CST 363: Databases – Project Database Design and Implementation

11x11 #1

January 25, 2022

Axel Saldana, Jake Horne, Justin Nguyen

Introduction

Our underlying purpose for this project is to be able to create a database from scratch, logically identify how each individual table relates to others around it, and discern information about the relational schemas through selected queries. We want to be able to use these created data tables to notice trends between our entities, as well as document individual data points with our entities (a singular patient or drug within the patient or drug category, respectively).

Our database is centered around a drugstore chain, and the entities that comprise this functional group: patients, doctors, pharmaceutical companies, drugs, prescriptions, supervisors, and pharmacies. Below, we illustrate the relationship between these entities as well as the attributes of these entities, illustrated in parentheses.

- A patient (patientID, age, name, SSN address) reports to a single primary doctor and can have multiple doctors and prescriptions.
- A doctor (doctorID, SSN, name, specialty, years of experience) can have multiple patients and is the person who prescribes the drug.
- A prescription (drugID, patientID, pharmacyID, doctorID, quantity, filled, rxnumber, date) is written by the doctor, is received by the patient, and is sent to the pharmacy to validate and order the drug.
- A pharmacy (pharmacyID, pharmacyName, pharmacyPhone, pharmacyAddress) has many drugs, and chooses which drug to sell based on the one being prescribed. They have a contract (end Date, startDate, pharmacyCompanyID, supervisor, pharmacyID, contract) with at least one pharmaceutical company.
- A drug (pharmacyCompanyID, drugId, drugTradeName, drugFormula) is sold by (price, pharmacyID, drugID) the pharmacy based on the corresponding prescription requested by a doctor for a patient.
- A pharmaceutical company (pharmacyCompanyID, companyName, phoneNumber) can have contracts with many companies.

With that understanding of the foundations of our database explained, we created a complementary entity relationship diagram (ER diagram), with primary keys and correct cardinality to define the number of occurrences in one entity which is associated with number of occurrences in another (ie many patients can have John Doe as their primary doctor). This gives a strong visual representation to fall back on as well as give a quick understanding of the scope of our database.

After creating such a diagram, we converted the ER diagram into a relational database schema, which was composed of “create” tables that in essence exist to be populated with our future drugs, patients, doctors and so forth. This will be the backbone of the database and will be manipulated further by our queries, which we later layout further in this documentation. In the future, we will populate these tables with a thousand patients, ten doctors, and five thousand prescriptions.

Our relational schema had to satisfy third normal form criteria, or 3NF. This entailed that the database satisfied:

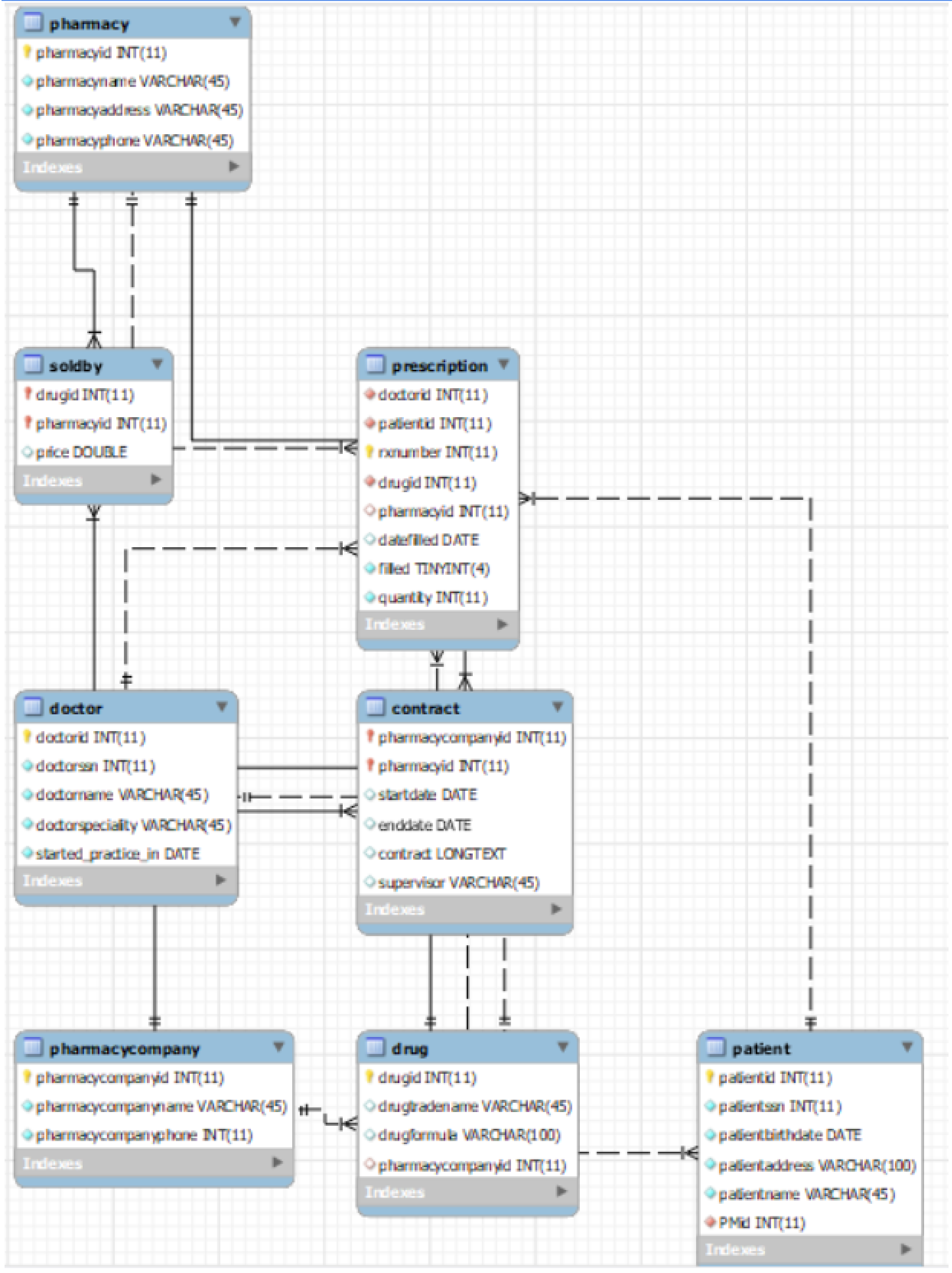
- First normal form (only atomic values, no repeating groups)
- Second normal form (non-key attributes fully dependent on primary key)
- No transitive functional dependency

3NF is typically used for relational database design since it removes insertion, update, and deletion anomalies which these operations would be conducted repetitively within our drugstore chain database; we want to have the freedom to manipulate and group data freely without being confined by anomalies by such actions. Within our database, we found no sign of any tables that deviated from the third normal form, but by checking we reinforced strong habits on proper database upkeep.

Finally, we created five SQL queries that we felt were most vital in the maintenance and further progression within this drugstore chain, both from a consumer and producer perspective. With the complexity of these queries, we hoped to gain more insight into our databases, potentially revealing new relationships that were previously glossed over.

This project will test our abilities to create logical data from scratch, as well as find meaningful interpretation of what relationships are existent. From this, we can make logical inferences on how different entities and attributes within entities correlate, and how we can exploit such trends to increase the revenues of our drugstore chain.

ER Diagram



pharmacy
pharmacyid INT(11)
pharmacyname VARCHAR(45)
pharmacyaddress VARCHAR(45)
pharmacyphone VARCHAR(45)
Indexes
soldby
drugid INT(11)
pharmacyid INT(11)
price DOUBLE
Indexes
prescription
doctorid INT(11)
patientid INT(11)
rxnumber INT(11)
drugid INT(11)
pharmacyid INT(11)
datefilled DATE
filled TINYINT(4)
quantity INT(11)
Indexes
doctor
doctorid INT(11)
doctorssn INT(11)
doctorname VARCHAR(45)
doctorspeciality VARCHAR(45)
started_practice_in DATE
Indexes
contract
pharmacycompanyid INT(11)
pharmacyid INT(11)
startdate DATE
enddate DATE
contract LONGTEXT
supervisor VARCHAR(45)
Indexes
pharmacycompany
pharmacycompanyid INT(11)
pharmacycompanyname VARCHAR(45)
pharmacycompanyphone INT(11)
Indexes
drug
drugid INT(11)
drugtradename VARCHAR(45)
drugformula VARCHAR(100)
pharmacycompanyid INT(11)
Indexes
patient
patientid INT(11)
patientssn INT(11)
patientbirthdate DATE
patientaddress VARCHAR(100)
patientname VARCHAR(45)
PMid INT(11)
Indexes

