

**Client Database Design Report: Easy Pharmacies**

KS Consulting Company

Keldin Maldonado, Samuel Ulloa

February 8, 2023

## **Client Requirements Overview**

Easy Pharmacies is a network of pharmacies that stand across the United States providing patients with pain free access to experienced doctors. Easy Pharmacies are able to fulfill prescriptions of many different drugs from many different drug manufacturers. Because of this, it is important that Easy Pharmacies has a robust and well throughout database for easy querying of not only customer data, but also relationships with drug manufacturers.

When a new patient visits one of Easy Pharmacies' locations, their information is entered into the Easy Pharmacies' database through an easy to fill interface. On the back end, patient information is uniquely identified through the patients social security number. Additional information that is required for the patient's file include their name, age, address, and who is their primary doctor.

Furthermore, because Easy Pharmacies has a list of doctors under their employment, a solution similar to the one mentioned above was needed. Easy Pharmacies' doctors are uniquely identified through their social security number. Additionally, the doctor's name is captured, the doctor's speciality, and how many years of experience the doctor has.

Apart from having to deal with patients and their on staff doctors, Easy Pharmacies also required a way of keeping track of their growing list of pharmacies across the country. Within the Easy Pharmacies database system, there exists a list of the names of the pharmacies, the addresses, and the phone number in case any information of a pharmacy needs to be quickly referenced.

Being that Easy Pharmacies is in the healthcare industry, many different types of drugs are prescribed and sold and their locations. The solution to this was designing three different sections where data can be entered to maintain the ability to combine data from different places.

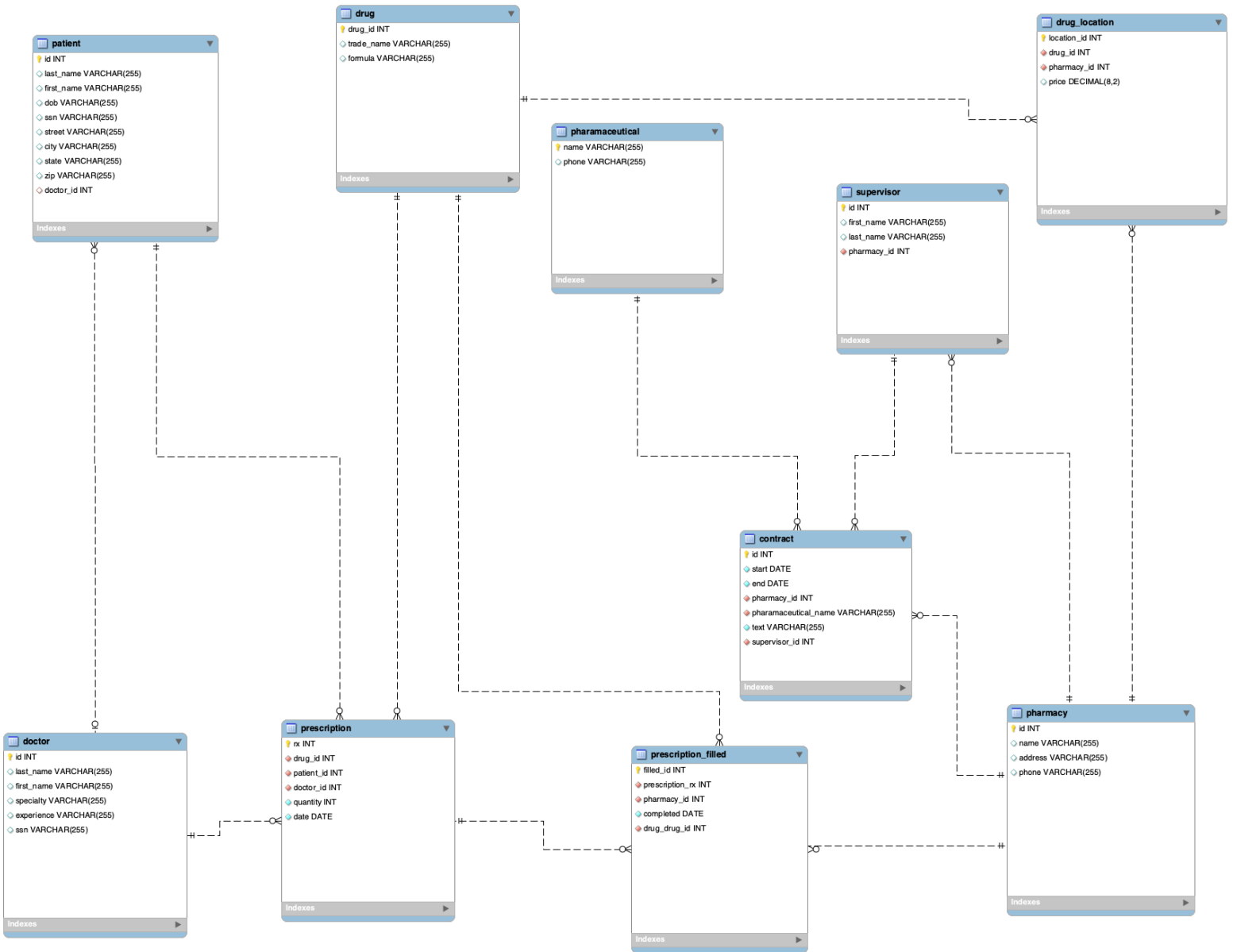
Each drug that is entered into the Easy Pharmacies system requires a unique id to easily identify it. On top of that, each drug requires the manufacturer of the drug, a generic drug name, and trade name. Because each drug listing requires the manufacturer information, the complete Easy Pharmacies solution includes an ability to house the pharmaceutical name, as well as the phone number for easy identification. Lastly, since Easy Pharmacies has a network of different pharmacies across the country, there needed to be a way to identify where a particular drug is located. There is a list of different store locations, identified by a unique pharmacy id, and the price for that particular drug at that location.

