Database Systems

(331-32)

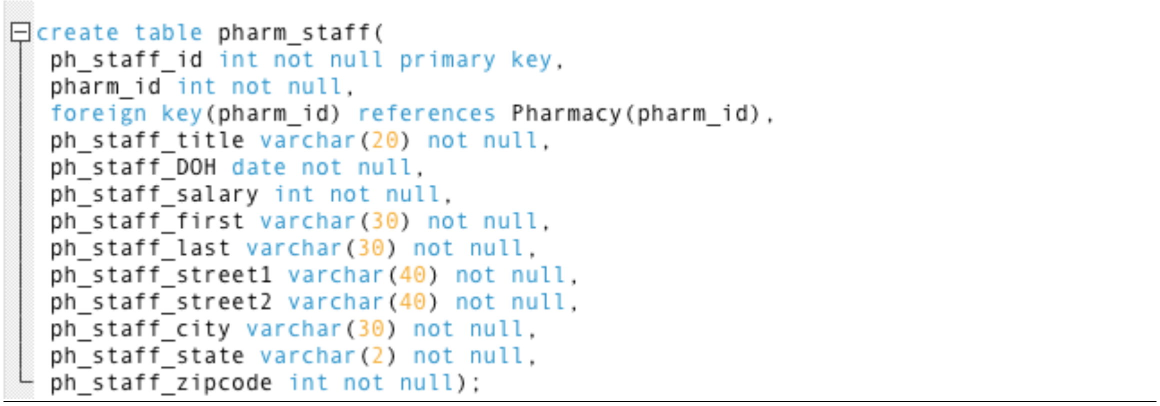
I-Stop Database - Part2

May 23, 2016

Sumaita Hussain
12064861

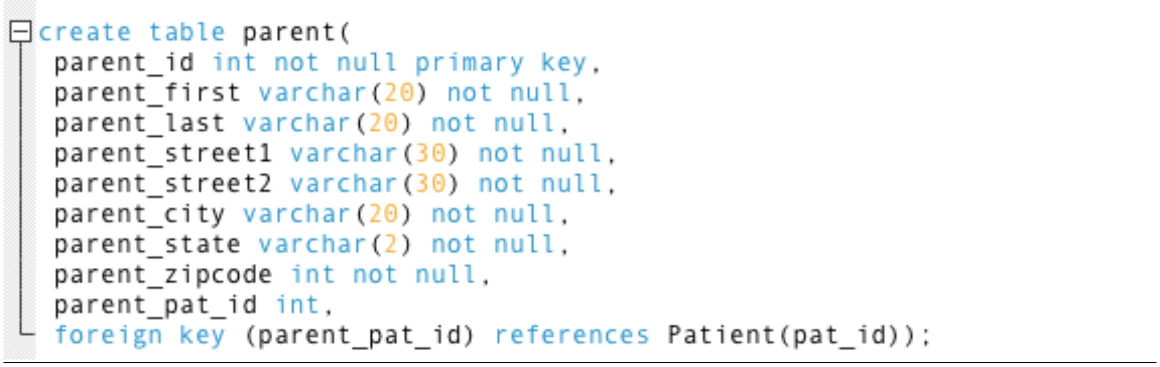
Creating Tables – New

- **Pharm_Staff**



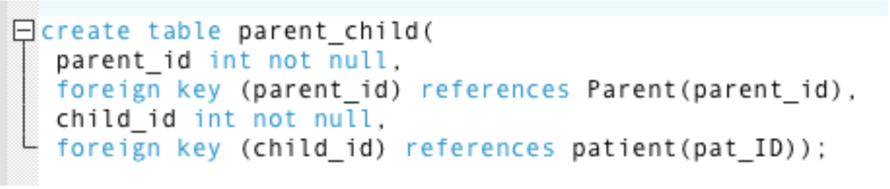
```
create table pharm_staff(
 ph_staff_id int not null primary key,
 pharm_id int not null,
 foreign key(pharm_id) references Pharmacy(pharm_id),
 ph_staff_title varchar(20) not null,
 ph_staff_DOH date not null,
 ph_staff_salary int not null,
 ph_staff_first varchar(30) not null,
 ph_staff_last varchar(30) not null,
 ph_staff_street1 varchar(40) not null,
 ph_staff_street2 varchar(40) not null,
 ph_staff_city varchar(30) not null,
 ph_staff_state varchar(2) not null,
 ph_staff_zipcode int not null);
```

- **Parent**



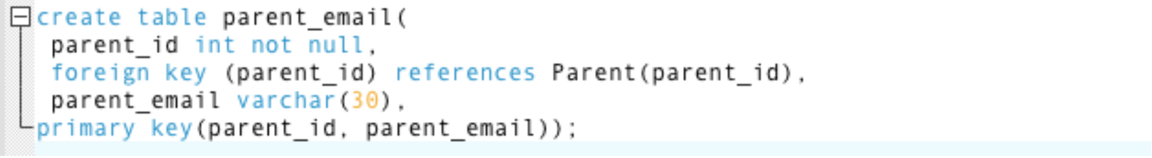
```
create table parent(
 parent_id int not null primary key,
 parent_first varchar(20) not null,
 parent_last varchar(20) not null,
 parent_street1 varchar(30) not null,
 parent_street2 varchar(30) not null,
 parent_city varchar(20) not null,
 parent_state varchar(2) not null,
 parent_zipcode int not null,
 parent_pat_id int,
 foreign key (parent_pat_id) references Patient(pat_id));
```

- **Parent_Child**



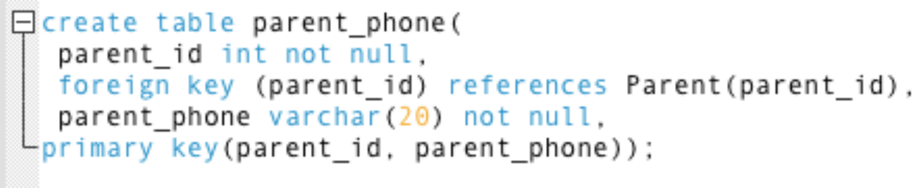
```
create table parent_child(
 parent_id int not null,
 foreign key (parent_id) references Parent(parent_id),
 child_id int not null,
 foreign key (child_id) references patient(pat_ID));
```

- **Parent_Email**



```
create table parent_email(
 parent_id int not null,
 foreign key (parent_id) references Parent(parent_id),
 parent_email varchar(30),
primary key(parent_id, parent_email));
```

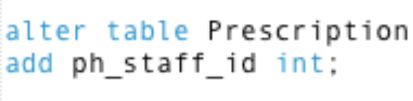
- **Parent_Phone**



```
create table parent_phone(
 parent_id int not null,
 foreign key (parent_id) references Parent(parent_id),
 parent_phone varchar(20) not null,
primary key(parent_id, parent_phone));
```

Alter Previous Tables

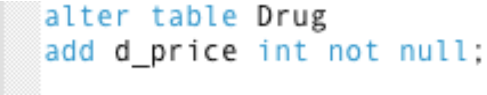
- **Prescription**



```
alter table Prescription
add ph_staff_id int;
```

- **Drug**

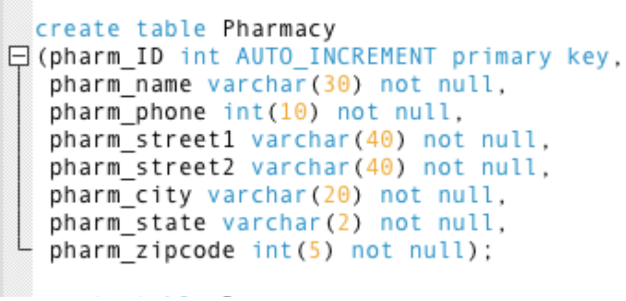
/* I added a d_price in drugs in order to calculate revenue*/



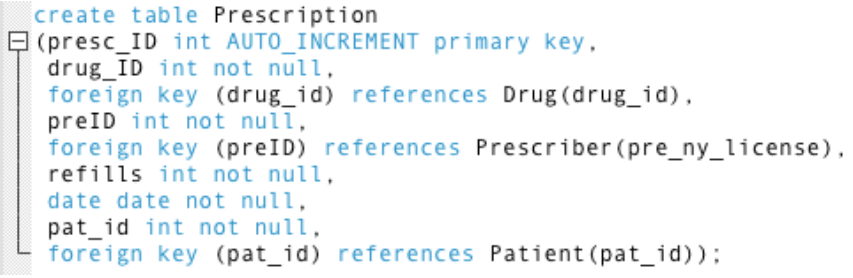
```
alter table Drug
add d_price int not null;
```

Creating Tables- Old

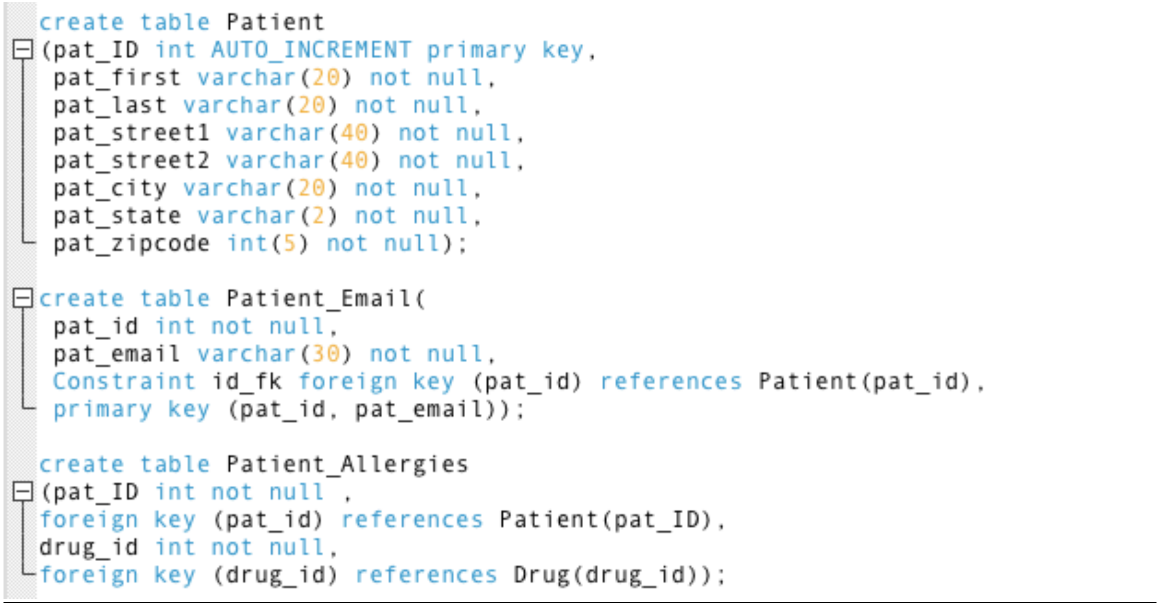
- Includes **Organization, Prescriber, Presc_Org, Prescriber_Phone, Pharmacy, Patient, Patient_Email, Prescription, Drug**