This ER diagram sets up the relationship between different entities, and the attributes that are associated with each entity. For relationships between entities, there are two indicators:

- Multiplicity-maximum number of times an instance of one entity can be associated with instances of another entity (shown by Crow's feet for many or double vertical line for one)
- Minimum number of times one instance can be related to others (shown by vertical line for mandatory and oval for optional)

For all ER diagrams, the symbol for multiplicity comes first, and the symbol indicating whether the relationship is mandatory or optional comes second. Take, for example, the relationship between a drug and prescription. Drug has a double vertical line and prescription has an oval and singular line. This reads as "a prescription needs to have one drug associated with it" as well as "a drug can be associated with a prescription".

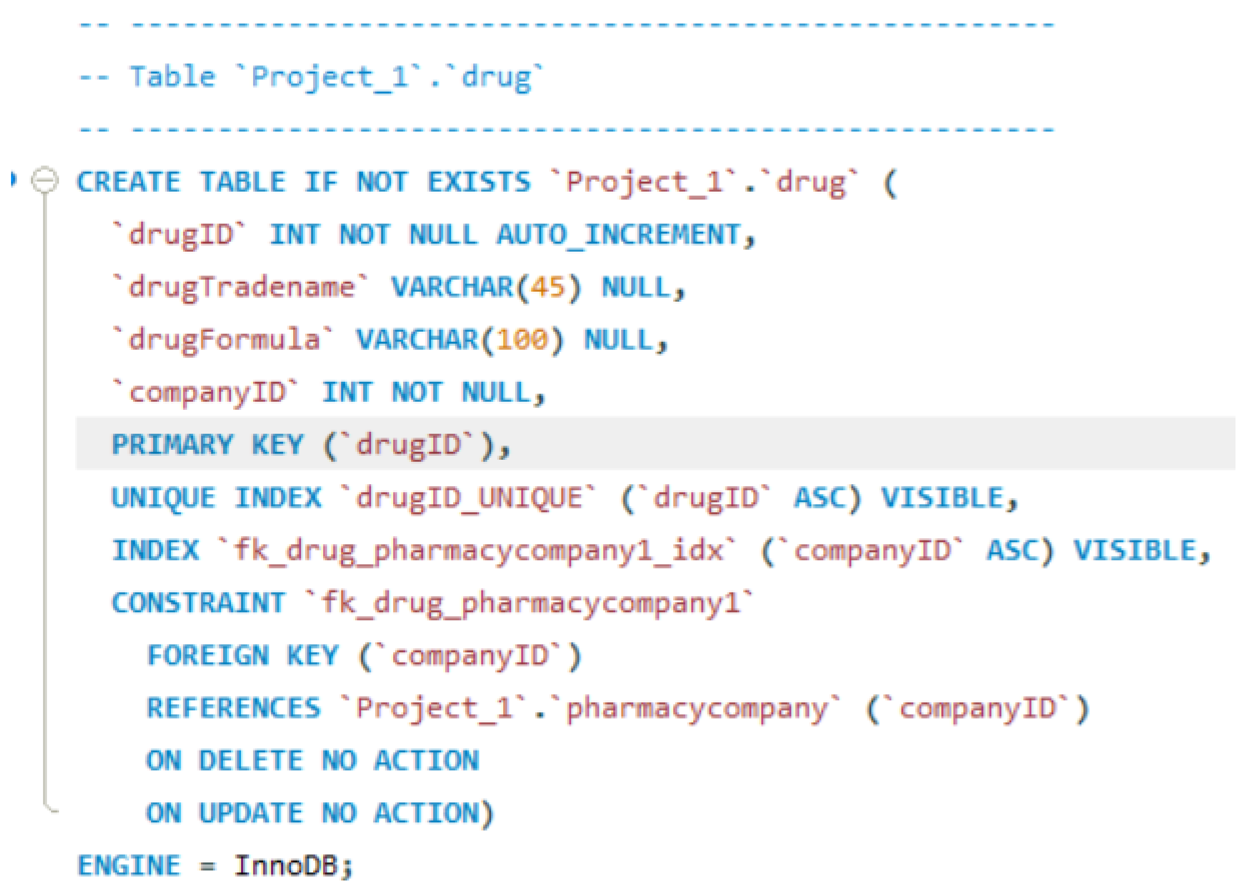
For relationships between entities, we can have a strong or weak relationship, weak being that the entity's existence is independent of other entities and the primary key of the child does not contain primary key components of the parent entity and strong having the attributes that the child entity's existence depends on that of the parent.