Easy Pharmacies also deal with patients' prescriptions, this is extremely important, in that, the prescriptions need to be filled for the correct customer, and so that no fatal mistake in drugs being handed out occurs. Because of this, the robust solution is to incorporate the patient's social security number and the doctor's social security number so that no mistakes in identity occur. Furthermore, each prescription contains the quantity, the date the prescription was created, and the specific drug identification number. Once a perception is filled, it is then saved for safe keeping in case an issue were to arise in the future. The information that is saved includes: the prescription number (RX), the pharmacy where it was completed, the date, and the type of drug.

To conclude, apart from having a list of patients, doctors, Easy Pharmacies also have the obligation to keep track of their contracts with drug manufacturers. Supervisors are part of the completed Easy Pharmacies solution where a unique ID identifies the name and the pharmacy where a supervisor is from. This supervisor ID is then assigned to the list of contracts Easy Pharmacies has with manufacturers that include the start date of the contract, the end date, which pharmacy the contract belongs to, which manufacturer, and the actual text contract. At any point

the supervisors assigned to the contract can be updated, removed, or changed. Likewise, supervisors can be part of multiple contracts at a time.

# ER Model (ksdb)



## Methodology Behind Design

The database design contains ten different tables that are vital to the Easy Pharmacies operation. At a glance, the main tables are that of patient, doctor, pharmacy, drug, pharmaceutical, and prescription. The tables that are not mentioned, i.e. drug location, supervisor, contract, and prescription filled are essentially helpers to the main tables with extra information about the operation.

## Database Structure

Table	Description
patient	Unique patient id, however, a unique SSN also collected as identification in the table. Other information: name, age, address, and phone number.
doctor	Unique doctor id, however, a unique SSN also collected as identification in the table. Other information: name, speciality, and experience years.
pharmacy	Unique pharmacy id used for identification in the table. Other information collected: name of pharmacy, address, and phone number.
drug	Unique drug id to identify the drug within the table. Other information: generic drug name, trade name.
pharmaceutical	Unique pharmaceutical name to identify in table. Other basic information: phone number.
prescription	Unique rx number is used to identify the table. Other basic information: quantity, date of prescription, patient id, drug id from drug table, doctor that gave out the prescription using doctor id from doctor table
drug location	Unique location id used to identify on the table. Other information that is stored includes the price, and the pharmacy at which the drug is located using the pharmacy id

	from the pharmacy table. The drug id is also stored.
supervisor	Uniquely identified using a supervisor id in the table. Other info that is stored is the name (first, last) of the supervisor and the pharmacy the supervisor is located at.
contract	Uniquely identified with a contract id on the table. Other information that is stored is the start date, the end date for the contract, the full text contract. Additionally, the pharmacy id from the pharmacy, supervisor id from the supervisor table, and the pharmaceutical name from the pharmaceutical table.
prescription filled	Unique filled id. Other information stored in this table: prescription rx number, pharmacy id, completed date, and drug id.

**ER Model Assumptions**

- Tables can exist without an implicit foreign key needed when creating a new row.
- Prescription\_filled table forcefully needs all fields not be null, otherwise the prescription shouldn't be in the prescription\_filled table to begin with.
- Client does not want to use SSN of patients and doctors to identify them across the database. Although SSN is still there, for privacy reasons, patient\_id, and doctor\_id is better.

## Table Creation Script

```
-- MySQL Script generated by MySQL Workbench
-- Wed Feb  8 03:17:58 2023
-- Model: New Model      Version: 1.0
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZE
RO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-----
-- Schema ksdb
-----

-----
-- Schema ksdb
-----
CREATE SCHEMA IF NOT EXISTS `ksdb` DEFAULT CHARACTER SET utf8 ;
USE `ksdb` ;

-----
-- Table `ksdb`.`doctor`
-----
CREATE TABLE IF NOT EXISTS `ksdb`.`doctor` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `last_name` VARCHAR(255) NULL,
  `first_name` VARCHAR(255) NULL,
  `specialty` VARCHAR(255) NULL,
  `experience` VARCHAR(255) NULL,
  `ssn` VARCHAR(255) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

-----
-- Table `ksdb`.`patient`
-----
CREATE TABLE IF NOT EXISTS `ksdb`.`patient` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `last_name` VARCHAR(255) NULL,
  `first_name` VARCHAR(255) NULL,
  `dob` VARCHAR(255) NULL,
  `ssn` VARCHAR(255) NULL,
  `street` VARCHAR(255) NULL,
  `city` VARCHAR(255) NULL,
  `state` VARCHAR(255) NULL,
  `zip` VARCHAR(255) NULL,
```



```

`doctor_id` INT NULL,
PRIMARY KEY (`id`),
INDEX `fk_patient_doctor1_idx` (`doctor_id` ASC) VISIBLE,
CONSTRAINT `fk_patient_doctor1`
  FOREIGN KEY (`doctor_id`)
  REFERENCES `ksdb`.`doctor` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `ksdb`.`pharamaceutical`
-----