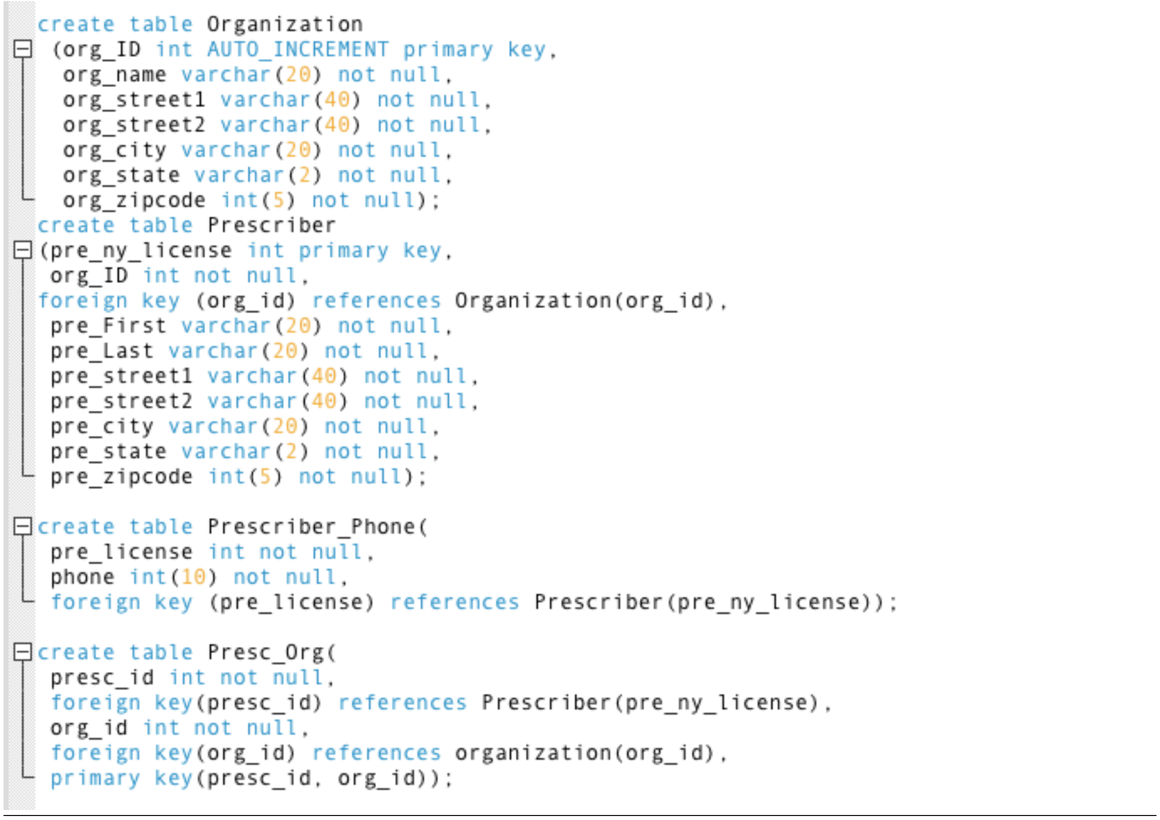
```
create table Pharmacy
(pharm_ID int AUTO_INCREMENT primary key,
 pharm_name varchar(30) not null,
 pharm_phone int(10) not null,
 pharm_street1 varchar(40) not null,
 pharm_street2 varchar(40) not null,
 pharm_city varchar(20) not null,
 pharm_state varchar(2) not null,
 pharm_zipcode int(5) not null);
```



```
create table Prescription
(presc_ID int AUTO_INCREMENT primary key,
 drug_ID int not null,
 foreign key (drug_id) references Drug(drug_id),
 preID int not null,
 foreign key (preID) references Prescriber(pre_ny_license),
 refills int not null,
 date date not null,
 pat_id int not null,
 foreign key (pat_id) references Patient(pat_id));
```

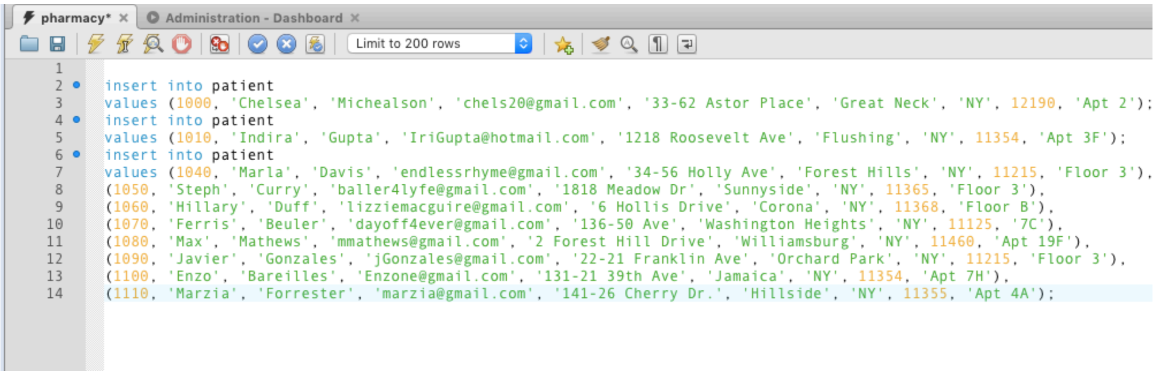


```
create table Patient
(pat_ID int AUTO_INCREMENT primary key,
 pat_first varchar(20) not null,
 pat_last varchar(20) not null,
 pat_street1 varchar(40) not null,
 pat_street2 varchar(40) not null,
 pat_city varchar(20) not null,
 pat_state varchar(2) not null,
 pat_zipcode int(5) not null);

create table Patient_Email(
 pat_id int not null,
 pat_email varchar(30) not null,
 Constraint id_fk foreign key (pat_id) references Patient(pat_id),
 primary key (pat_id, pat_email));

create table Patient_Allergies
(pat_ID int not null ,
foreign key (pat_id) references Patient(pat_ID),
drug_id int not null,
foreign key (drug_id) references Drug(drug_id));
```

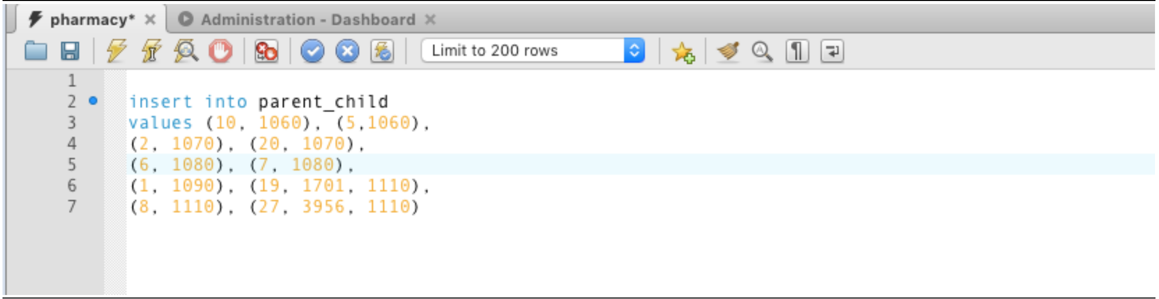


```
create table Organization
 (org_ID int AUTO_INCREMENT primary key,
  org_name varchar(20) not null,
  org_street1 varchar(40) not null,
  org_street2 varchar(40) not null,
  org_city varchar(20) not null,
  org_state varchar(2) not null,
  org_zipcode int(5) not null);
create table Prescriber
(pre_ny_license int primary key,
 org_ID int not null,
foreign key (org_id) references Organization(org_id),
 pre_First varchar(20) not null,
 pre_Last varchar(20) not null,
 pre_street1 varchar(40) not null,
 pre_street2 varchar(40) not null,
 pre_city varchar(20) not null,
 pre_state varchar(2) not null,
 pre_zipcode int(5) not null);

create table Prescriber_Phone(
 pre_license int not null,
 phone int(10) not null,
 foreign key (pre_license) references Prescriber(pre_ny_license));

create table Presc_Org(
 presc_id int not null,
 foreign key(presc_id) references Prescriber(pre_ny_license),
 org_id int not null,
 foreign key(org_id) references organization(org_id),
 primary key(presc_id, org_id));
```

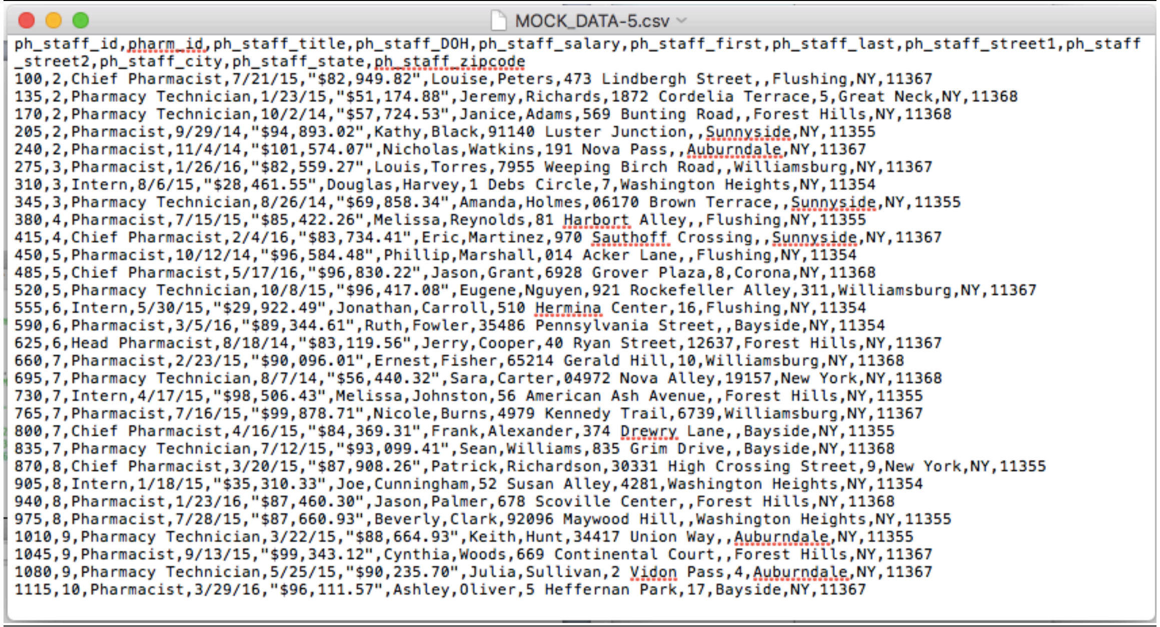
Inserting Data



```sql

insert into patient
values (1000, 'Chelsea', 'Michealson', 'chels20@gmail.com', '33-62 Astor Place', 'Great Neck', 'NY', 12190, 'Apt 2');
insert into patient
values (1010, 'Indira', 'Gupta', 'IriGupta@hotmail.com', '1218 Roosevelt Ave', 'Flushing', 'NY', 11354, 'Apt 3F');
insert into patient
values (1040, 'Marla', 'Davis', 'endlessrhyme@gmail.com', '34-56 Holly Ave', 'Forest Hills', 'NY', 11215, 'Floor 3'),
(1050, 'Steph', 'Curry', 'baller4lyfe@gmail.com', '1818 Meadow Dr', 'Sunnyside', 'NY', 11365, 'Floor 3'),
(1060, 'Hillary', 'Duff', 'lizziemacguire@gmail.com', '6 Hollis Drive', 'Corona', 'NY', 11368, 'Floor B'),
(1070, 'Ferris', 'Beuler', 'dayoff4ever@gmail.com', '136-50 Ave', 'Washington Heights', 'NY', 11125, '7C'),
(1080, 'Max', 'Mathews', 'mmathews@gmail.com', '2 Forest Hill Drive', 'Williamsburg', 'NY', 11460, 'Apt 19F'),
(1090, 'Javier', 'Gonzales', 'jGonzales@gmail.com', '22-21 Franklin Ave', 'Orchard Park', 'NY', 11215, 'Floor 3'),
(1100, 'Enzo', 'Bareilles', 'Enzone@gmail.com', '131-21 39th Ave', 'Jamaica', 'NY', 11354, 'Apt 7H'),
(1110, 'Marzia', 'Forrester', 'marzia@gmail.com', '141-26 Cherry Dr.', 'Hillside', 'NY', 11355, 'Apt 4A');
```



```sql

insert into parent_child
values (10, 1060), (5,1060),
(2, 1070), (20, 1070),
(6, 1080), (7, 1080),
(1, 1090), (19, 1701, 1110),
(8, 1110), (27, 3956, 1110)
```

