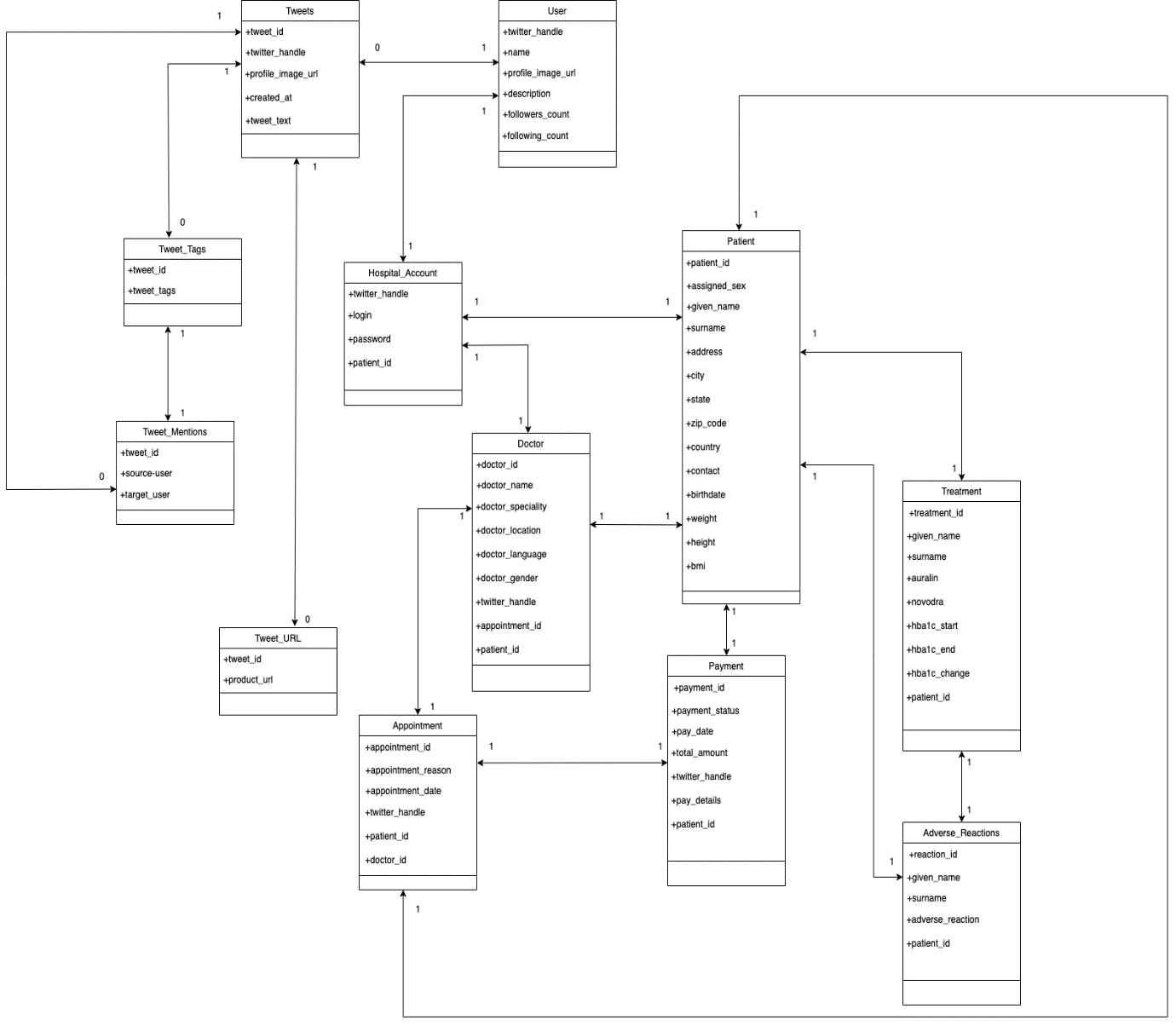
**DMDD ASSIGNMENT- 2**

**Anuja Shinde (002747948)**

**Anuja Kale (002700699)**

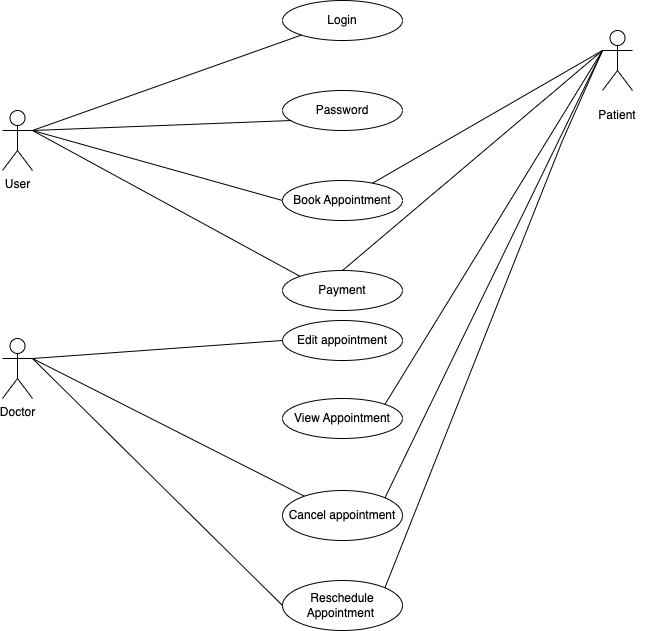