```

```

CREATE TABLE IF NOT EXISTS `ksdb`.`pharamaceutical` (
  `name` VARCHAR(255) NOT NULL,
  `phone` VARCHAR(255) NULL,
  PRIMARY KEY (`name`))
ENGINE = InnoDB;

```

```

-----
-- Table `ksdb`.`drug`
-----

```

```

CREATE TABLE IF NOT EXISTS `ksdb`.`drug` (
  `drug_id` INT NOT NULL,
  `trade_name` VARCHAR(255) NULL,
  `formula` VARCHAR(255) NULL,
  PRIMARY KEY (`drug_id`))
ENGINE = InnoDB;

```

```

-----
-- Table `ksdb`.`pharmacy`
-----

```

```

CREATE TABLE IF NOT EXISTS `ksdb`.`pharmacy` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NULL,
  `address` VARCHAR(255) NULL,
  `phone` VARCHAR(255) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

```

```

-----
-- Table `ksdb`.`drug_location`
-----

```

```

CREATE TABLE IF NOT EXISTS `ksdb`.`drug_location` (
  `location_id` INT NOT NULL AUTO_INCREMENT,
  `drug_id` INT NOT NULL,
  `pharmacy_id` INT NOT NULL,
  `price` DECIMAL(8,2) NULL,

```

```

PRIMARY KEY (`location_id`),
INDEX `fk_drug_location_drug1_idx` (`drug_id` ASC) VISIBLE,
INDEX `fk_drug_location_pharmacy1_idx` (`pharmacy_id` ASC) VISIBLE,
CONSTRAINT `fk_drug_location_drug1`
  FOREIGN KEY (`drug_id`)
  REFERENCES `ksdb`.`drug` (`drug_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_drug_location_pharmacy1`
  FOREIGN KEY (`pharmacy_id`)
  REFERENCES `ksdb`.`pharmacy` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `ksdb`.`prescription`
-----

```

```

CREATE TABLE IF NOT EXISTS `ksdb`.`prescription` (
  `rx` INT NOT NULL AUTO_INCREMENT,
  `drug_id` INT NOT NULL,
  `patient_id` INT NOT NULL,
  `doctor_id` INT NOT NULL,
  `quantity` INT NOT NULL,
  `date` DATE NOT NULL,
  PRIMARY KEY (`rx`),
  INDEX `fk_prescription_drug1_idx` (`drug_id` ASC) VISIBLE,
  INDEX `fk_prescription_patient1_idx` (`patient_id` ASC) VISIBLE,
  INDEX `fk_prescription_doctor1_idx` (`doctor_id` ASC) VISIBLE,
  CONSTRAINT `fk_prescription_drug1`
    FOREIGN KEY (`drug_id`)
    REFERENCES `ksdb`.`drug` (`drug_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_prescription_patient1`
    FOREIGN KEY (`patient_id`)
    REFERENCES `ksdb`.`patient` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_prescription_doctor1`
    FOREIGN KEY (`doctor_id`)
    REFERENCES `ksdb`.`doctor` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `ksdb`.`supervisor`
-----

```

```

CREATE TABLE IF NOT EXISTS `ksdb`.`supervisor` (

```

```

`id` INT NOT NULL AUTO_INCREMENT,
`first_name` VARCHAR(255) NULL,
`last_name` VARCHAR(255) NULL,
`pharmacy_id` INT NOT NULL,
PRIMARY KEY (`id`),
INDEX `fk_supervisor_pharmacy1_idx` (`pharmacy_id` ASC) VISIBLE,
CONSTRAINT `fk_supervisor_pharmacy1`
  FOREIGN KEY (`pharmacy_id`)
  REFERENCES `ksdb`.`pharmacy` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `ksdb`.`contract`
-----

```

```

CREATE TABLE IF NOT EXISTS `ksdb`.`contract` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `start` DATE NOT NULL,
  `end` DATE NOT NULL,
  `pharmacy_id` INT NOT NULL,
  `pharamaceutical_name` VARCHAR(255) NOT NULL,
  `text` VARCHAR(255) NOT NULL,
  `supervisor_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_contract_pharmacy1_idx` (`pharmacy_id` ASC) VISIBLE,
  INDEX `fk_contract_pharamaceutical1_idx` (`pharamaceutical_name`
ASC) VISIBLE,
  INDEX `fk_contract_supervisor1_idx` (`supervisor_id` ASC) VISIBLE,
  CONSTRAINT `fk_contract_pharmacy1`
    FOREIGN KEY (`pharmacy_id`)
    REFERENCES `ksdb`.`pharmacy` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_contract_pharamaceutical1`
    FOREIGN KEY (`pharamaceutical_name`)
    REFERENCES `ksdb`.`pharamaceutical` (`name`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_contract_supervisor1`
    FOREIGN KEY (`supervisor_id`)
    REFERENCES `ksdb`.`supervisor` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `ksdb`.`prescription_filled`
-----