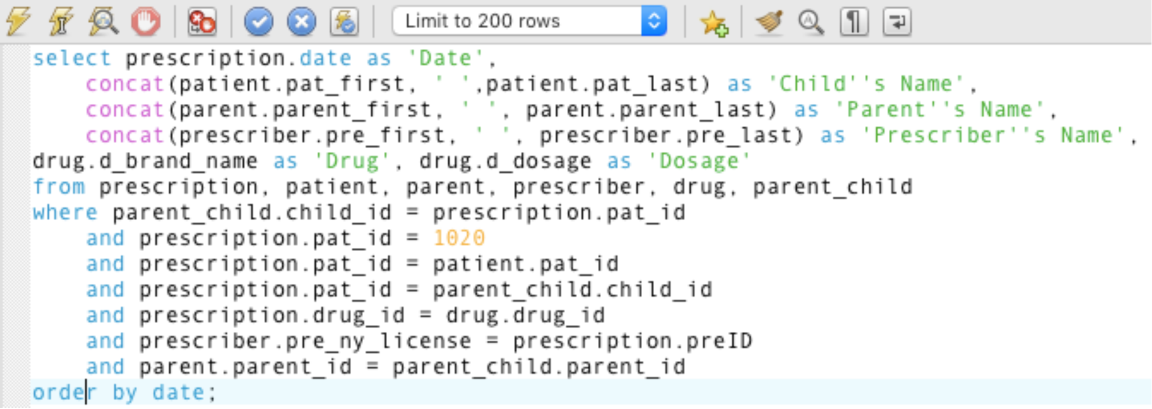


```
ph_staff_id,pharm_id,ph_staff_title,ph_staff_DOH,ph_staff_salary,ph_staff_first,ph_staff_last,ph_staff_street1,ph_staff_street2,ph_staff_city,ph_staff_state,ph_staff_zipcode
100,2,Chief Pharmacist,7/21/15,"$82,949.82",Louise,Peters,473 Lindbergh Street,,Flushing,NY,11367
135,2,Pharmacy Technician,1/23/15,"$51,174.88",Jeremy,Richards,1872 Cordelia Terrace,5,Great Neck,NY,11368
170,2,Pharmacy Technician,10/2/14,"$57,724.53",Janice,Adams,569 Bunting Road,,Forest Hills,NY,11368
205,2,Pharmacist,9/29/14,"$94,893.02",Kathy,Black,91140 Luster Junction,,Sunnyside,NY,11355
240,2,Pharmacist,11/4/14,"$101,574.07",Nicholas,Watkins,191 Nova Pass,,Auburndale,NY,11367
275,3,Pharmacist,1/26/16,"$82,559.27",Louis,Torres,7955 Weeping Birch Road,,Williamsburg,NY,11367
310,3,Intern,8/6/15,"$28,461.55",Douglas,Harvey,1 Debs Circle,7,Washington Heights,NY,11354
345,3,Pharmacy Technician,8/26/14,"$69,858.34",Amanda,Holmes,06170 Brown Terrace,,Sunnyside,NY,11355
380,4,Pharmacist,7/15/15,"$85,422.26",Melissa,Reynolds,81 Harbort Alley,,Flushing,NY,11355
415,4,Chief Pharmacist,2/4/16,"$83,734.41",Eric,Martinez,970 Sauthoff Crossing,,Sunnyside,NY,11367
450,5,Pharmacist,10/12/14,"$96,584.48",Phillip,Marshall,014 Acker Lane,,Flushing,NY,11354
485,5,Chief Pharmacist,5/17/16,"$96,830.22",Jason,Grant,6928 Grover Plaza,8,Corona,NY,11368
520,5,Pharmacy Technician,10/8/15,"$96,417.08",Eugene,Nguyen,921 Rockefeller Alley,311,Williamsburg,NY,11367
555,6,Intern,5/30/15,"$29,922.49",Jonathan,Carroll,510 Hermina Center,16,Flushing,NY,11354
590,6,Pharmacist,3/5/16,"$89,344.61",Ruth,Fowler,35486 Pennsylvania Street,,Bayside,NY,11354
625,6,Head Pharmacist,8/18/14,"$83,119.56",Jerry,Cooper,40 Ryan Street,12637,Forest Hills,NY,11367
660,7,Pharmacist,2/23/15,"$90,096.01",Ernest,Fisher,65214 Gerald Hill,10,Williamsburg,NY,11368
695,7,Pharmacy Technician,8/7/14,"$56,440.32",Sara,Carter,04972 Nova Alley,19157,New York,NY,11368
730,7,Intern,4/17/15,"$98,506.43",Melissa,Johnston,56 American Ash Avenue,,Forest Hills,NY,11355
765,7,Pharmacist,7/16/15,"$99,878.71",Nicole,Burns,4979 Kennedy Trail,6739,Williamsburg,NY,11367
800,7,Chief Pharmacist,4/16/15,"$84,369.31",Frank,Alexander,374 Drewry Lane,,Bayside,NY,11355
835,7,Pharmacy Technician,7/12/15,"$93,099.41",Sean,Williams,835 Grim Drive,,Bayside,NY,11368
870,8,Chief Pharmacist,3/20/15,"$87,908.26",Patrick,Richardson,30331 High Crossing Street,9,New York,NY,11355
905,8,Intern,1/18/15,"$35,310.33",Joe,Cunningham,52 Susan Alley,4281,Washington Heights,NY,11354
940,8,Pharmacist,1/23/16,"$87,460.30",Jason,Palmer,678 Scoville Center,,Forest Hills,NY,11368
975,8,Pharmacist,7/28/15,"$87,660.93",Beverly,Clark,92096 Maywood Hill,,Washington Heights,NY,11355
1010,9,Pharmacy Technician,3/22/15,"$88,664.93",Keith,Hunt,34417 Union Way,,Auburndale,NY,11355
1045,9,Pharmacist,9/13/15,"$99,343.12",Cynthia,Woods,669 Continental Court,,Forest Hills,NY,11367
1080,9,Pharmacy Technician,5/25/15,"$90,235.70",Julia,Sullivan,2 Vidon Pass,4,Auburndale,NY,11367
1115,10,Pharmacist,3/29/16,"$96,111.57",Ashley,Oliver,5 Heffernan Park,17,Bayside,NY,11367
```

Questions

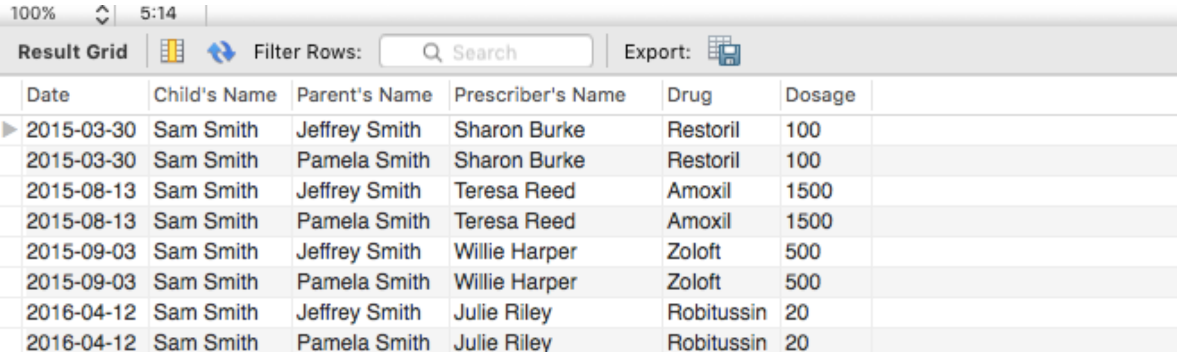
1. Identify the medication history of [child patient name]. Display the patient name, parent names, physician, drug, date of prescription and dosage. Order chronologically by date.

Input:



```
select prescription.date as 'Date',
    concat(patient.pat_first, ' ',patient.pat_last) as 'Child''s Name',
    concat(parent.parent_first, ' ', parent.parent_last) as 'Parent''s Name',
    concat(prescriber.pre_first, ' ', prescriber.pre_last) as 'Prescriber''s Name',
drug.d_brand_name as 'Drug', drug.d_dosage as 'Dosage'
from prescription, patient, parent, prescriber, drug, parent_child
where parent_child.child_id = prescription.pat_id
    and prescription.pat_id = 1020
    and prescription.pat_id = patient.pat_id
    and prescription.pat_id = parent_child.child_id
    and prescription.drug_id = drug.drug_id
    and prescriber.pre_ny_license = prescription.preID
    and parent.parent_id = parent_child.parent_id
order by date;
```

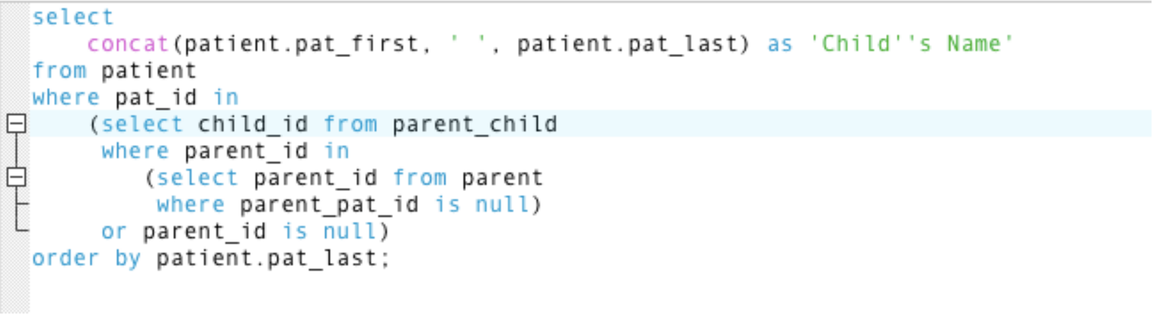
Output:



Date	Child's Name	Parent's Name	Prescriber's Name	Drug	Dosage
2015-03-30	Sam Smith	Jeffrey Smith	Sharon Burke	Restoril	100
2015-03-30	Sam Smith	Pamela Smith	Sharon Burke	Restoril	100
2015-08-13	Sam Smith	Jeffrey Smith	Teresa Reed	Amoxil	1500
2015-08-13	Sam Smith	Pamela Smith	Teresa Reed	Amoxil	1500
2015-09-03	Sam Smith	Jeffrey Smith	Willie Harper	Zoloft	500
2015-09-03	Sam Smith	Pamela Smith	Willie Harper	Zoloft	500
2016-04-12	Sam Smith	Jeffrey Smith	Julie Riley	Robitussin	20
2016-04-12	Sam Smith	Pamela Smith	Julie Riley	Robitussin	20

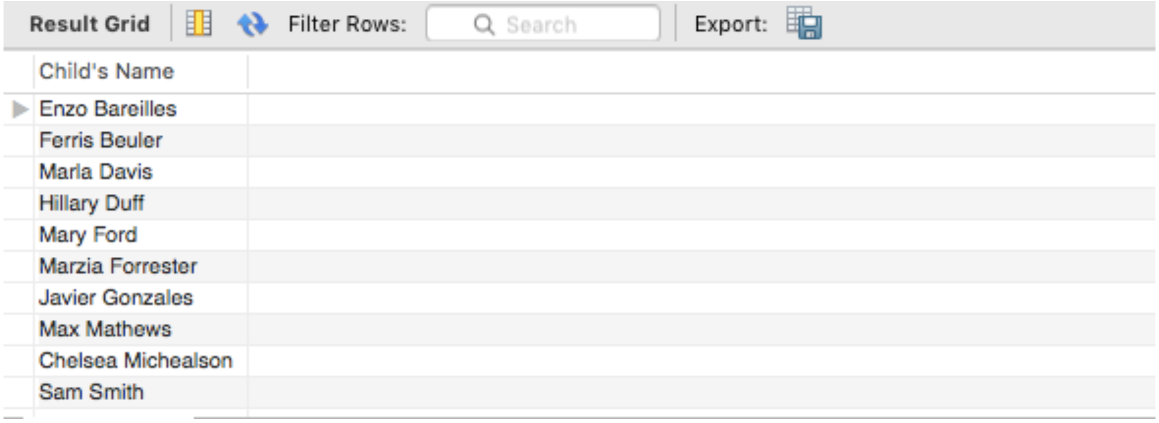
2. Identify child patients without parents in the database. Display the child patient name. Use a nested select to answer this question.

Input:



```
select
    concat(patient.pat_first, ' ', patient.pat_last) as 'Child''s Name'
from patient
where pat_id in
    (select child_id from parent_child
     where parent_id in
        (select parent_id from parent
         where parent_pat_id is null)
     or parent_id is null)
order by patient.pat_last;
```

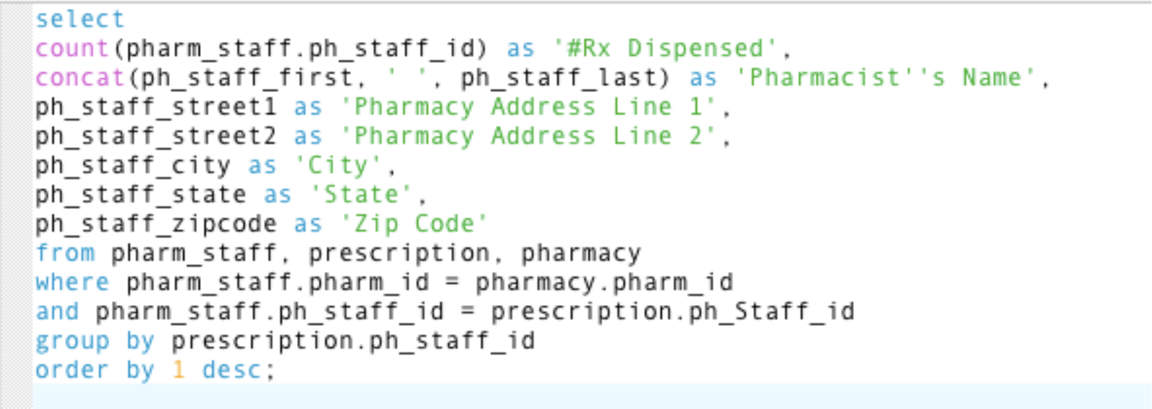
Output:



Child's Name
Enzo Bareilles
Ferris Beuler
Marla Davis
Hillary Duff
Mary Ford
Marzia Forrester
Javier Gonzales
Max Mathews
Chelsea Michealson
Sam Smith

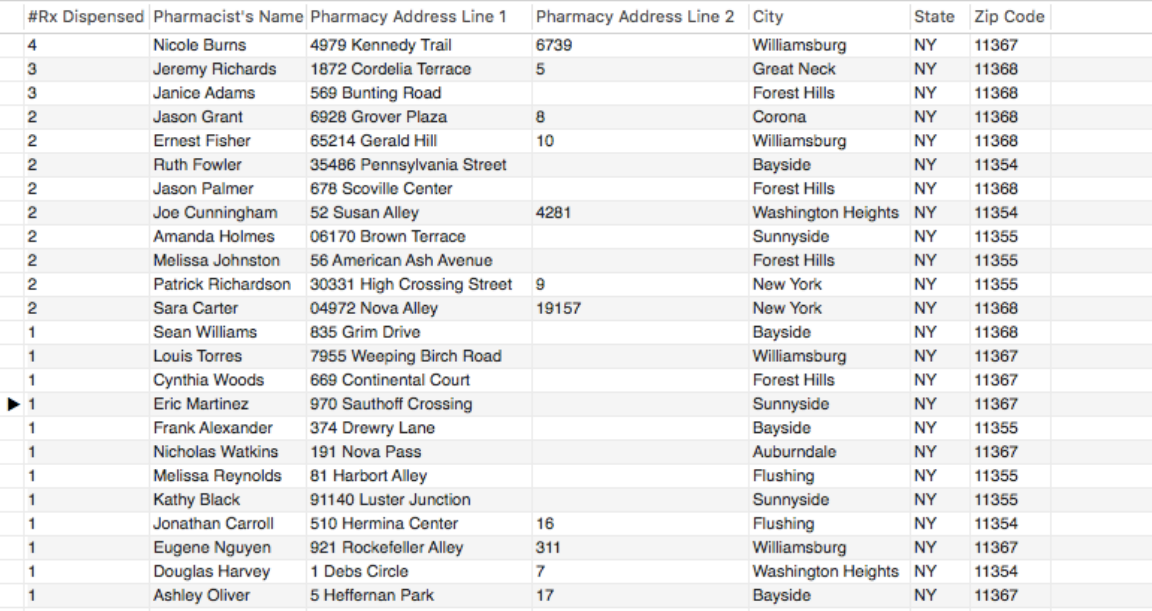
3. Identify pharmacy staff that dispensed the most prescriptions in the last year. Display the pharmacy staff name, store address and number of medications. Display one row for each pharmacy staff. The staff with the most medications will be displayed first. Use a nested select to answer this question.

Input:



```
select
count(pharm_staff.ph_staff_id) as '#Rx Dispensed',
concat(ph_staff_first, ' ', ph_staff_last) as 'Pharmacist''s Name',
ph_staff_street1 as 'Pharmacy Address Line 1',
ph_staff_street2 as 'Pharmacy Address Line 2',
ph_staff_city as 'City',
ph_staff_state as 'State',
ph_staff_zipcode as 'Zip Code'
from pharm_staff, prescription, pharmacy
where pharm_staff.pharm_id = pharmacy.pharm_id
and pharm_staff.ph_staff_id = prescription.ph_Staff_id
group by prescription.ph_staff_id
order by 1 desc;
```

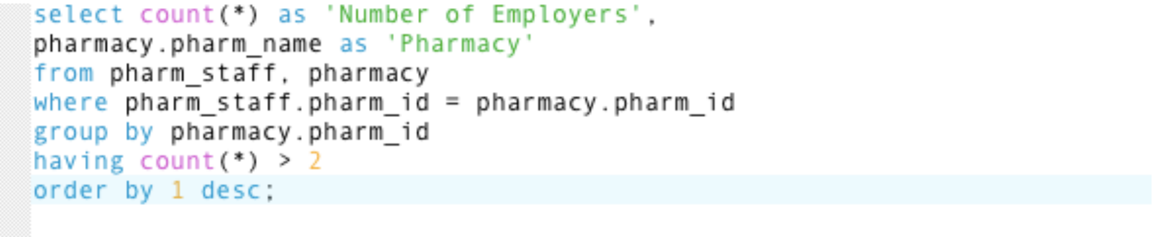
Output:



	#Rx Dispensed	Pharmacist's Name	Pharmacy Address Line 1	Pharmacy Address Line 2	City	State	Zip Code
	4	Nicole Burns	4979 Kennedy Trail	6739	Williamsburg	NY	11367
	3	Jeremy Richards	1872 Cordelia Terrace	5	Great Neck	NY	11368
	3	Janice Adams	569 Bunting Road		Forest Hills	NY	11368
	2	Jason Grant	6928 Grover Plaza	8	Corona	NY	11368
	2	Ernest Fisher	65214 Gerald Hill	10	Williamsburg	NY	11368
	2	Ruth Fowler	35486 Pennsylvania Street		Bayside	NY	11354
	2	Jason Palmer	678 Scoville Center		Forest Hills	NY	11368
	2	Joe Cunningham	52 Susan Alley	4281	Washington Heights	NY	11354
	2	Amanda Holmes	06170 Brown Terrace		Sunnyside	NY	11355
	2	Melissa Johnston	56 American Ash Avenue		Forest Hills	NY	11355
	2	Patrick Richardson	30331 High Crossing Street	9	New York	NY	11355
	2	Sara Carter	04972 Nova Alley	19157	New York	NY	11368
	1	Sean Williams	835 Grim Drive		Bayside	NY	11368
	1	Louis Torres	7955 Weeping Birch Road		Williamsburg	NY	11367
	1	Cynthia Woods	669 Continental Court		Forest Hills	NY	11367
▶	1	Eric Martinez	970 Sauthoff Crossing		Sunnyside	NY	11367
	1	Frank Alexander	374 Drewry Lane		Bayside	NY	11355
	1	Nicholas Watkins	191 Nova Pass		Auburndale	NY	11367
	1	Melissa Reynolds	81 Harbort Alley		Flushing	NY	11355
	1	Kathy Black	91140 Luster Junction		Sunnyside	NY	11355
	1	Jonathan Carroll	510 Hermina Center	16	Flushing	NY	11354
	1	Eugene Nguyen	921 Rockefeller Alley	311	Williamsburg	NY	11367
	1	Douglas Harvey	1 Debs Circle	7	Washington Heights	NY	11354
	1	Ashley Oliver	5 Heffernan Park	17	Bayside	NY	11367

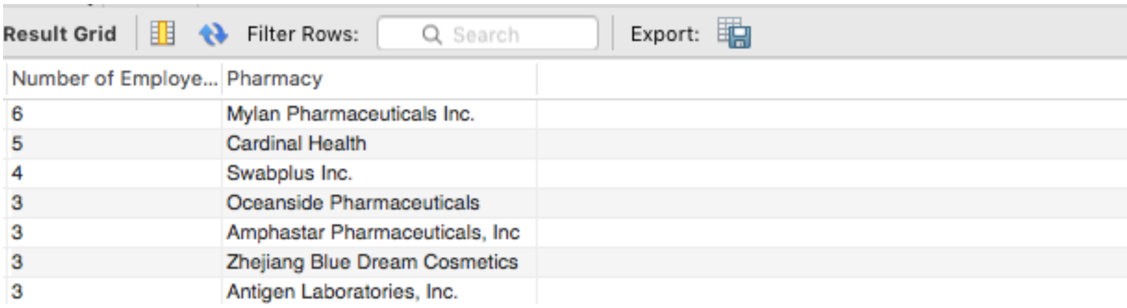
4. Identify pharmacies with more than three staff. Display the store name and number of staff. Display one row for each store. The store with the most staff will be displayed first.