**Entity Relationship Diagram:**

****

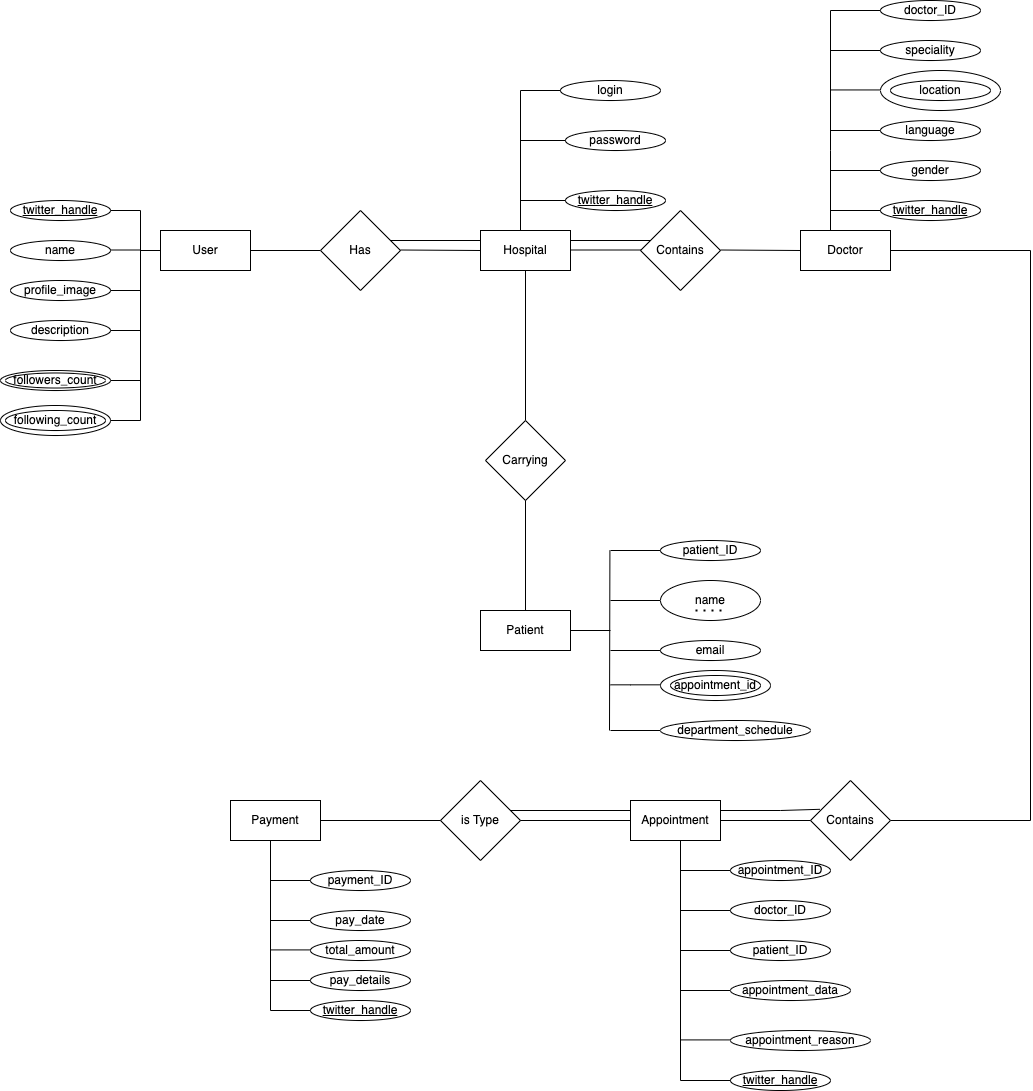
**Explanation of some of the design decisions:**