```

```

CREATE TABLE IF NOT EXISTS `ksdb`.`prescription_filled` (

```

```

`filled_id` INT NOT NULL AUTO_INCREMENT,
`prescription_rx` INT NOT NULL,
`pharmacy_id` INT NOT NULL,
`completed` DATE NOT NULL,
`drug_drug_id` INT NOT NULL,
PRIMARY KEY (`filled_id`),
INDEX `fk_prescription_filled_prescription1_idx` (`prescription_rx`
ASC) VISIBLE,
INDEX `fk_prescription_filled_pharmacy1_idx` (`pharmacy_id` ASC)
VISIBLE,
INDEX `fk_prescription_filled_drug1_idx` (`drug_drug_id` ASC)
VISIBLE,
CONSTRAINT `fk_prescription_filled_prescription1`
FOREIGN KEY (`prescription_rx`)
REFERENCES `ksdb`.`prescription` (`rx`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_prescription_filled_pharmacy1`
FOREIGN KEY (`pharmacy_id`)
REFERENCES `ksdb`.`pharmacy` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_prescription_filled_drug1`
FOREIGN KEY (`drug_drug_id`)
REFERENCES `ksdb`.`drug` (`drug_id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

## Query Examples

1. Which contracts does a specific contract supervisor have?
  - a. `SELECT * FROM contract INNER JOIN supervisor ON supervisor.id = contract.supervisor_id WHERE Contract.contract_id = '523';`
2. How many times a patient has gotten prescriptions over a specified date?

```
a. SELECT * FROM prescription INNER JOIN patient ON
patient.id = prescription.patient_id INNER JOIN
doctor ON doctor.id = prescription.doctor_id INNER
JOIN drug ON drug.drug_id = prescription.drug_id WHERE
prescription.date = '2023-01-25';
```

3. Which Doctors have more than 10 years of experience?

```
a. SELECT doctor.id, doctor.experience FROM Doctor WHERE
experience > 9;
```

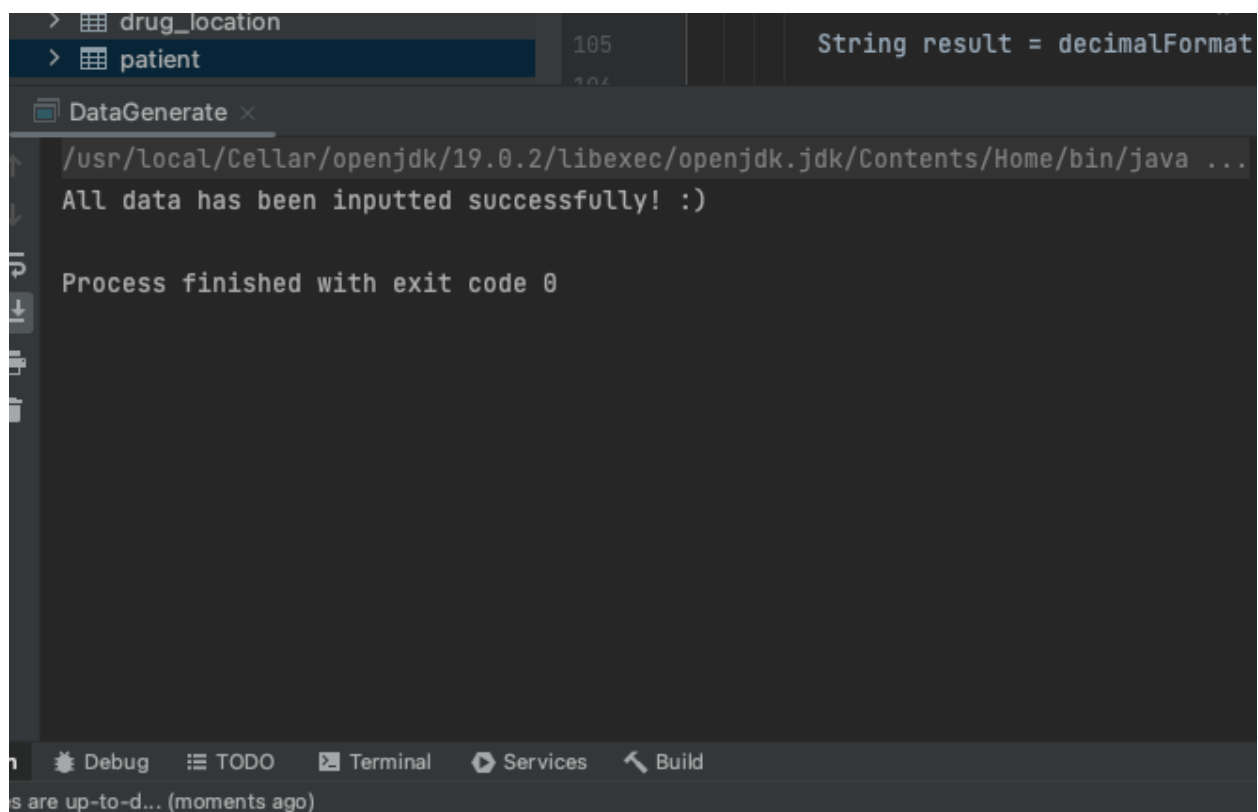
4. List all contracts with a specific pharmacy?