Input:



```
select count(*) as 'Number of Employers',
pharmacy.pharm_name as 'Pharmacy'
from pharm_staff, pharmacy
where pharm_staff.pharm_id = pharmacy.pharm_id
group by pharmacy.pharm_id
having count(*) > 2
order by 1 desc;
```

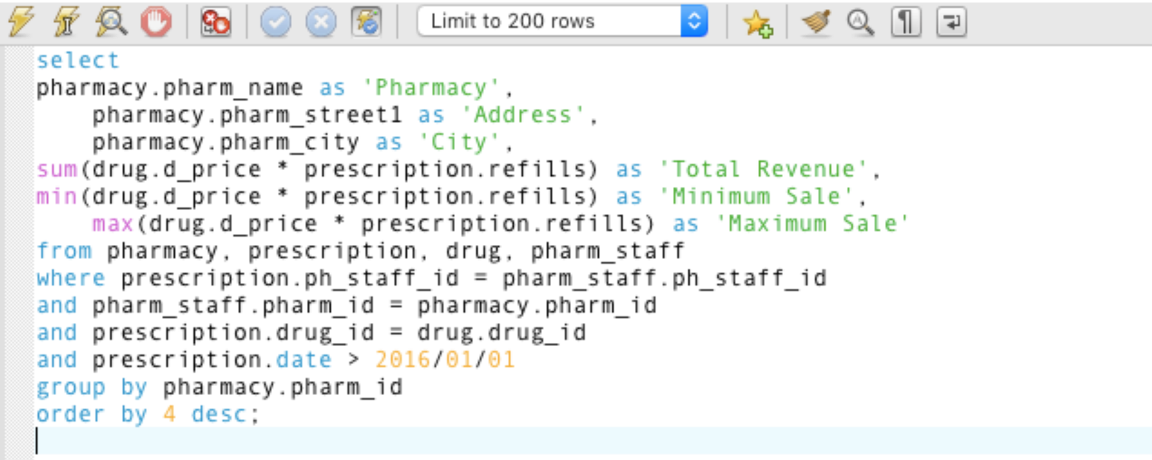
Output:



Number of Employe...	Pharmacy
6	Mylan Pharmaceuticals Inc.
5	Cardinal Health
4	Swabplus Inc.
3	Oceanside Pharmaceuticals
3	Amphastar Pharmaceuticals, Inc
3	Zhejiang Blue Dream Cosmetics
3	Antigen Laboratories, Inc.

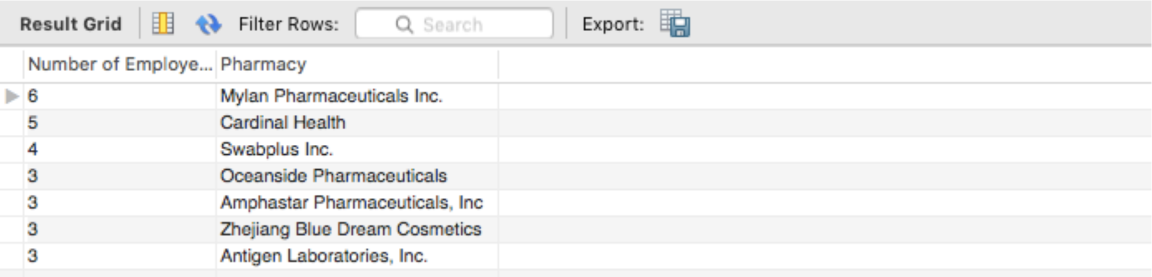
5. Identify stores with the most sales in 2016. Display one row for each store. Display the store address, city, total revenue, smallest sale and largest sale. Use functions to answer this question. The store with the highest revenue will display first.

Input



```
select
pharmacy.pharm_name as 'Pharmacy',
    pharmacy.pharm_street1 as 'Address',
    pharmacy.pharm_city as 'City',
sum(drug.d_price * prescription.refills) as 'Total Revenue',
min(drug.d_price * prescription.refills) as 'Minimum Sale',
    max(drug.d_price * prescription.refills) as 'Maximum Sale'
from pharmacy, prescription, drug, pharm_staff
where prescription.ph_staff_id = pharm_staff.ph_staff_id
and pharm_staff.pharm_id = pharmacy.pharm_id
and prescription.drug_id = drug.drug_id
and prescription.date > 2016/01/01
group by pharmacy.pharm_id
order by 4 desc;
```

Output:

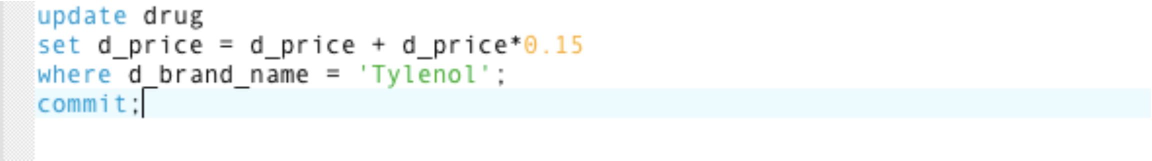


Number of Employe...	Pharmacy
6	Mylan Pharmaceuticals Inc.
5	Cardinal Health
4	Swabplus Inc.
3	Oceanside Pharmaceuticals
3	Amphastar Pharmaceuticals, Inc
3	Zhejiang Blue Dream Cosmetics
3	Antigen Laboratories, Inc.

6. Increase the price of [drug name] by [percent change] at all stores. Identify the SQL commands to perform this operation.

(for [drug name] = ‘Tylenol’ and [percent change] = 15%]

Scenario #1: If the price of ‘Tylenol’ is the same for all stores, then it is a single-value column that has functional dependency on the primary key of the table ‘Drug’. In this case, we can update the price solely in this table.

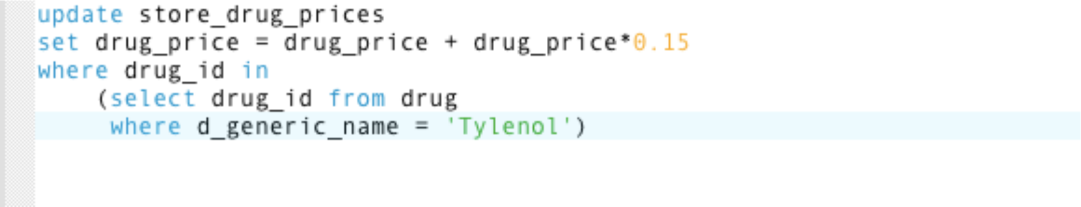


```
update drug
set d_price = d_price + d_price*0.15
where d_brand_name = 'Tylenol';
commit;
```

Scenario #2: If each store has a different price for ‘Tylenol’, another table (‘Store_Drug_Prices’) can keep track of the drug price for each store.

Store_Drug_Prices(pharm_id, drug_id, drug_price)

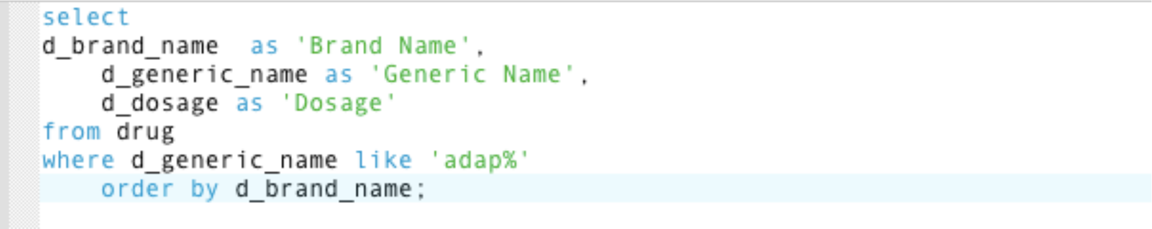
In this case, the SQL commands would be:



```
update store_drug_prices
set drug_price = drug_price + drug_price*0.15
where drug_id in
    (select drug_id from drug
     where d_generic_name = 'Tylenol')
```

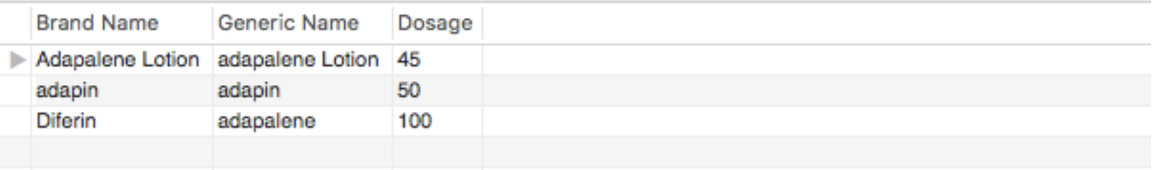
7. The pharmacist doesn't know how to spell a drug name, but the first few letters are *adap*. Identify all drugs with a similar spelling. Display the brand name, generic name and dosage.

Input:



```
select
d_brand_name  as 'Brand Name',
    d_generic_name as 'Generic Name',
    d_dosage as 'Dosage'
from drug
where d_generic_name like 'adap%'
    order by d_brand_name;
```

Output:

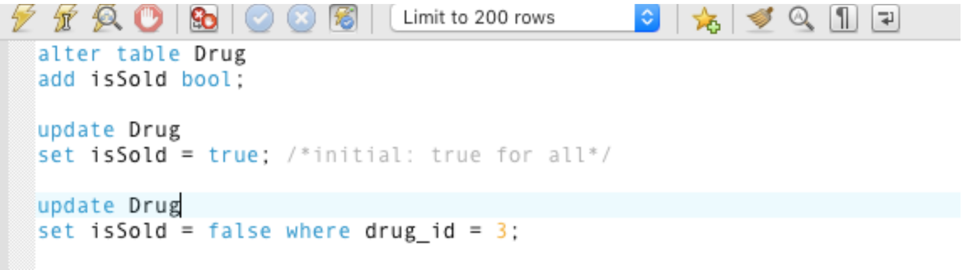


Brand Name	Generic Name	Dosage
Adapalene Lotion	adapalene Lotion	45
adapin	adapin	50
Diferin	adapalene	100

8. The drug [drug name] will no longer be sold at all stores. What is the best process to implement. Identify the SQL commands to perform this operation.