Consider the case of a prescription and a drug. A prescription has a strong relationship with a drug, since without a drug's existence, a prescription for that drug cannot exist. However, a drug can exist without a prescription, since you can get over the counter drugs that do not require doctor's approval and ordering to be acquired. All of our entities exhibit strong relationships minus doctor-patient, since they are all dependent on each other, without a doctor there is no prescription, no prescription there are no drugs for patients, no pharmacies then no pharmacy companies, and so on. With doctor-patient, there does not need to be a patient for every doctor, there is a situation where a new doctor may not have patients initially. However, we do have to have a primary doctor for every patient, per the guidelines and jurisdiction of this project.

A primary key is a column or a set of columns in a table whose values uniquely identify a row in the table. A relational database is designed to enforce the uniqueness of primary keys by allowing only one row with a given primary key value in a table. A foreign key is a column or a set of columns in a table whose values correspond to the values of the primary key in another table. In order to add a row with a given foreign key value, there must exist a row in the related table with the same primary key value. THis fact is more evident in the relational schema derived from the ER model part of our explanation, but simply put the foreign key is not in the entity's table but directly correlates to a column in another table that it is directly related to.

Relational Schema Derived from the ER Model

For each of our entities, we created tables and defined their attributes as having "null" or "not null" values. All identification values could not be null, as were names, addresses, ages and phone numbers; null values were used on attributes that could have a value, but at the time of table creation we did not know what that value was. For instance, we know the phone number of a pharmacy on creation of the tables, but the chemical formula of a drug could be an integer, a varchar, tinyint, we do not know at the time of creation.



```
-- -----------------------------------------------------
-- Table `Project_1`.`drug`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `Project_1`.`drug` (
  `drugID` INT NOT NULL AUTO_INCREMENT,
  `drugTradename` VARCHAR(45) NULL,
  `drugFormula` VARCHAR(100) NULL,
  `companyID` INT NOT NULL,
  PRIMARY KEY (`drugID`),
  UNIQUE INDEX `drugID_UNIQUE` (`drugID` ASC) VISIBLE,
  INDEX `fk_drug_pharmacycompany1_idx` (`companyID` ASC) VISIBLE,
  CONSTRAINT `fk_drug_pharmacycompany1`
    FOREIGN KEY (`companyID`)
    REFERENCES `Project_1`.`pharmacycompany` (`companyID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Take for example, our drug table on create. Our drugID cannot be null since we know it is of an integer value on table creation, and we auto increment the ID by one every time we insert a new drug into the table. DrugTradename and drugFormula were given an arbitrary upper limit of how many characters are allowed to identify each, set to 45 characters and 100 respectively. CompanyID is an integer that cannot be null since we know it has a value and know that it is of the int type.