```
a. SELECT * FROM contact WHERE contract.pharmacy_id =
'7';
```

5. How many prescriptions were filled today?

```
a. SELECT * FROM prescription_filled WHERE
prescription_filled.completed '2023-02-07';
```

## Member 1 Java JDBC: SUCCESS



## Results: 10 Doctors

WHERE ORDER BY

	id	last_name	first_name	specialty	experience	ssn
1	11	Fahey	Avis	Dermatology	2017	202224467
2	12	Veum	Ayesha	Internal medicine	1977	237421041
3	13	Huel	Margeret	Pathology	2002	242736581
4	14	Goodwin	Ricardo	Urology	1929	104580141
5	15	Considine	Ada	Dermatology	2011	704891344
6	16	Hirthe	Adria	Dermatology	1931	238752035
7	17	Shields	Terrell	Preventive medicine	2008	342360055
8	18	Corkery	Chrissy	Surgery	1998	247041438
9	19	Robel	Kenton	Neurology	1992	120833225
10	20	Carroll	Dong	Psychiatry	1951	845216808

## Results: 100 Patients and 100 Prescriptions

id	last_name	first_name	dob	ssn	street	city	state	zip	doctor_id
84	Wunsch	Oliver	1966-02-22	235470432	3785 Alejandro Inlet	Port Lindsay	West Virginia	21168	17
85	Goyette	Tequila	1969-01-30	115141681	3791 Amos Oval	Muellerhaven	South Carolina	35762	20
86	Tillman	Ted	1960-11-07	277251338	523 Thompson Lights	Corinnaborough	South Carolina	81065	16
87	Weissnat	Bud	2002-06-30	124543657	57429 Trisha Hollow	Reynoldsfurt	Wyoming	29031	11
88	Wolff	Leonard	1985-12-21	825480629	153 Herzog Flats	Rogahnmouth	Maryland	82508	19
89	Mertz	Olivia	1918-04-01	371881865	73763 Hackett Loaf	North Berryton	Illinois	43470	11
90	Emard	Morris	1906-05-24	266576218	2053 Yundt Point	Boyleville	Washington	28028	14
91	Casper	Clair	1953-02-12	161228056	04506 Kessler Brook	East Felix	Mississippi	91409	13
92	Graham	Providencia	2011-12-03	251311369	940 Kiehn Square	Keithton	Alaska	50490	20
93	Swift	Granville	1924-02-10	312898523	4087 Kuhlman Spring	North Detra	California	49664	20
94	Wisozk	Krystyna	1990-08-10	654451642	09875 Johnson Camp	Cedrickmouth	Maryland	71964	19
95	King	Charmaine	1946-11-18	114571228	57937 Weissnat Junction	Wilhelminahaven	North Dakota	24145	14
96	Stracke	Rubie	2020-10-29	622374855	05449 Heaney Garden	Bartonhaven	Kansas	31438	13
97	Bogan	Randall	1924-02-07	225275081	6315 Hilpert Radial	Kevington	New York	29274	14
98	Parisian	Marna	1948-09-26	548628505	6818 Romana Unions	Lake Flor	Michigan	19876	14
99	Schneider	Chuck	1951-04-30	472250653	3660 Otha Well	Harveyemouth	South Dakota	13047	16