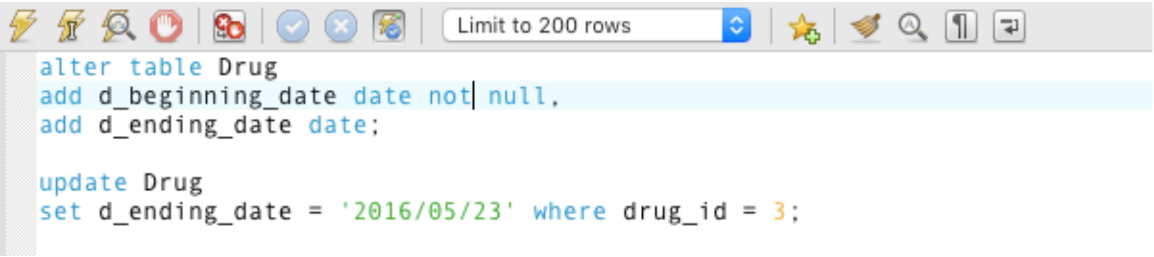
([drug name] = 'Tylenol')

If 'Tylenol' in table 'Drug' is deleted, and is referenced by other tables, then we will risk getting orphan records. In order to prevent this, I suggest creating a column in 'Drug' called 'isSold', which takes a Boolean input (T = is sold, F = is not sold). Before referencing a drug in the future, the user will check if isSold is true. For example, a pharmacy staff member will check drug.isSold before dispensing a prescription.



```
alter table Drug
add isSold bool;

update Drug
set isSold = true; /*initial: true for all*/

update Drug
set isSold = false where drug_id = 3;
```

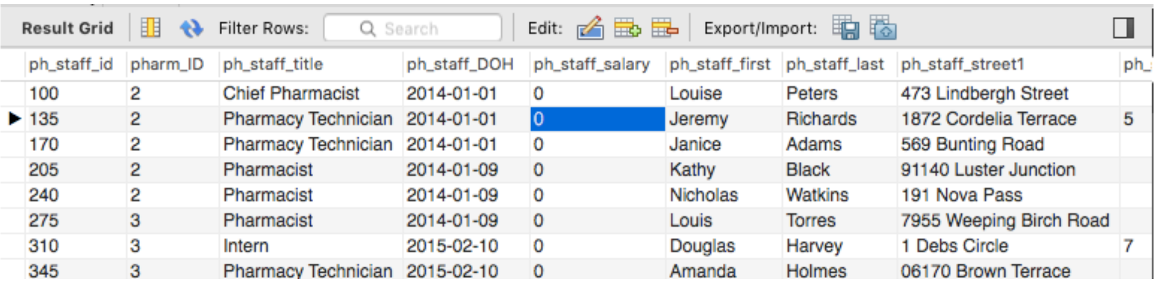
An alternative method would be to create columns 'Date_start_selling' (not null) and 'Date_Stop_Selling' (can be null), and checking to see if the latter is null. This method would also keep track of the duration, as well as the specific time the drug stopped selling. (SQL commands below)



```sql
alter table Drug
add d_beginning_date date not null,
add d_ending_date date;

update Drug
set d_ending_date = '2016/05/23' where drug_id = 3;
```

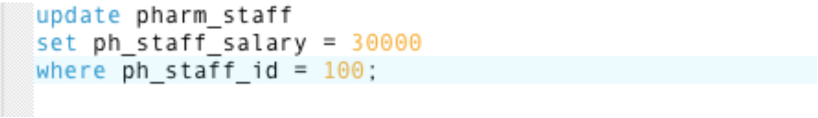
9. In one SQL window, change the staff salary for record 1. Don't commit. In another SQL window, change the staff salary for record 1. Don't commit. Resolve the problem. Disable the auto commit flag at the top of the windows before performing this operation. Explain your results.

Initial Table:



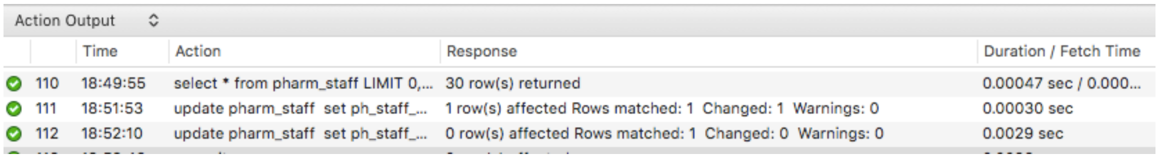
	ph_staff_id	pharm_ID	ph_staff_title	ph_staff_DOH	ph_staff_salary	ph_staff_first	ph_staff_last	ph_staff_street1	ph_
	100	2	Chief Pharmacist	2014-01-01	0	Louise	Peters	473 Lindbergh Street	
▶	135	2	Pharmacy Technician	2014-01-01	0	Jeremy	Richards	1872 Cordelia Terrace	5
	170	2	Pharmacy Technician	2014-01-01	0	Janice	Adams	569 Bunting Road	
	205	2	Pharmacist	2014-01-09	0	Kathy	Black	91140 Luster Junction	
	240	2	Pharmacist	2014-01-09	0	Nicholas	Watkins	191 Nova Pass	
	275	3	Pharmacist	2014-01-09	0	Louis	Torres	7955 Weeping Birch Road	
	310	3	Intern	2015-02-10	0	Douglas	Harvey	1 Debs Circle	7
	345	3	Pharmacy Technician	2015-02-10	0	Amanda	Holmes	06170 Brown Terrace	

Input in first window:



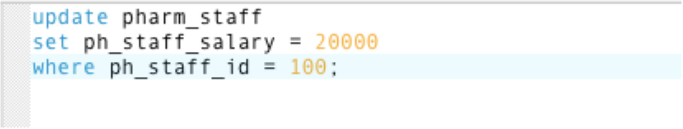
```sql
update pharm_staff
set ph_staff_salary = 30000
where ph_staff_id = 100;
```

Output in first window: *(line 111, 112)*



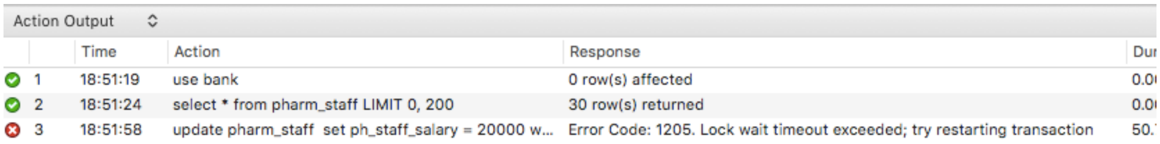
```
     Time      Action                               Response                                                   Duration / Fetch Time
✓ 110  18:49:55  select * from pharm_staff LIMIT 0,...  30 row(s) returned                                      0.00047 sec / 0.000...
✓ 111  18:51:53  update pharm_staff  set ph_staff_...   1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0   0.00030 sec
✓ 112  18:52:10  update pharm_staff  set ph_staff_...   0 row(s) affected Rows matched: 1  Changed: 0  Warnings: 0   0.0029 sec
```

Input in Second Window:



```sql
update pharm_staff
set ph_staff_salary = 20000
where ph_staff_id = 100;
```

Output in second window: (Lock wait)



```
Action Output  ⌃⌄

      Time       Action                                        Response                                                           Dur
✓  1  18:51:19   use bank                                      0 row(s) affected                                                  0.0
✓  2  18:51:24   select * from pharm_staff LIMIT 0, 200        30 row(s) returned                                                 0.0
✗  3  18:51:58   update pharm_staff  set ph_staff_salary = 20000 w...   Error Code: 1205. Lock wait timeout exceeded; try restarting transaction   50.
```

After entering the command for the first window, the result was successful and took little wait time. But when entering the command for the second window, I received an error code for ‘lock wait timeout’. There is a writelock() placed on the record, which prevents other users from accessing this record to do DML commands. The solution was to ‘commit’ in the first window, which allowed for the second window to finish the command.

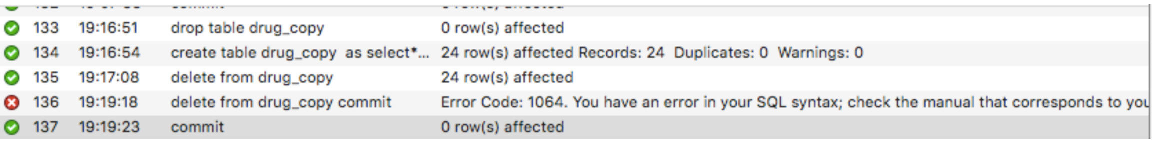
10. In one SQL window, delete all drugs. Don’t commit. In another SQL window, increase the price of all drugs by 5%. Don’t commit. Explain your results. Resolve the problem. Create a backup of your table before implementing. To create a backup table, enter CREATE TABLE <NEWTABLE> AS SELECT * FROM <ORIGINALTABLE>; COMMIT; Then you can rename a table using the RENAME TABLE commit. Disable the auto commit flag at the top of the windows before performing this operation.

I was not able to drop the primary key for ‘Drugs’. Because of this, I was able to change the drug prices by 5%.

However, when I deleted the copy of the table, and tried to increase the price by 5% in another window (for the copy of the table), I received a timeout error once again. This is because there is a lock in place for all the records in the table. I resolved by committing.

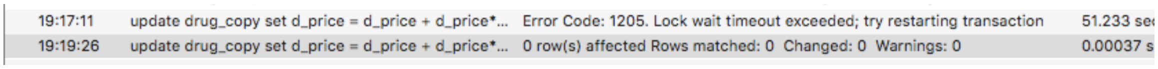
Having a primary key prevents all data from being deleted, but if a person wishes to delete all information, then it may be best to drop the table altogether.