Other inferences that we are making is that years of experience for doctor’s has to be an integer between zero and less than the age of the doctor (realistically, age minus total schooling since it can be inferred that they would not be practicing while still in academia). Price and quantity of drugs has to be an integer greater than 0, since you cannot have a decimal amount of a drug as well as all quantities have to be of positive magnitude, otherwise we could have a situation where a patient owes a pharmaceutical company an amount of a drug (negative quantity amount). Any phone number has to have a form of three digit area code followed by a seven digit phone number, to remove any extraneous characters that could be introduced. Finally, filled is a tinyint since values are only 0 or 1, corresponding to false or true for a boolean statement, respectively.

We use indexes to eliminate rows from consideration when selecting and making them explicitly visible to the user, since we want all pertinent data readily available for manipulation and relational comparisons. For any constraints we had, we wanted to make sure that tables that accessed foreign keys of related tables were not able to update or delete selected data. If it were allowed to alter data input that was shared, referencing and indexing rows and columns that were subject to change would be unwieldy, and would undo the beneficial effects of having the data in third normal form, where the purpose is for us to avoid such complications.

SQL Queries

The goal that we hope to achieve with our database is how we can increase sales within pharmacies and also incite more people to want to come into pharmacies based on prevalent needs in the area. What can the store do to push sales further is what we hope to answer. We will be looking at current revenue as well as traffic flow within each store, seeing which products are of higher demand than others. We also will look at the demographic of people coming into stores by looking at key ages (26 is when adults can no longer be covered by a guardian's insurance and 65 is typically when health conditions and insurance increases as the body ages), and curtailing our products to the largest demands.

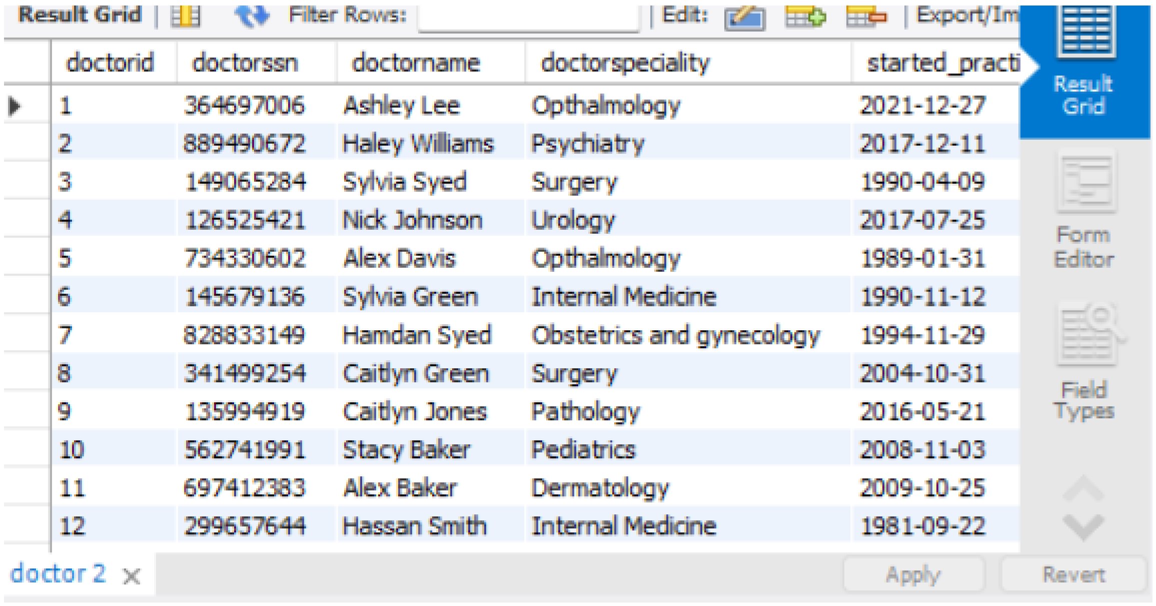
Since our overarching goal is to create more revenue through attraction of new customers and increased quantities for high demand drugs, having a network of pharmacies that are aware of the products that they themselves carry that other pharmacies also carry could prove beneficial. For one, having pharmacies carry the same drug in close locational proximity is detrimental to both stores' maximal returns since they are in direct competition with one another. However, on the flip side, if a pharmacy runs out of a high visibility product, it is useful to know which pharmacies are close enough to ship additional supplies of the drug so as to not lose sales at that specific location. This balancing act of competition and collaboration, although complex initially, can be touched on through our database by tracking which stores carry each drug. From there, we could look at local proximity and see if having a cluster of stores is affected by competition or if it in fact creates an oligopoly situation where pharmacies maximize their total revenue as a collective.

