synthetic Patient Data in OMOP

Business Sector: HealthCare and Pharmaceuticals

Description:

It's a drug observation system, that comprises of multiple entities, of which we will be using the following:

- Condition Occurrence
- Provider
- Drug Exposure
- Person
- Location
- Observation
- Drug

Data Source :

www.cms.gov
([Center of Medicare and Medicaid](#))

This data was sourced from the CDC, it is a set of real time observations to study the drug interactions under various conditions.

Reason for Picking this Concept:

Building this database system would help us get a good understanding of how HealthCare and Pharmaceutical data is structured and how the database is maintained.

Description of Entities:

Table Name: Conditions

Contains details about the conditions about the conditions and drugs used to treat them.

Attributes: ConditionID, ConditionType, DrugID

Table Name: Provider

Contains details about providers for the care sites.

Attributes: ProviderID, CareSiteID, ProviderName, YearOfBirth, PhoneNumber, ZipCode

Table Name: Person

Contains details about the patients conditions and care site and other details.

Attributes: PersonID, ProviderID, CareSiteID, ConditionID, Name, DOB, Gender

Table Name: Drug_Exposure

Contains details about treatment history of the patient for a condition.

Attributes: DrugExposureID, PersonID, ProviderID, ConditionID, DrugExposureStartDate, DrugExposureEndDate, StopReason, Refills, Quantity, DaysofSupply

Table Name: Care_site

Contains details about the care site type and location.

Attributes: CareSiteID, CareSiteName, ProviderID, PlaceOfService, ZipCode, DateOfService

Table Name: Observation

Contains details about the patient's diagnosis.

Attributes: ObservationID, PersonID, ConditionID, ObservationDate, ObservationTime, ObservationType

Table Name: Drug

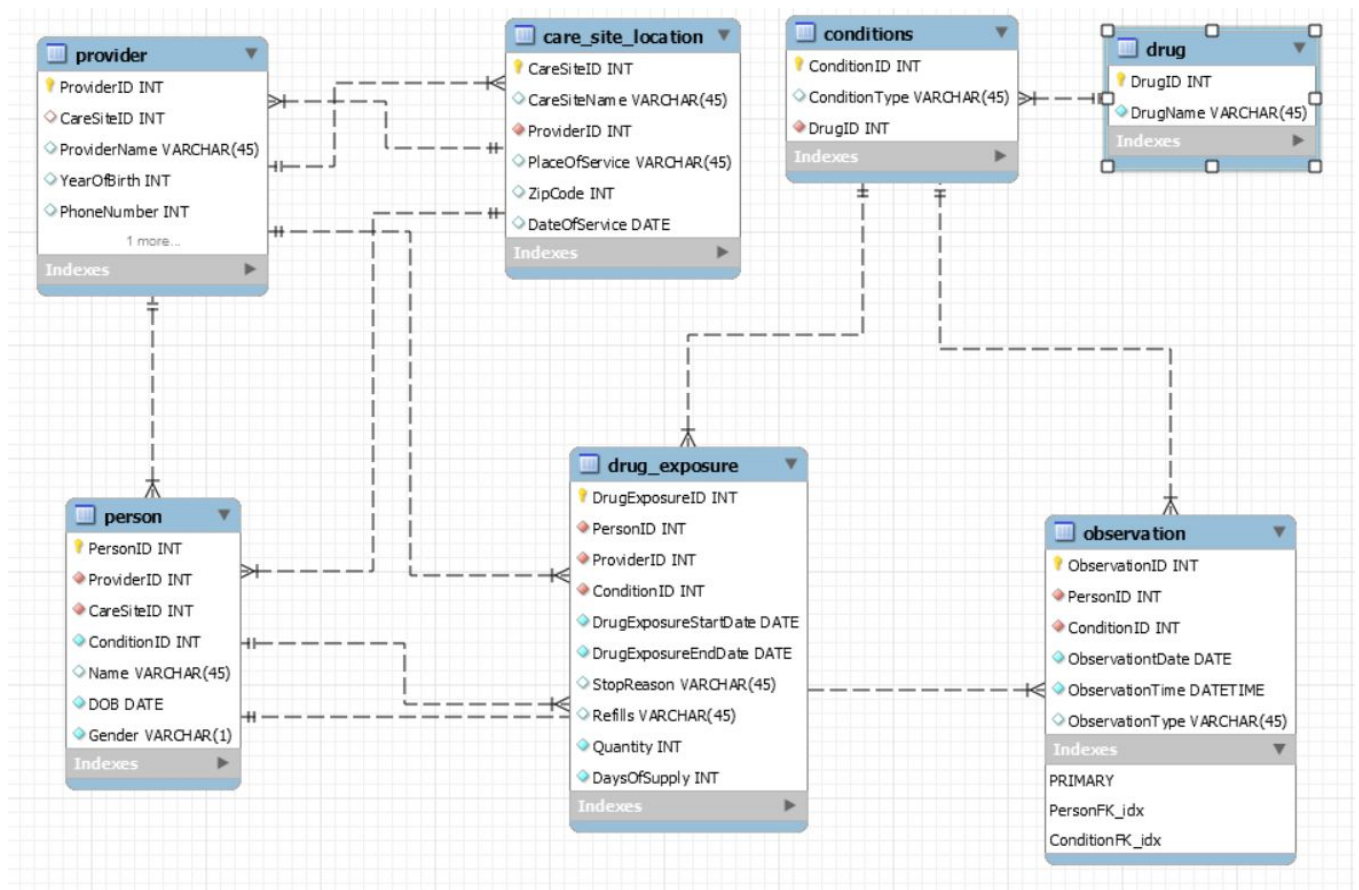
Contains details about the drug.