* The Brigham and Women's Hospital account has a login and password. This login is the same as a user’s Twitter handle. The Twitter handle is unique – hence it can also be treated as the primary key of the table.
* Each user can tweet any number of tweets. The Brigham and Women's Hospital -user admin user of the Brigham and Women's Hospital is also one of the users and this information can be stored in the patient table itself.
* A patient can make an appointment through Twitter by tweeting the appointment detail and mentioning the doctor's account URL.
* ‘Payment’ has the ‘payment\_id’ of the tweet which uniquely distinguishes each tweet, ‘appointment-url’ which is a foreign key reference to the ‘appointment\_url’ in the ‘Tweet\_Url’ table, ‘appointment\_id’ which is a foreign key reference to ‘appointment\_id’ in ‘Women\_Appointments’ table corresponding to the particular ‘appointment\_url’ mentioned by a patient in the tweet.
* A patient can tweet how many ever appointments he/she wants. Hence the ‘payment’ has a ‘payment\_id’ which is the primary key of the table (since it uniquely distinguishes each order). Note that each appointment in the payment can have more than one Twitter appointment.

**Use Case Diagram:**

****

**UML Diagram:**

****

**Use Cases:**

**1. Use Case:** Register for an account at Brigham and Women's Hospital

**Description:** The user registers for an account in Brigham and Women's Hospital

**Actor:** User

**Precondition:** When a patient wants to book an appointment in the hospital, firstly he will be registered

**Actor action:** User request for registration

**System Responses:** If patient information is correct then the patient is registered and the use case ends.

**Post Condition:** Patient successfully registered

**Alternate Path:** The customer request is not valid and the system throws an error

**Error:** User information is incorrect

**2. Use Case:** Make an appointment at Brigham and Women's Hospital

**Description:** The patient makes an appointment at Brigham and Women's Hospital

**Actors:** Patient

**Precondition:** The patient must have a unique Twitter handle to tweet

**Actor action –** Patient tweets about an appointment along with the doctor's URL

**System Responses –** An appointment is made for the patient that matches the appointment reason with respect to that particular qualified doctor's URL

**Post Condition:** An appointment is added to the department\_schedule table for the appointment the user tweeted.

**Alternate Path:** The appointment is not currently available in the Brigham and Women's Hospital

**Error:** Appointment Not Available

**3. Use Case:** View an appointment already booked through Twitter by a patient

**Description:** The patient views an appointment already booked

**Actors:** Patient

**Precondition:** The patient must have made an appointment

**Actor action –** The patient views an appointment from its URL

**System Responses –** appointment URL would be displayed

**Post Condition:** system displays the appointment URL

4. **Use Case:** View the appointments above a particular price (say $100)