```
-- ----------------------------------------------------
-- Doctor names in alphabetical order grouped by speciality and their experience
-- ----------------------------------------------------
 SELECT doctorid, doctorname, doctorspeciality, doctorexperience
 FROM doctor
 GROUP BY doctorspeciality
 ORDER BY doctorspeciality and doctorname ASC;


--  -- ----------------------------------------------------
-- -- total amount of revenue made per store, Customers per store and average revenue per
customer
-- -- ----------------------------------------------------
SELECT s.drugid, s.pharmacyid, p.pharmacyid, p.drug_drugid,  SUM(s.price * p.filled) as
total_revenue
FROM soldby s, prescription p
WHERE s.drugid = p.drug_drugid AND s.pharmacyid = p.pharmacyid
```

```
GROUP BY p.pharmacyid;

--  -- ----------------------------------------------------
-- -- amount left of each drug in each store
-- -- ----------------------------------------------------
SELECT prescription.pharmacyid, prescription.drug_drugid,
(prescription.quantity -  prescription.filled) as amount_left
FROM  prescription
FULL JOIN pharmacy ON prescription.pharmacyid = pharmacy.pharmacyid
group by pharmacy.pharmacyname;

 --  -- ----------------------------------------------------
-- --  Similar drugs at different stores
-- -- ----------------------------------------------------
SELECT drug.drugtradename, drug.drugformula
FROM drug
FULL JOIN pharmacycompany on drug.pharmacycompanyid =
pharmacycompany.pharmacycompanyid
ORDER BY drug.drugformula;
 --  -- ----------------------------------------------------
-- -- how many are age 26 and 65+
-- -- ----------------------------------------------------
SELECT patientage
from patient
WHERE patientage = 26 and patientage >= 65;