Attributes: DrugID, DrugName

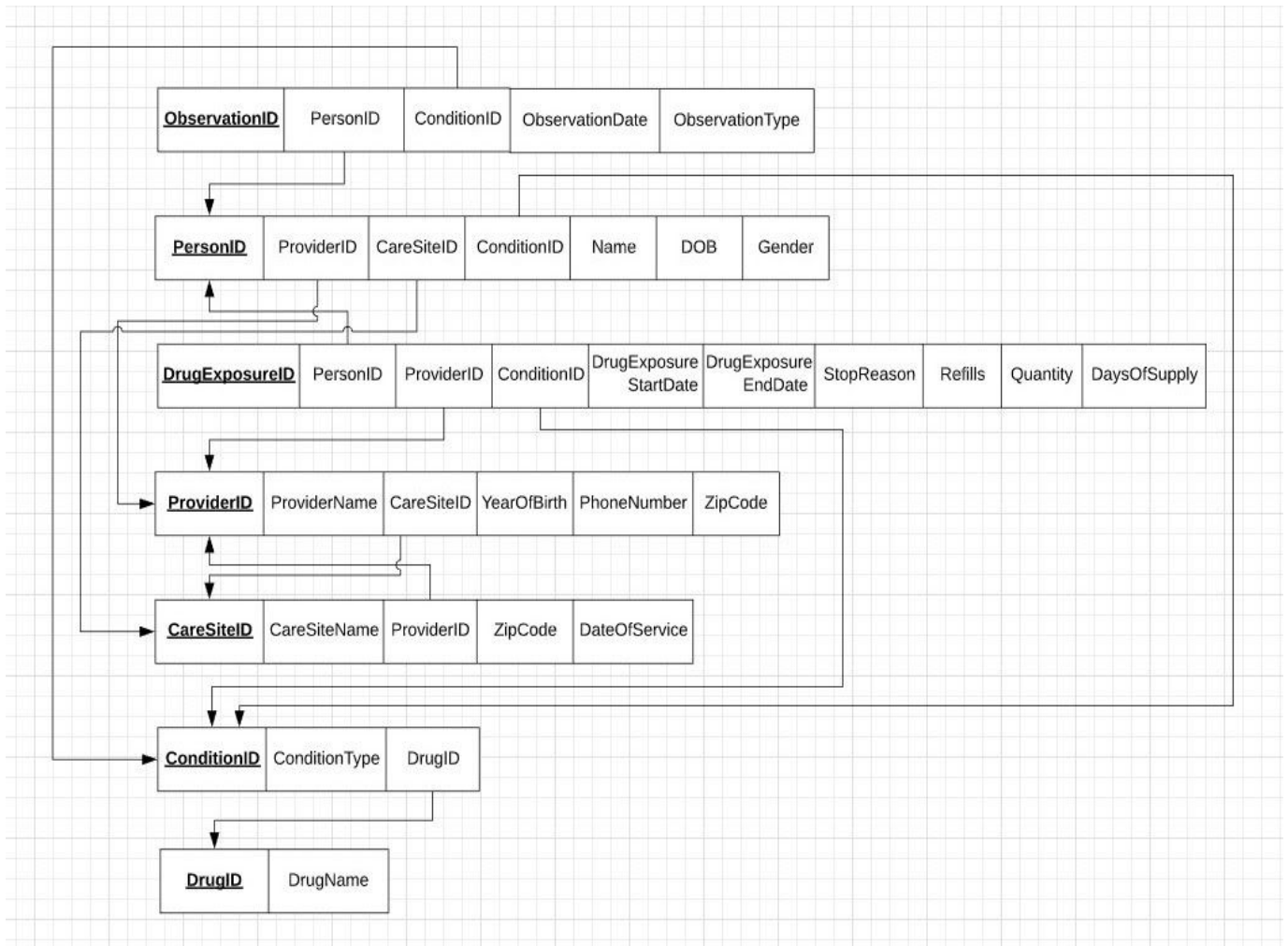
Business Rules:

- Each condition can have one and only one drug.
- One person can have multiple conditions and multiple people can have the same condition.
- One observation can have a maximum of one person and one condition.
- Each care site has at least one provider.
- A provider may or may not have a care site.

ER Diagram:



3NF Relational Model:



- The data that we acquired was already in 3NF.
- The name of the drugs administered for the diseases were missing from the data, so we created an additional 'drug' table for it.

SQL Create Queries:

1. CREATE TABLE IF NOT EXISTS `care_site_location` (
 `CareSiteID` INT NOT NULL,
 `CareSiteName` VARCHAR(45) NULL DEFAULT NULL,
 `ProviderID` INT NOT NULL,
 `ZipCode` INT NULL DEFAULT NULL,
 `DateOfService` DATE NULL DEFAULT NULL,
 PRIMARY KEY (`CareSiteID`));
2. CREATE TABLE IF NOT EXISTS `provider` (
 `ProviderID` INT NOT NULL,
 `ProviderName` VARCHAR(45) NULL DEFAULT NULL,
 `CareSiteID` INT NULL DEFAULT NULL,
 `YearOfBirth` INT NULL DEFAULT NULL,
 `PhoneNumber` INT NULL DEFAULT NULL,
 `ZipCode` INT NULL DEFAULT NULL,
 PRIMARY KEY (`ProviderID`),
 INDEX `CareSiteFK_idx` (`CareSiteID` ASC) VISIBLE,
 CONSTRAINT `CareSiteFK`
 FOREIGN KEY (`CareSiteID`)
 REFERENCES `care_site_location` (`CareSiteID`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION);
3. CREATE TABLE IF NOT EXISTS `drug` (
 `DrugID` BIGINT NOT NULL,
 `DrugName` VARCHAR(45) NOT NULL,
 PRIMARY KEY (`DrugID`));
4. CREATE TABLE IF NOT EXISTS `conditions` (
 `ConditionID` INT NOT NULL,
 `ConditionType` VARCHAR(45) NULL DEFAULT NULL,
 `DrugID` BIGINT NOT NULL,
 PRIMARY KEY (`ConditionID`),
 INDEX `DrugFk_idx` (`DrugID` ASC) VISIBLE,
 CONSTRAINT `DrugFk`
 FOREIGN KEY (`DrugID`)
 REFERENCES `drug` (`DrugID`)

```
ON DELETE NO ACTION  
ON UPDATE NO ACTION);
```

```
5. CREATE TABLE IF NOT EXISTS `person` (  
  `PersonID` INT NOT NULL,  
  `ProviderID` INT NOT NULL,  
  `CareSiteID` INT NOT NULL,  
  `ConditionID` INT NOT NULL,  
  `Name` VARCHAR(45) NULL DEFAULT NULL,  
  `DOB` DATE NOT NULL,  
  `Gender` VARCHAR(5) NOT NULL,  
  PRIMARY KEY (`PersonID`),  
  INDEX `ProviderFK_idx` (`ProviderID` ASC) VISIBLE,  
  INDEX `CareSiteFK_idx` (`CareSiteID` ASC) VISIBLE,  
  CONSTRAINT `ProviderFK`  
    FOREIGN KEY (`ProviderID`)  
    REFERENCES `provider` (`ProviderID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `CareSiteFK1`  
    FOREIGN KEY (`CareSiteID`)  
    REFERENCES `care_site_location` (`CareSiteID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `ConditionsFK2`  
    FOREIGN KEY (`ConditionID`)  
    REFERENCES `conditions` (`ConditionID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