**Description:** Use view the appointment above a particular price

**Actor:** Patient

**Precondition:** The patient must have made an appointment

**Actor action:** The patient views the appointment above a particular price

**System Responses:** the list of appointments above a price is displayed

**Post Condition:** system displays the list of appointments for the condition

5. **Use Case:** View the appointments made by a patient

**Description:** The patient views the appointments made by him/her

**Actor:** Patient

**Precondition:** The patient must have made at least one appointment to view an appointment

**Actor action:** The patient views the history of the appointment

**System Responses:** Displays all the appointments made by a patient

**Alternate Path:** There are no appointments made by a patient

**Error:** No history of appointments available.

**6**. **Use Case:** Cancel the appointments made by a patient

**Description:** The patient can cancel the appointments made by him/her

**Actor:** Patient

**Precondition:** The patient must have made at least one appointment to cancel an appointment

**Actor action:** The patient can view cancel history of the appointment

**System Responses:** Displays all the cancel appointments made by a patient

**Alternate Path:** There are no appointments cancelled by a patient

**Error:** No history of cancelled appointments available.

**7**. **Use Case:** Reschedule the appointments made by a patient

**Description:** The patient can reschedule the appointments made by him/her

**Actor:** Patient

**Precondition:** The patient must have made at least one appointment to reschedule an appointment

**Actor action:** The patient can view reschedule history of the appointment

**System Responses:** Displays all the rescheduled appointments made by a patient

**Alternate Path:** There are no appointments rescheduled by a patient

**Error:** No history of rescheduled appointments available.

**8**. **Use Case:** Doctor can edit the appointments made by a patient

**Description:** The doctor can edit the appointments made by him/her

**Actor:** Doctor

**Precondition:** The doctor must have made at least one appointment to edit an appointment

**Actor action:** The doctor can view edit history of the appointment

**System Responses:** Displays all the edited appointments made by a patient

**Alternate Path:** There are no appointments edited by a doctor

**Error:** No history of edited appointments available.

**9**. **Use Case:** View the payment details made by the patient

**Description:** The patient can view the payment details made by him/her

**Actor:** Patient

**Precondition:** The patient must have made at least one payment details to view an appointment

**Actor action:** The patient can view payment details of the appointment

**System Responses:** Displays all the payment details made by a patient

**Alternate Path:** There are no payment details by a patient

**Error:** No history of payment details available.

**10**. **Use Case:** Patient can find a doctor based on his/her specialization

**Description:** The patient can book an appointment on the basis of doctor specialization **Actor:** Patient

**Precondition:** The patient must book appointment based of his/her health issues

**Actor action:** The patient can view specialized doctors

**System Responses:** Displays all the specialized doctors for a patient

**Alternate Path:** There are no specialized doctor’s

**Error:** No history of specialized doctors available.

**RELATIONAL-ALGEBRA EXPRESSIONS FOR THE USE CASES**

**1.Use Case: View an appointment already ordered through Twitter**

Π{w.appointment\_url}(σ{w.appointment\_id = t.appointment\_id  ∧  t.Twitter\_handle = '@anna' }(ρ{w}(Women\_Appointment)  ×  ρ{t}(Twitter\_Order)))

**2.Use Case: View the products above a particular price (say $100)**

Π{w.appointment\_url, w.appointment\_url}(σ{w.price > 100 }(Women\_Appointment))

**3.Use Case: View the orders made by a user**

Π{s.Twitter\_handle, s.appointment\_id}(σ{s.Twitter\_handle = '@emma' }(Payment))

**SQL STATEMENTS**

1. Use Case: Register for an appointment at Brigham and Women's Hospital

INSERT INTO Hospital\_Account

(Twitter\_handle, login, password)

VALUES (@anna, anna123, xxxxxx)

INSERT INTO Hospital\_Account

(Twitter\_handle, login, password)

VALUES (@emily, emily123, xxxxxx);

INSERT INTO Hospital\_Account

(Twitter\_handle, login, password)

VALUES (@emma, emma123, xxxxxx);

2. Use Case: Make an appointment at Brigham and Women's Hospital

INSERT INTO Tweets

(tweet\_id, Twitter\_handle, tweet\_text, profile\_image\_url, created\_at )

VALUES (**12321**, @anna, ‘I would like to book  appointment https://www.brighamandwomens.org//product\_id=**2449**’ , ‘www.facebook.com/emma.smith/ photo.php?fbid=**10205**’ , **12**-**11**-**2022**  );

INSERT INTO Tweet\_url

(tweet\_id, hospital\_url )

VALUES (**12321**,‘https://www.brighamandwomens.org//product\_id=**2449**’);

INSERT INTO Twitter\_Appointment

(appointment\_ID, patient\_ID, doctor\_ID, appointmet\_date, appointment\_reason, twitter\_handle)

VALUES (**4532**, **12321**, @john, **2341**, ,13/11/2022, **Fever, emma123** )

INSERT INTO Payment

(payment\_id, payment\_date, total\_amount, payment\_details, appoinment\_ID Twitter\_handle)

VALUES ( **9876**, **13/11/2022**, **1**, $**78.4**,2453, @emma123 )

3. Use Case: View an appointment already ordered through Twitter

SELECT w.appointment\_url

FROM Women\_Appointment w, Twitter\_Appointment t

WHERE

t.appointment\_id = w.appointment\_id AND

t.Twitter\_handle = '@emaa123'

4. Use Case: View the appointments above a particular price (say $100)

SELECT w.patient\_name, w.appointment\_url

FROM Women\_Appointments w

WHERE

w.price > **100**;