Window 1:



```
✓  133  19:16:51   drop table drug_copy                      0 row(s) affected
✓  134  19:16:54   create table drug_copy  as select*...     24 row(s) affected Records: 24  Duplicates: 0  Warnings: 0
✓  135  19:17:08   delete from drug_copy                     24 row(s) affected
✗  136  19:19:18   delete from drug_copy commit              Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to you
✓  137  19:19:23   commit                                    0 row(s) affected
```

Window 2:



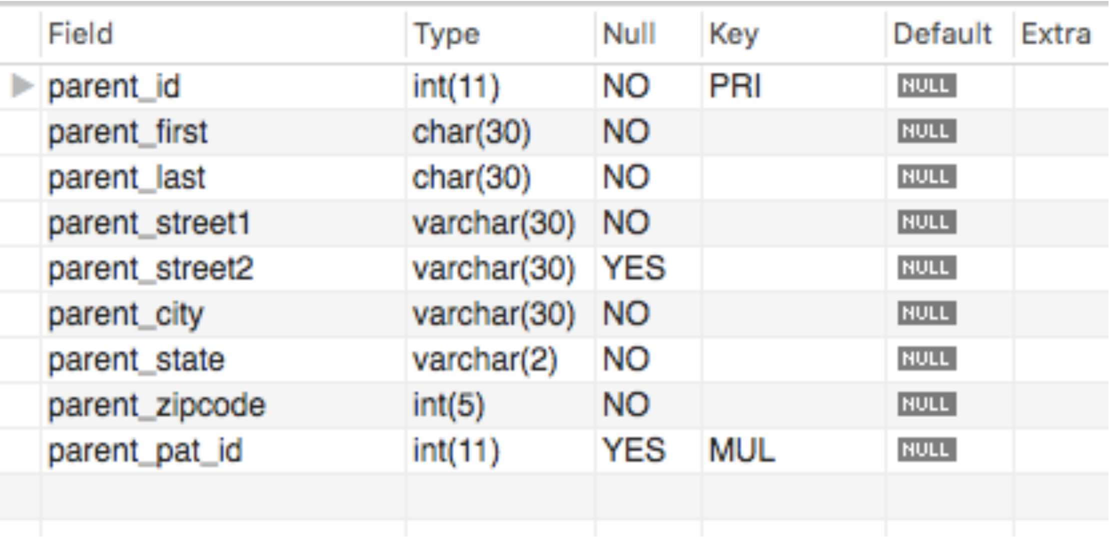
```
19:17:11   update drug_copy set d_price = d_price + d_price*...   Error Code: 1205. Lock wait timeout exceeded; try restarting transaction   51.233 se
19:19:26   update drug_copy set d_price = d_price + d_price*...   0 row(s) affected Rows matched: 0  Changed: 0  Warnings: 0                  0.00037 s
```

11. In one SQL window, null all patient addresses. Don’t commit. In another SQL window, null all patient allergies. Don’t commit. Quit both Oracle sessions. Login to Oracle and search for this information. Explain your results. Disable the auto commit flag at the top of the windows before performing this operation.

After quitting and restarting Oracle, both tables had their previous values before the null. This is because these commands were not committed, and once workbench quit, the tables were saved to the last committed values.

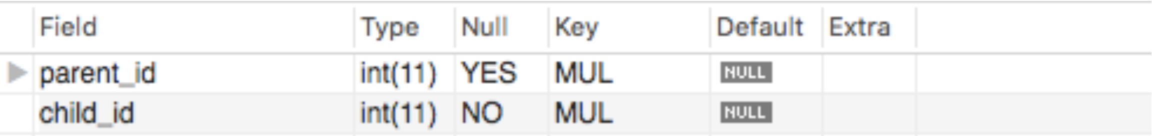
12. Use the SQL DESCRIBE operation to list the table structure for all tables.

- **Parent**



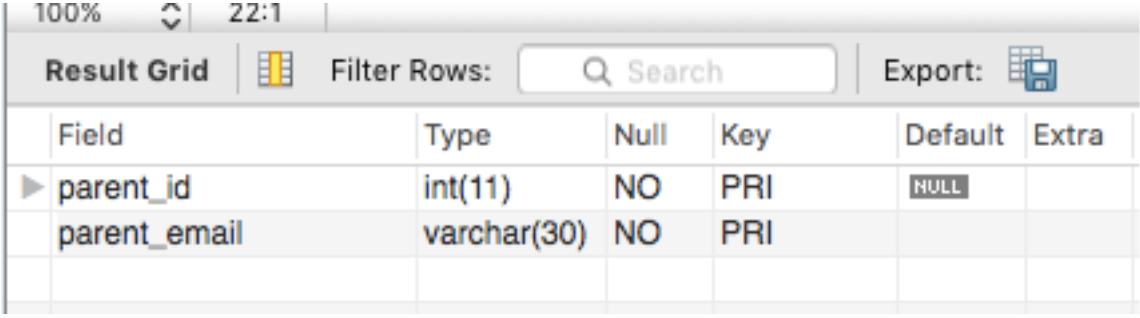
Field	Type	Null	Key	Default	Extra
parent_id	int(11)	NO	PRI	NULL	
parent_first	char(30)	NO		NULL	
parent_last	char(30)	NO		NULL	
parent_street1	varchar(30)	NO		NULL	
parent_street2	varchar(30)	YES		NULL	
parent_city	varchar(30)	NO		NULL	
parent_state	varchar(2)	NO		NULL	
parent_zipcode	int(5)	NO		NULL	
parent_pat_id	int(11)	YES	MUL	NULL	

- **Parent_child**



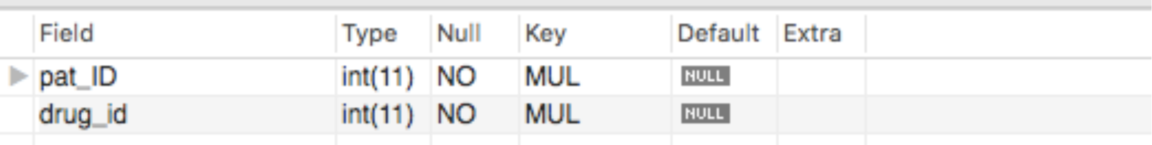
Field	Type	Null	Key	Default	Extra
parent_id	int(11)	YES	MUL	NULL	
child_id	int(11)	NO	MUL	NULL	

- **Parent_email**



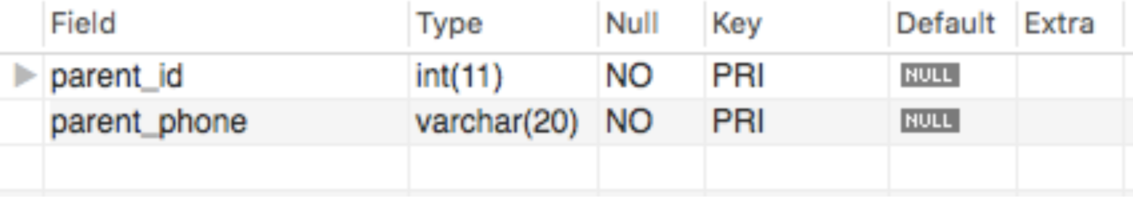
Field	Type	Null	Key	Default	Extra
parent_id	int(11)	NO	PRI	NULL	
parent_email	varchar(30)	NO	PRI		

- **Patient_Allergies**



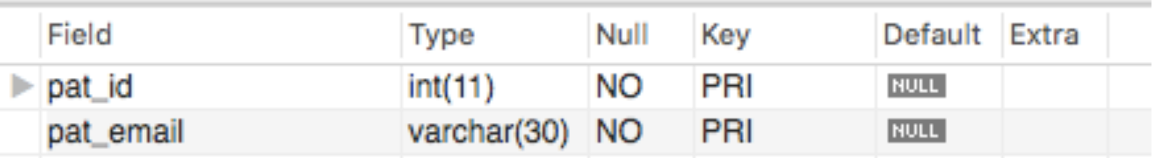
Field	Type	Null	Key	Default	Extra
pat_ID	int(11)	NO	MUL	NULL	
drug_id	int(11)	NO	MUL	NULL	

- **Parent_Phone**



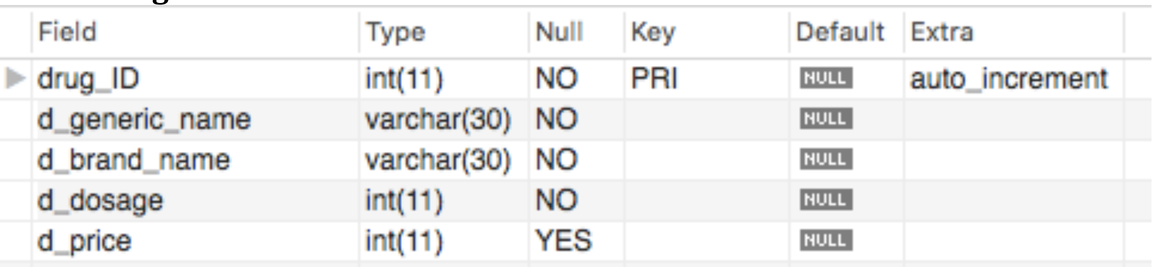
Field	Type	Null	Key	Default	Extra
parent_id	int(11)	NO	PRI	NULL	
parent_phone	varchar(20)	NO	PRI	NULL	

- **Patient_Email**



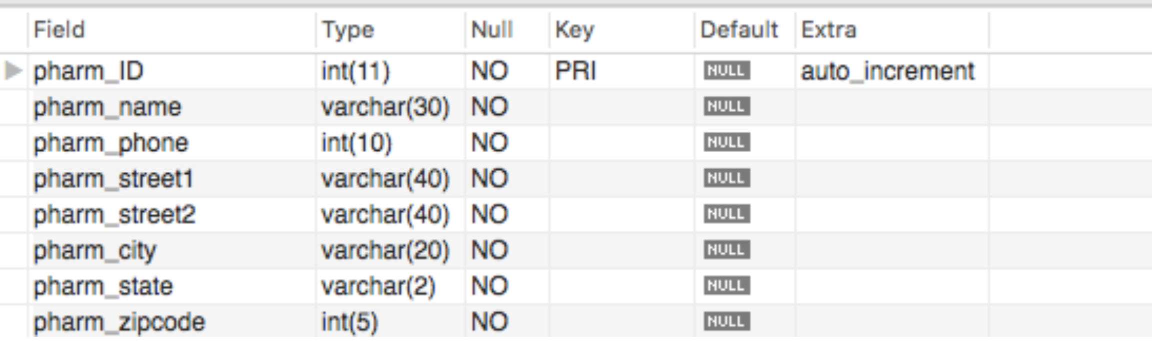
Field	Type	Null	Key	Default	Extra
pat_id	int(11)	NO	PRI	NULL	
pat_email	varchar(30)	NO	PRI	NULL	

- **Drug**



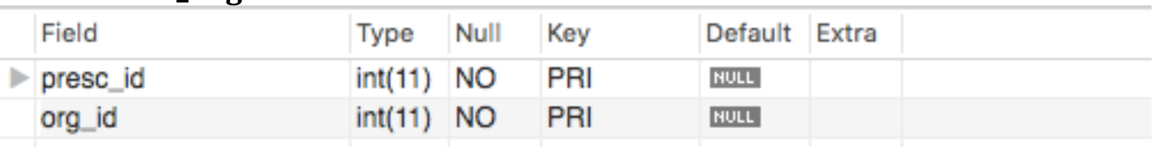
Field	Type	Null	Key	Default	Extra
drug_ID	int(11)	NO	PRI	NULL	auto_increment
d_generic_name	varchar(30)	NO		NULL	
d_brand_name	varchar(30)	NO		NULL	
d_dosage	int(11)	NO		NULL	
d_price	int(11)	YES		NULL	