100	Abbott	Sherrie	1911-01-06	222466779	03428 Kub Gardens	Ferrystad	Hawaii	82281	15

WHERE	ORDER BY	rx	drug_id	patient_id	doctor_id	quantity	date
1		102	50	174	11	32	2023-01-11
2		103	11	197	15	49	2023-02-04
3		104	95	152	19	30	2023-01-23
4		105	51	178	17	30	2023-01-27
5		106	59	109	16	23	2023-02-06
6		107	81	156	18	48	2023-01-31
7		108	39	147	13	66	2023-01-27
8		109	40	158	14	18	2023-02-06
9		110	4	108	12	28	2023-02-06
10		111	43	163	19	54	2023-01-23
11		112	11	117	18	5	2023-01-22
12		113	13	142	15	70	2023-01-16
13		114	40	113	11	30	2023-01-14
14		115	45	119	19	45	2023-01-29
15		116	99	180	13	40	2023-01-17
16		117	83	113	11	7	2023-01-09
17		118	59	178	20	95	2023-02-06
18		119	92	138	11	91	2023-01-21
19		120	9	168	16	10	2023-01-29
20		121	70	126	20	99	2023-02-04
21		122	80	187	12	82	2023-01-10
22		123	84	181	14	66	2023-01-31
23		124	67	119	11	20	2023-01-12
24		125	70	113	20	70	2023-01-23
25		126	7	193	14	55	2023-02-05
26		127	14	168	19	77	2023-01-15
27		128	16	199	19	47	2023-01-23
28		129	5	127	13	78	2023-01-13
29		130	77	178	13	18	2023-01-16
30		131	11	194	17	26	2023-02-07
31		132	23	112	15	97	2023-01-21
32		133	11	106	18	91	2023-01-16
33		134	71	143	18	94	2023-01-08
34		135	98	107	20	63	2023-01-23
35		136	64	141	19	44	2023-02-07
36		137	1	101	12	20	2023-02-01

## Results: 5 Test Pharmacies for Future Use

	id	name	address	phone
1	6	CVS	933 Vida Rapids 28517 Nevada 49634	(507) 256-1709
2	7	Rite Aid	180 Murray Ways 29461 Vermont 65927	(515) 984-6321
3	8	Walgreens	053 Funk Freeway 14905 Kansas 69672	(516) 757-3408
4	9	CVS	152 Thanh Pine 87261 Ohio 25084	(719) 678-0985
5	10	CVS	855 Bartoletti Valleys 24592 Alaska 46528	(732) 386-8700 x4648

Results: After running provided drug script, and using those drugs to add location of where those drugs are located.

	location_id	drug_id	pharmacy_id	price
1	11	44	7	22.80
2	12	48	6	19.40
3	13	74	9	98.90
4	14	33	10	5.40
5	15	6	7	57.70
6	16	34	6	66.50
7	17	12	10	66.80
8	18	89	8	94.20
9	19	84	7	66.20
10	20	51	6	3.20



## **Disclaimer**

The following two screenshots are of member one web app section. These next two screenshots only show the success of creating a new prescription and getting that prescription filled. However, there is error checking and sanitization of the inputted data. The same sanitization is being used in member two web app sections. So, there was no need to include proof of sanitization for member one, and proof will be provided for member two.