```
6. CREATE TABLE IF NOT EXISTS `drug_exposure` (  
  `DrugExposureID` INT NOT NULL,  
  `PersonID` INT NOT NULL,  
  `ProviderID` INT NOT NULL,  
  `ConditionID` INT NOT NULL,  
  `DrugExposureStartDate` DATE NOT NULL,  
  `DrugExposureEndDate` DATE NOT NULL,  
  `StopReason` VARCHAR(45) NULL DEFAULT NULL,
```

```
`Refills` VARCHAR(45) NULL DEFAULT NULL,  
`Quantity` INT NOT NULL,  
`DaysOfSupply` INT NOT NULL,  
PRIMARY KEY (`DrugExposureID`),  
INDEX `PersonFK_idx` (`PersonID` ASC) VISIBLE,  
INDEX `ProviderFK_idx` (`ProviderID` ASC) VISIBLE,  
INDEX `ConditionFK_idx` (`ConditionID` ASC) VISIBLE,  
CONSTRAINT `PersonFK`  
  FOREIGN KEY (`PersonID`)  
    REFERENCES `person` (`PersonID`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
CONSTRAINT `ProviderFK1`  
  FOREIGN KEY (`ProviderID`)  
    REFERENCES `provider` (`ProviderID`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
CONSTRAINT `ConditionsFK`  
  FOREIGN KEY (`ConditionID`)  
    REFERENCES `conditions` (`ConditionID`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION);
```

7. CREATE TABLE IF NOT EXISTS `observation` (
 `ObservationID` INT NOT NULL,
 `PersonID` INT NOT NULL,
 `ConditionID` INT NOT NULL,
 `ObservationDate` DATE NOT NULL,
 `ObservationType` VARCHAR(45) NULL DEFAULT NULL,
 PRIMARY KEY (`ObservationID`),
 INDEX `PersonFK_idx` (`PersonID` ASC) VISIBLE,
 INDEX `ConditionsFK_idx` (`ConditionID` ASC) VISIBLE,
 CONSTRAINT `PersonFK1`
 FOREIGN KEY (`PersonID`)
 REFERENCES `person` (`PersonID`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION,
 CONSTRAINT `ConditionsFK1`
 FOREIGN KEY (`ConditionID`)

```
REFERENCES `conditions` (`ConditionID`)
ON DELETE NO ACTION
ON UPDATE NO ACTION);
```

SQL Insert Queries:

1. INSERT INTO `care_site_location` (`CareSiteID`, `CareSiteName`, `ProviderID`, `ZipCode`, `DateOfService`) VALUES (1, 'Inpatient Facility', 519, 98074, '2016-11-23');
2. INSERT INTO `provider` (`ProviderID`, `ProviderName`, `CareSiteID`, `YearOfBirth`, `PhoneNumber`, `ZipCode`) VALUES (500, 'Provider1', 1, 1962, 422963402, 96343);
3. INSERT INTO `drug` (`DrugID`, `DrugName`) VALUES (893547593, 'Orencia');
4. INSERT INTO `conditions` (`ConditionID`, `ConditionType`, `DrugID`) VALUES (453453454, 'Rheumatoid Arthritis', 893547593);
5. INSERT INTO `person` (`PersonID`, `ProviderID`, `CareSiteID`, `ConditionID`, `Name`, `DOB`, `Gender`) VALUES (1, 519, 1, 453453454, 'Jack', '1923-05-01', 'F');
6. INSERT INTO `drug_exposure` (`DrugExposureID`, `PersonID`, `ProviderID`, `ConditionID`, `DrugExposureStartDate`, `DrugExposureEndDate`, `StopReason`, `Refills`, `Quantity`, `DaysOfSupply`) VALUES (9, 1, 511, 453453858, '2016-11-25', '2017-08-03', 'Ineffective', '26', 93, 46);
7. INSERT INTO `observation` (`ObservationID`, `PersonID`, `ConditionID`, `ObservationDate`, `ObservationType`) VALUES (1, 9, 453453593, '2018-07-16', 'Recheck');

SQL Queries:

Query 1: Display details of people suffering from Asthma.