5. Use Case: View the appointments made by a patient

SELECT s.Twitter\_handle, s.patient\_id

FROM Payment s

WHERE

s.Twitter\_handle = 'anna';

**SQL Statements for the conceptual model:**

**User Table:**

CREATE TABLE `User` (

  `Twitter\_handle` VARCHAR(10),

  `name` VARCHAR(20),

  `profile\_image\_url` VARCHAR(200),

  `description` VARCHAR(100),

  `followers\_count` INT,

  `following\_count` INT,

  PRIMARY KEY  (`Twitter\_handle`)

);

**Tweets Table:**

CREATE TABLE `Tweets` (

  `tweet\_id` INT NOT NULL AUTO\_INCREMENT,

  `Twitter\_handle` VARCHAR(10),

  `tweet\_text` VARCHAR(140),

  `profile\_image\_url` VARCHAR(200),

  `created\_at` DATETIME,

  PRIMARY KEY  (`tweet\_id`)

);

**Tweet\_Tags Table:**

CREATE TABLE `Tweet\_Tags` (

  `tweet\_id` INT NOT NULL,

  `tags` VARCHAR(20),

  PRIMARY KEY  (`tweet\_id`)

);

**Tweet\_Tags Table:**

CREATE TABLE `Tweet\_Tags` (

  `tweet\_id` INT NOT NULL,

  `tags` VARCHAR(20),

  PRIMARY KEY  (`tweet\_id`)

);

**Tweet\_Mentions Table:**

CREATE TABLE `Tweet\_Mentions` (

  `tweet\_id` INT NOT NULL,

  `source\_user` VARCHAR(10),

  `target\_user` VARCHAR(10),

  PRIMARY KEY  (`tweet\_id`)

);

**Tweet\_Url Table:**

CREATE TABLE `Tweet\_Url` (

  `tweet\_id` INT NOT NULL,

  ‘hospital\_url’ VARCHAR(200),

  PRIMARY KEY  (`tweet\_id`)

);

**Hospital\_Account Table:**

CREATE TABLE `Hospital\_Account` (

  `Twitter\_handle` VARCHAR(10) NOT NULL,

  `password` VARCHAR(10),

  ‘login’ VARCHAR(10),

  PRIMARY KEY  (`Twitter\_handle`)

);

**Twitter\_Appoinment Table:**

CREATE TABLE `Twitter\_Appointment` (

  `appointment\_id` **INT** NOT NULL AUTO\_INCREMENT,

  `doctor\_id` INT NOT NULL,

  `patient\_id` INT NOT NULL,

  `appointment\_date` INT NOT NULL,

  ‘appointment\_reason’ VARCHAR(10),

  `Twitter\_handle` VARCHAR(10) NOT NULL,

  PRIMARY KEY  (`appointmentt\_id`)

);

**Patient Table:**

CREATE TABLE `Patient` (

  `patient\_id` INT NOT NULL AUTO\_INCREMENT,

  `department\_schedule` VARCHAR,

  `product\_id` INT,

  ‘name’ VARCHAR(20),

  ‘phone’ INT,

  ‘email’ VARCHAR(20),

`appointment\_id` INT,

  PRIMARY KEY  (`patient\_id`)

);

**Payment Table:**

CREATE TABLE `Payment` (

  `payment\_id` INT NOT NULL AUTO\_INCREMENT,

  `total\_amount` FLOAT,

  `Twitter\_handle` VARCHAR(10),

  `pay\_details` INT,

  `pay\_date` INT,

  PRIMARY KEY  (`payment\_id`)

);

**Doctor Table:**

CREATE TABLE `Doctor` (

  `doctor\_id` INT NOT NULL AUTO\_INCREMENT,

  `speciality` VARCHAR(20),

  `Twitter\_handle` VARCHAR(10),

  `location` VARCHAR(20),

  `language` VARCHAR(10),

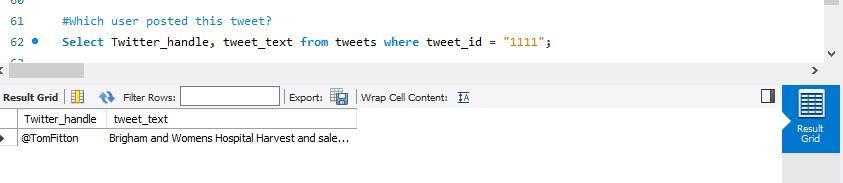
   `gender` VARCHAR(10),

  PRIMARY KEY  (`doctor\_id`)