## Results: New Prescription Was Created.

rx_id	drug_id	patient_id	doctor_id	quantity	date
73	174	28	127	12	2023-01-23
74	175	17	141	20	2023-01-24
75	176	16	163	11	2023-01-28
76	177	52	183	17	2023-01-14
77	178	32	135	11	2023-02-01
78	179	30	140	14	2023-01-31
79	180	99	123	15	2023-01-15
80	181	29	137	12	2023-02-05
81	182	4	145	18	2023-01-24
82	183	62	127	18	2023-01-17
83	184	17	150	13	2023-02-04
84	185	58	197	16	2023-01-28
85	186	38	188	12	2023-01-24
86	187	57	144	12	2023-01-11
87	188	66	129	12	2023-02-01
88	189	98	176	17	2023-01-25
89	190	54	159	20	2023-01-12
90	191	55	176	19	2023-01-14
91	192	97	133	15	2023-01-09
92	193	23	169	20	2023-01-26
93	194	82	159	19	2023-02-07
94	195	44	153	13	2023-01-27
95	196	12	180	20	2023-01-19
96	197	2	192	15	2023-01-17
97	198	38	130	18	2023-01-20
98	199	59	137	20	2023-01-23
99	200	87	140	13	2023-01-08
100	201	38	183	18	2023-01-16
101	202	44	101	11	2023-02-08
102	203	74	183	13	2023-02-08
103	204	6	185	15	2023-02-08
104	205	12	187	17	2023-02-08
105	206	84	127	19	2023-02-08
106	308	6	306	12	2023-02-08

Prescription created.

Rx: 308  
 Doctor: 237421041  
 First Name: Ayesha  
 Last Name: Veum  
 Patient: 673829562  
 First Name: Keldin  
 Last Name: Maldonado  
 Drug: Xanax  
 Quantity: 31  
 Pharmacy: 7  
 Name: Rite Aid  
 Address: 180 Murray Ways 29461 Vermont 65927  
 Phone: (515) 984-6321  
 Date Filled: NA--Patient can request fill from Main Menu  
 Cost: \$ 57.70

[Main Menu](#)

## Results: Prescription Was Filled.

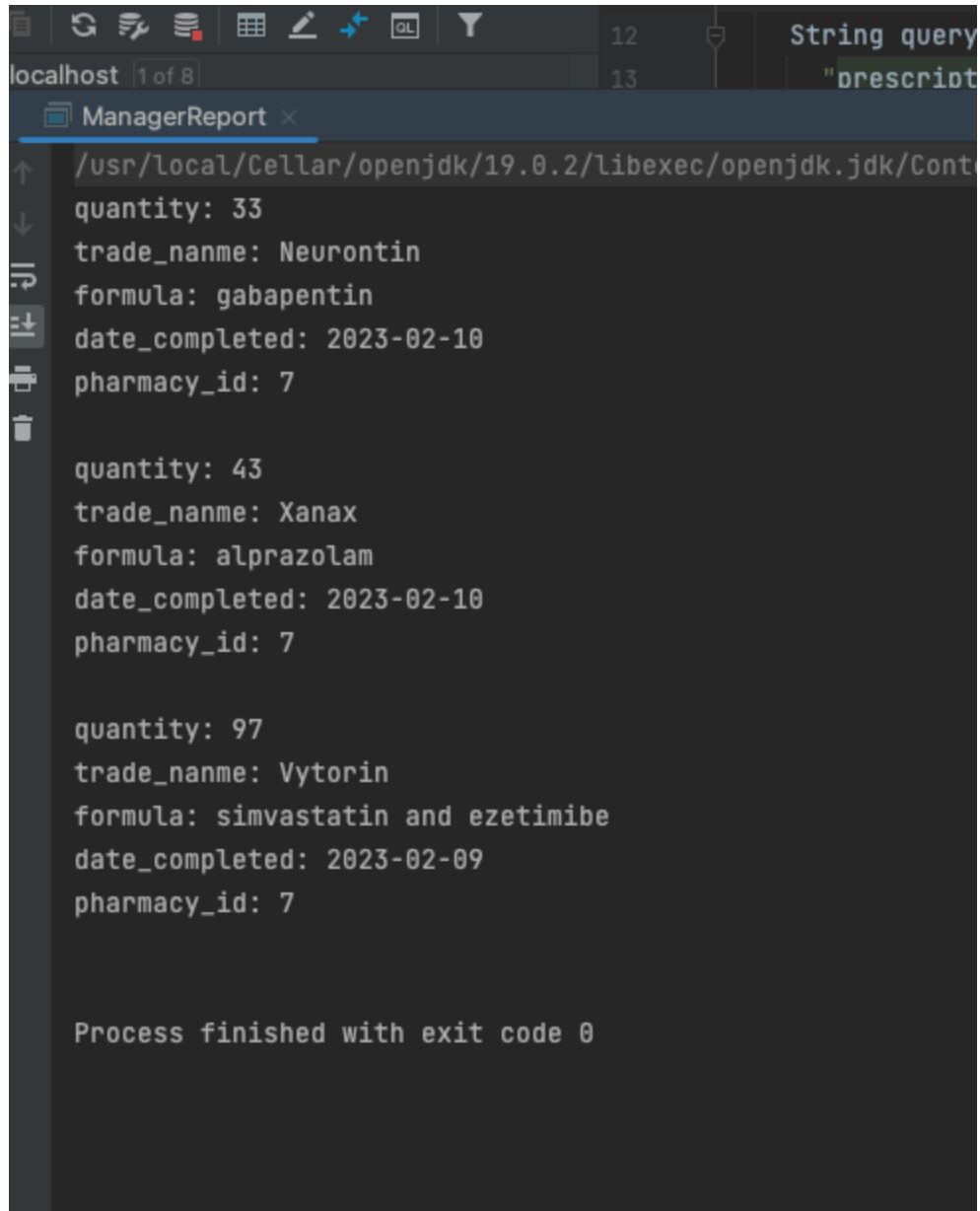
filled_id	prescription_rx	pharmacy_id	completed	drug_drug_id
1	2	202	7 2023-02-10	44
2	3	203	9 2023-02-10	74
3	4	204	7 2023-02-10	6
4	5	205	10 2023-02-09	12
5	6	206	7 2023-02-09	84
6	7	308	7 2023-02-09	6

Prescription has been filled.

Rx: 308  
 Doctor: 12  
 First Name: Ayesha  
 Last Name: Veum  
 Patient: 306  
 First Name: Keldin  
 Last Name: Maldonado  
 Drug: alprazolam  
 Quantity: 31  
 Pharmacy: 7  
 Name: Rite Aid  
 Address: 180 Murray Ways 29461 Vermont 65927  
 Phone: (515) 984-6321  
 Date Filled: 2023-02-09  
 Cost: \$ 57.70

[Main Menu](#)

## Results: Manager Report–Member 2 Java JDBC: SUCCESS