```
select Name, DOB, CareSiteID, Gender from person p
left join observation o
on p.PersonID = o.PersonID
join conditions c on p.ConditionID = c.ConditionID
where c.ConditionType = 'Asthma';
```



Result 1:

Result Grid   Filter Rows: <input type="text"/>				
	Name	DOB	CareSiteID	Gender
▶	Curtis	1922-07-01	7	Other

Query 2: Display details of Providers for Inpatient Facility.

```
select ProviderName, p1.CareSiteID, PhoneNumber from provider p1
join care_site_location c
on c.CareSiteID = p1.CareSiteID
where c.CareSiteName = 'Inpatient Facility';
```

Result 2:

Result Grid   Filter Rows: <input type="text"/>			
	ProviderName	CareSiteID	PhoneNumber
▶	Provider 1	1	422963402
	Provider 17	1	163506025
	Provider 25	1	435318290
	Provider 3	3	623375844
	Provider 18	3	793561003

Query 3: Display the count of various reasons for stopping medication.

```
select StopReason, count(*) from drug_exposure
group by StopReason;
```

Result 3:

Result Grid			Filter Rows:
	StopReason	count(*)	
▶	Recovered	2	
	Abused	3	
	Ineffective	3	
	Trial Period End	1	
	Placebo	1	

Query 4: List of person their condition and corresponding drug.

```
select Name, ConditionType, Drugname from person p
inner join conditions c on p.ConditionID=c.ConditionID
inner join drug d on c.DrugID=d.DrugID;
```



Result 4:

Result Grid				Filter Rows:
	Name	ConditionType	Drugname	
▶	Jack	Rheumatoid Arthritis	Orencia	
	Ryan	Bulimia Nervosa	Fluoxetine	
	Matt	ADHD	Amphetamine	
	Damon	Auto Immune DS1	Xeljanz	
	Angelina	Pulmonary Edema	Nitroglycerin	
	Matt	Depression	Aripiprazole	

Query 5: Display the treatment history of patients.

```
select Name, ConditionType, Drugname, Quantity, DaysOfSupply from person p
inner join conditions c on p.conditionID= c.conditionID
inner join drug_exposure de on p.PersonID=de.personID
inner join drug d on d.DrugID=c.DrugID
order by ConditionType, DaysOfSupply DESC;
```

Result 5:

<div> <div>Result Grid</div> <div>  Filter Rows: <input type="text"/> </div> <div> Export:  Wrap Cell Cont </div> </div>					
	Name	ConditionType	Drugname	Quantity	DaysOfSupply
▶	Matt	ADHD	Amphetamine	92	54
	Meg	Allergies(Food)	Cetirizine	84	51
	Ellyse	Allergies(Food)	Cetirizine	86	42
	Curtis	Asthma	Albuterol	120	69
	Damon	Auto Immune DS1	Xeljanz	76	46
	Ryan	Bulimia Nervosa	Fluoxetine	116	61
	Matt	Depression	Aripiprazole	107	47

Challenges:

Organizing the raw data into tables was a huge challenge. Also, since it is sensitive data we had lots of columns that were encoded with alternate values. Understanding the meaning and segregating it into tables took us over a week.

Additionally we had to research the drugs interaction details in order to do a sanity check after constructing the entire database structure.

Conclusion:

Overall it was a good learning experience, We gained better understanding of databases, knowledge about an organization's structure. We now have an in-depth understanding of the nature of data that would be present in the Pharmaceuticals industry.