 --  -- ----------------------------------------------------
-- -- Which prescription is getting prescriped the most
-- -- ----------------------------------------------------
SELECT prescription.filled
FROM prescription
join drug ON prescription.drug_drugid = drug.drugid
GROUP BY  drugtradename;
 --  -- ----------------------------------------------------
-- -- Average contract date
-- -- ----------------------------------------------------
SELECT AVG(contract.enddate - contract.startdate) as average_contract
FROM contract
JOIN prescription on contract.pharmacyid = prescription.pharmacyid
ORDER BY average_contract DESC;
```

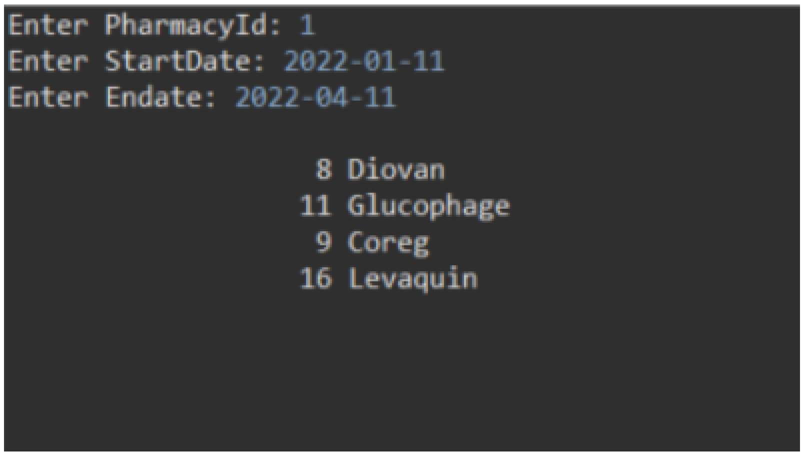
Screenshots

Manager Report:



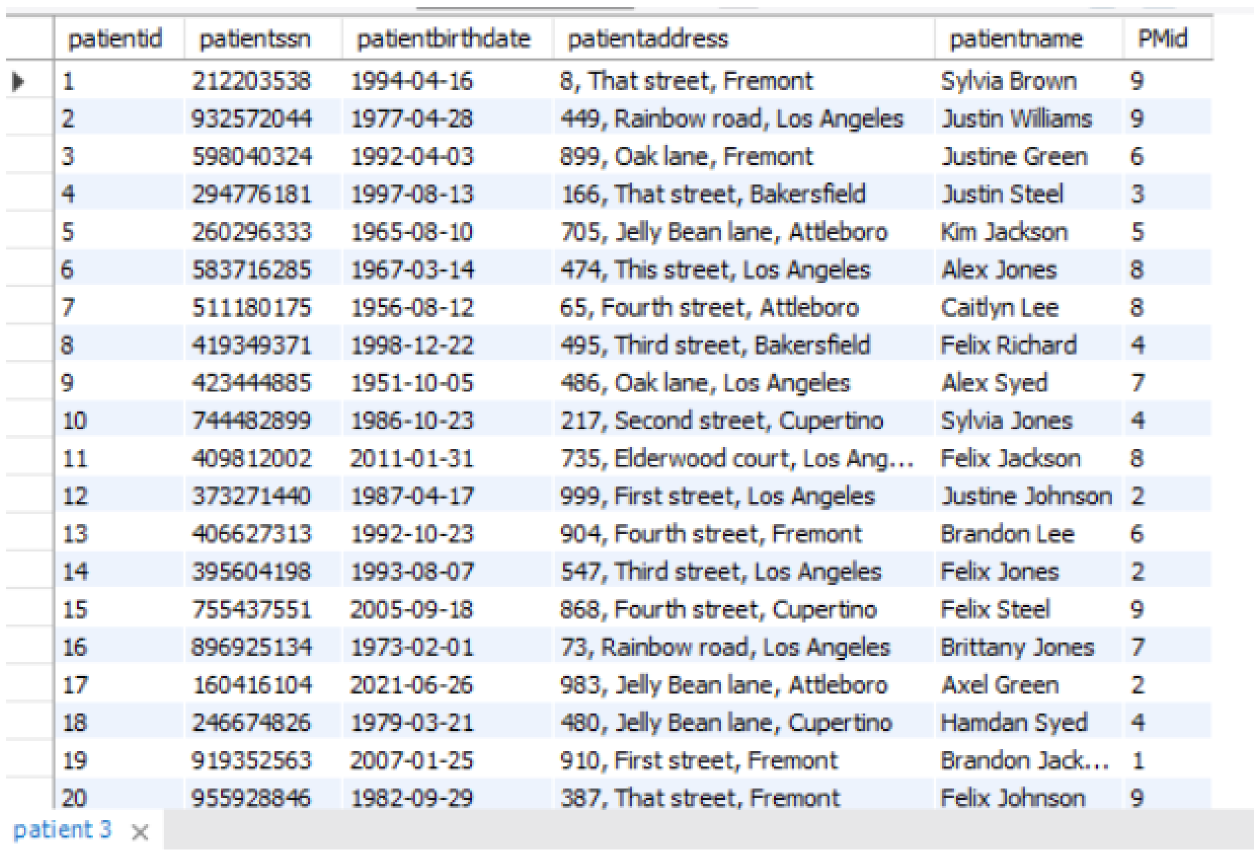
doctorid	doctorssn	doctorname	doctorspeciality	started_practi
1	364697006	Ashley Lee	Opthalmology	2021-12-27
2	889490672	Haley Williams	Psychiatry	2017-12-11
3	149065284	Sylvia Syed	Surgery	1990-04-09
4	126525421	Nick Johnson	Urology	2017-07-25
5	734330602	Alex Davis	Opthalmology	1989-01-31
6	145679136	Sylvia Green	Internal Medicine	1990-11-12
7	828833149	Hamdan Syed	Obstetrics and gynecology	1994-11-29
8	341499254	Caitlyn Green	Surgery	2004-10-31
9	135994919	Caitlyn Jones	Pathology	2016-05-21
10	562741991	Stacy Baker	Pediatrics	2008-11-03
11	697412383	Alex Baker	Dermatology	2009-10-25
12	299657644	Hassan Smith	Internal Medicine	1981-09-22

Random Generator:



```
Enter PharmacyId: 1
Enter StartDate: 2022-01-11
Enter Endate: 2022-04-11

                     8 Diovan
                    11 Glucophage
                     9 Coreg
                    16 Levaquin
```

Random Patient:



	patientid	patientssn	patientbirthdate	patientaddress	patientname	PMid
▶	1	212203538	1994-04-16	8, That street, Fremont	Sylvia Brown	9
	2	932572044	1977-04-28	449, Rainbow road, Los Angeles	Justin Williams	9
	3	598040324	1992-04-03	899, Oak lane, Fremont	Justine Green	6
	4	294776181	1997-08-13	166, That street, Bakersfield	Justin Steel	3
	5	260296333	1965-08-10	705, Jelly Bean lane, Attleboro	Kim Jackson	5
	6	583716285	1967-03-14	474, This street, Los Angeles	Alex Jones	8
	7	511180175	1956-08-12	65, Fourth street, Attleboro	Caitlyn Lee	8
	8	419349371	1998-12-22	495, Third street, Bakersfield	Felix Richard	4
	9	423444885	1951-10-05	486, Oak lane, Los Angeles	Alex Syed	7
	10	744482899	1986-10-23	217, Second street, Cupertino	Sylvia Jones	4
	11	409812002	2011-01-31	735, Elderwood court, Los Ang...	Felix Jackson	8
	12	373271440	1987-04-17	999, First street, Los Angeles	Justine Johnson	2
	13	406627313	1992-10-23	904, Fourth street, Fremont	Brandon Lee	6
	14	395604198	1993-08-07	547, Third street, Los Angeles	Felix Jones	2
	15	755437551	2005-09-18	868, Fourth street, Cupertino	Felix Steel	9
	16	896925134	1973-02-01	73, Rainbow road, Los Angeles	Brittany Jones	7
	17	160416104	2021-06-26	983, Jelly Bean lane, Attleboro	Axel Green	2
	18	246674826	1979-03-21	480, Jelly Bean lane, Cupertino	Hamdan Syed	4
	19	919352563	2007-01-25	910, First street, Fremont	Brandon Jack...	1
	20	955928846	1982-09-29	387, That street, Fremont	Felix Johnson	9

patient 3 ×

Conclusion

Our overarching purpose in this project is to create a database from scratch on a drugstore chain, composed of patients, doctors, pharmacies, drugs, pharmacy companies, contracts and prescriptions. Initially, we set up an entity relationship diagram to give a pictorial representation on the relationship between these entities and the attributes that are respectively associated with each entity. From there, we identified the strength of the relationship between its relationship as well as its relationship cardinality, identifying if it was a mandatory/optional and many/one relationship.

From this point, we converted this visual representation to a schema of tables that were initialized, but not yet populated. These tables will function as the backbone of our data manipulation and selection, so having them constrained on updating and deletion was crucial towards the integrity of the data that would eventually populate them. We established that the tables followed the third normal form, removing anomalies in insertion, update, and deletion actions to further this procedure of data normalization and integrity. Additionally, we established the indices we wanted to be visible when selecting data, so as to hide all non pertinent data from view.

Finally, manipulation of data is redundant without having a primary goal guiding the project. For our group, we identified revenue of the drugstore chain as the most tangible goal and set off to explore which factors had a positive externality on this effect: identifying which drugs were in the highest demands in their relative area, how to invite an oligopolistic situation where pharmacies in local regions collaborated towards maximizing net utility and profits, and identifying the age demographic of key ages that would be more dependent on drug prescription and by extension, regular visitors to pharmacies (26 years old is independent insurance and 65 is when insurance prices increase due to increase in health complications).

This project up to this point has taught SQL concepts, but more importantly goal creation. We as a group have to determine which questions we value in a vague rubric, giving us creativity in our analysis and responses that would be lost if a definitive goal was set for us prior. Going forward, we want to utilize our skills in SQL and data relationships further, to build on this knowledge and become adept in web application in the following part of this project.