```
localhost 1 of 8 | 12 | String query  
| 13 | "prescript  
ManagerReport x  
/usr/local/Cellar/openjdk/19.0.2/libexec/openjdk.jdk/Cont  
quantity: 33  
trade_name: Neurontin  
formula: gabapentin  
date_completed: 2023-02-10  
pharmacy_id: 7  
  
quantity: 43  
trade_name: Xanax  
formula: alprazolam  
date_completed: 2023-02-10  
pharmacy_id: 7  
  
quantity: 97  
trade_name: Vytorin  
formula: simvastatin and ezetimibe  
date_completed: 2023-02-09  
pharmacy_id: 7  
  
Process finished with exit code 0
```

**Validation: Member 2–Check For Blank Fields (Other SSN Specific Checks Are Also Ran)**

## Register as new user

Check SSN field

Your SSN:

Your First Name:

Your Last Name:

Birth Date:

Street:

City:

State:

Zipcode:

Primary Physician  
Name:

**Validation: Member 2–Check For Illegal Characters**

## Register as new user

Check your first name field.

Your SSN: .....

Your First Name: Keldin\$

Your Last Name: Maldonado

Birth Date: 05/24/1997

Street: 123 Wood St

City: Riverside

State: California

Zipcode: 92502

Primary Physician  
Name: Ayesha Veum

Register

## Validation: Member 2–Zip Code Checks

# Register as new user

Check zip code

Your SSN:

Your First Name:

Your Last Name:

Birth Date:

Street:

City:

State:

Zipcode:

Primary Physician  
Name:

# Results: New Patient Registered.

id	last_name	first_name	dob	ssn	street	city	state	zip	doctor_id
68	Waelchi	Ernest	1971-12-06	775790163	102 Stokes Heights	East Hazel	New Jersey	64416	18
69	Kunde	Pricilla	1930-03-08	535468682	8124 Houston Union	East Jena	Wisconsin	95187	17
70	Homenick	Vito	2013-10-30	142141861	75175 Beahan Path	Trentfurt	Oklahoma	26040	14
71	Mosciski	Preston	2014-02-06	888316828	142 Lockman Loop	West Vance	Georgia	30798	15
72	Paucck	Genevie	1942-09-19	866720146	0956 Corliss Spring	Jakubowskishire	Massachusetts	86913	16
73	Kuhlic	Hank	1977-10-03	395365955	45698 Cremin Radial	East Arielle	Hawaii	38778	20
74	Tuncotte	Doretha	1912-08-09	740875471	25405 Top Squares	North Yvette	Delaware	94473	18
75	Glover	Elanor	1981-06-25	407842731	453 Zella Fall	Lake Sina	Hawaii	02698	20
76	Hahn	Lemuel	2008-07-16	525691459	6222 Pfannerstill Flats	Jonestown	Delaware	59720	15
77	Haag	Lovella	1951-05-07	474693608	42669 Reid Circle	Lake Garlandmouth	Washington	05476	13
78	DuBuque	Blake	1914-06-20	528276671	4769 Shanae Inlet	Parkerside	Montana	33487	15
79	Mitchell	Stephan	1986-07-02	187588733	170 Oberbrunner Parkways	East Tammara	Kentucky	13416	15
80	Kozey	Garry	1910-08-17	351117514	159 Haag Fords	New Damion	Washington	12424	13
81	Ziemann	John	2005-10-24	721065729	89944 Erwin Parkways	Norbertmouth	California	13050	14
82	Walker	Rudolph	1965-05-29	150325129	35153 Tran Harbor	Lindgrenside	Maine	59881	15
83	Smith	Mirtha	2002-07-09	395520539	427 Silva Ports	Tomliemouth	Kentucky	42408	18
84	Wunsch	Oliver	1966-02-22	235470432	3785 Alejandro Inlet	Port Lindsay	West Virginia	21168	17
85	Goyette	Tequila	1969-01-30	115141681	3791 Amos Oval	Muellerhaven	South Carolina	35762	20
86	Tillman	Ted	1960-11-07	277251338	523 Thompson Lights	Corinnaborough	South Carolina	81065	16
87	Weissnat	Bud	2002-06-30	124543657	57429 Trisha Hollow	Reynoldsfurt	Wyoming	29031	11
88	Wolff	Leonard	1985-12-21	825480629	153 Herzog Flats	Rogahnmouth	Maryland	82508	19
89	Mertz	Olivia	1910-04-01	371881865	73763 Hackett Loaf	North Bennyton	Illinois	43470	11
90	Emard	Morris	1986-05-24	266576218	2053 Yundt Point	Boyleville	Washington	28028	14
91	Casper	Clair	1953-02-12	161228056	04506 Kessler Brook	East Felix	Mississippi	91409	13
92	Graham	Providencia	2011-12-03	251311369	940 Kiehn Square	Keithton	Alaska	50490	20
93	Swift	Granville	1924-02-10	312898523	4087 Kuhlman Spring	North Detra	California	49664	20
94	Wisozk	Krystyna	1990-08-10	654451642	09875 Johnson Camp	Cedrickmouth	Maryland	71964	19
95	King	Charmaine	1946-11-18	114571228	57937 Weissnat Junction	Wilhelmahaven	North Dakota	24145	14
96	Stracke	Rubie	2020-10-29	622374855	05449 Heaney Garden	Bartonhaven	Kansas	31438	13
97	Bogan	Randall	1924-02-07	225275081	6315 Hilpert Radial	Kevinnton	New York	29274	14
98	Parisian	Marna	1948-09-26	548628505	6818 Romana Unions	Lake Flor	Michigan	19876	14
99	Schneider	Chuck	1951-04-30	472250653	3660 Otha Well	Harveyemouth	South Dakota	13047	16
100	Abbott	Sherrie	1911-01-06	222466779	03428 Kub Gardens	Ferrystad	Hawaii	82281	15
101	Maldonado	Keldin	1997-05-24	673829562	123 Wood St	Riverside	California	92502	12

Registration successful.

Patient ID: 306  
 First Name: Keldin  
 Last Name: Maldonado  
 Birthdate: 1997-05-24  
 Street: 123 Wood St  
 City: Riverside  
 State: California  
 Zipcode: 92502  
 Primary Physician: Ayesha Veum

[Edit | Main Menu](#)

## **Conclusion**

To conclude, this project mostly taught us how to organize our thoughts into a repeatable format. In essence, databases are 80% planning and 20% implementation. Without proper planning then we run the risk of repeating data in different tables/columns. Overall, the ability to think of this project as if we were a consolidation company was a great learning experience.