- **Pharmacy**



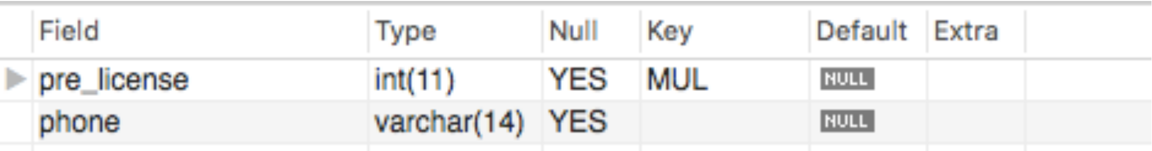
Field	Type	Null	Key	Default	Extra
pharm_ID	int(11)	NO	PRI	NULL	auto_increment
pharm_name	varchar(30)	NO		NULL	
pharm_phone	int(10)	NO		NULL	
pharm_street1	varchar(40)	NO		NULL	
pharm_street2	varchar(40)	NO		NULL	
pharm_city	varchar(20)	NO		NULL	
pharm_state	varchar(2)	NO		NULL	
pharm_zipcode	int(5)	NO		NULL	

- **Presc_Org**



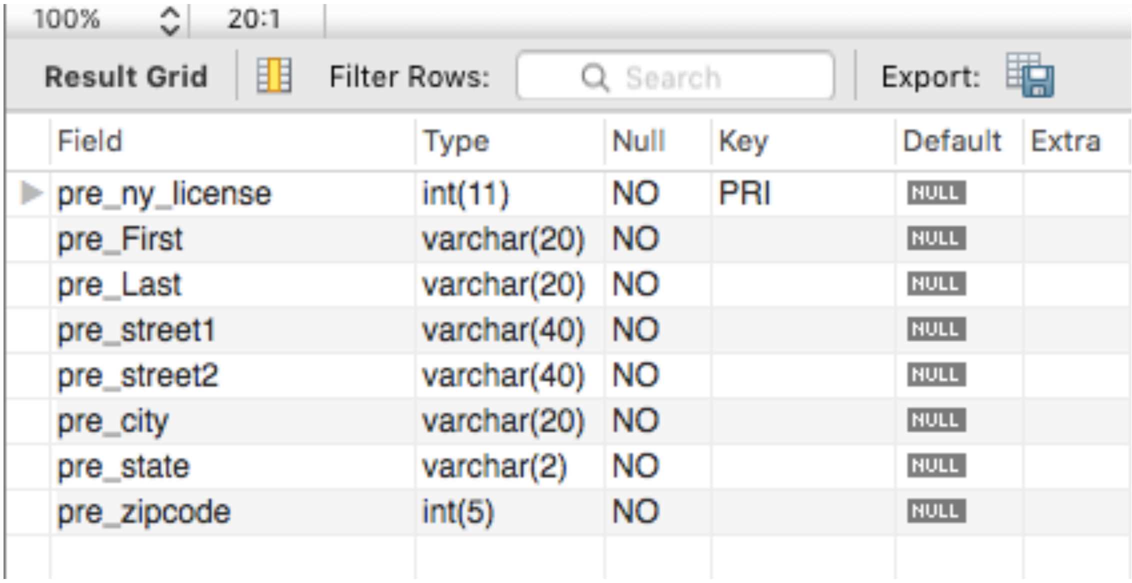
Field	Type	Null	Key	Default	Extra
presc_id	int(11)	NO	PRI	NULL	
org_id	int(11)	NO	PRI	NULL	

- **Prescriber_Phone**



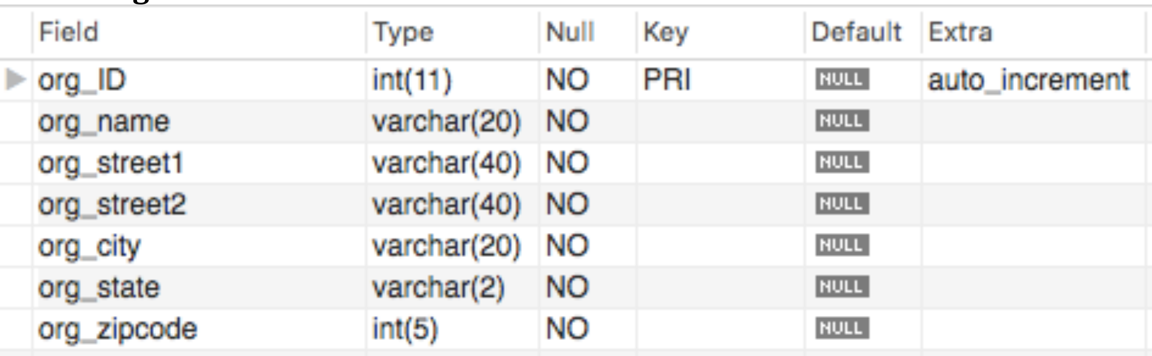
Field	Type	Null	Key	Default	Extra
pre_license	int(11)	YES	MUL	NULL	
phone	varchar(14)	YES		NULL	

- **Prescriber**



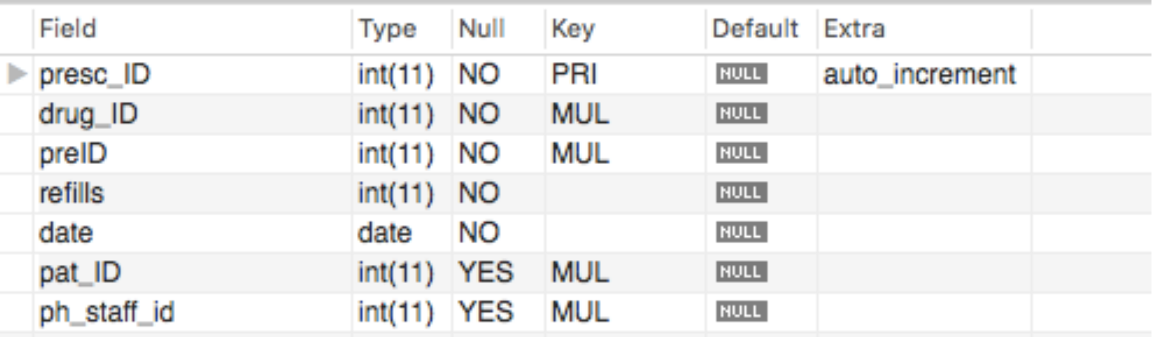
Field	Type	Null	Key	Default	Extra
pre_ny_license	int(11)	NO	PRI	NULL	
pre_First	varchar(20)	NO		NULL	
pre_Last	varchar(20)	NO		NULL	
pre_street1	varchar(40)	NO		NULL	
pre_street2	varchar(40)	NO		NULL	
pre_city	varchar(20)	NO		NULL	
pre_state	varchar(2)	NO		NULL	
pre_zipcode	int(5)	NO		NULL	

- **Organization**



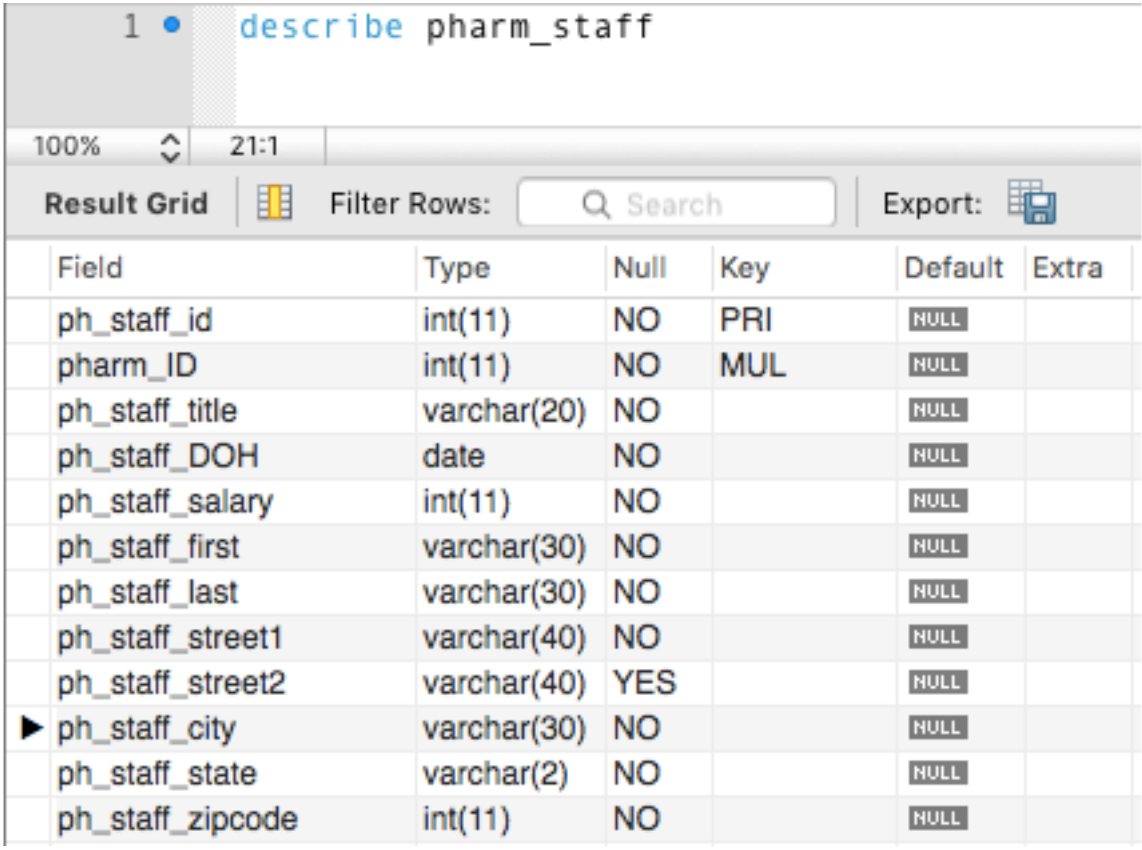
Field	Type	Null	Key	Default	Extra
org_ID	int(11)	NO	PRI	NULL	auto_increment
org_name	varchar(20)	NO		NULL	
org_street1	varchar(40)	NO		NULL	
org_street2	varchar(40)	NO		NULL	
org_city	varchar(20)	NO		NULL	
org_state	varchar(2)	NO		NULL	
org_zipcode	int(5)	NO		NULL	

- **Prescription**



Field	Type	Null	Key	Default	Extra
presc_ID	int(11)	NO	PRI	NULL	auto_increment
drug_ID	int(11)	NO	MUL	NULL	
preID	int(11)	NO	MUL	NULL	
refills	int(11)	NO		NULL	
date	date	NO		NULL	
pat_ID	int(11)	YES	MUL	NULL	
ph_staff_id	int(11)	YES	MUL	NULL	

- **Pharm_staff**



```
describe pharm_staff
```

Field	Type	Null	Key	Default	Extra
ph_staff_id	int(11)	NO	PRI	NULL	
pharm_ID	int(11)	NO	MUL	NULL	
ph_staff_title	varchar(20)	NO		NULL	
ph_staff_DOH	date	NO		NULL	
ph_staff_salary	int(11)	NO		NULL	
ph_staff_first	varchar(30)	NO		NULL	
ph_staff_last	varchar(30)	NO		NULL	
ph_staff_street1	varchar(40)	NO		NULL	
ph_staff_street2	varchar(40)	YES		NULL	
ph_staff_city	varchar(30)	NO		NULL	
ph_staff_state	varchar(2)	NO		NULL	
ph_staff_zipcode	int(11)	NO		NULL	