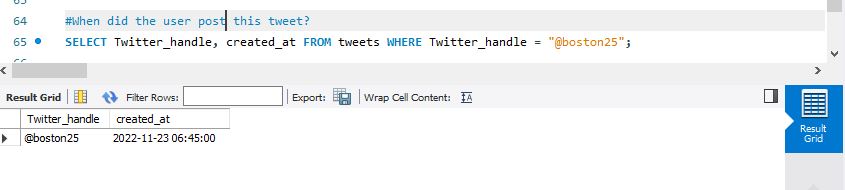
);

**Database SQL STATEMENTS**

1. Which user posted this tweet?

****

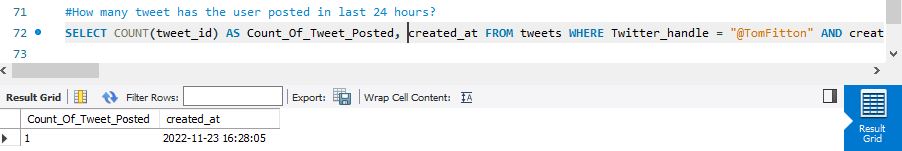
1. When did the user post this tweet?

****

1. What tweet has the user posted in last 24 hours?

****

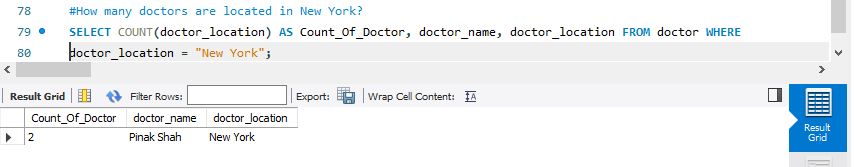
1. How many tweets has the user posted in last 24 hours?

****

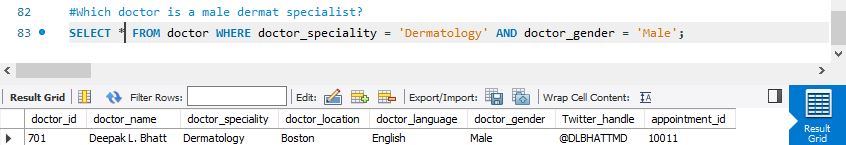
1. How many users have the followers more than 3 million?

****

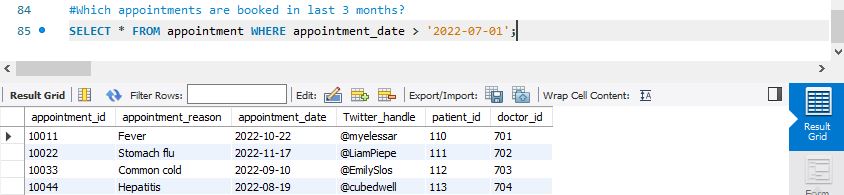
1. How many doctors are located in New York?

****

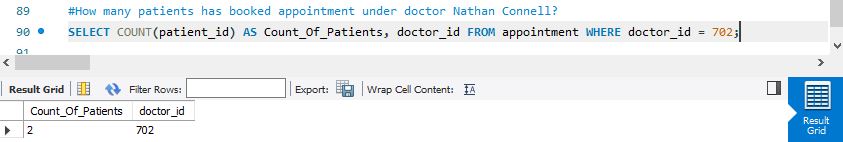
1. Which doctor is a male dermat specialist?

****

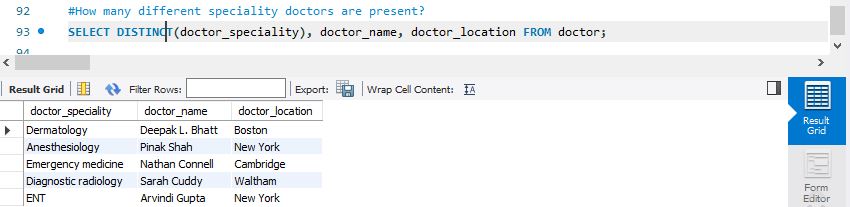
1. Which appointments are booked in last 3 months?

****

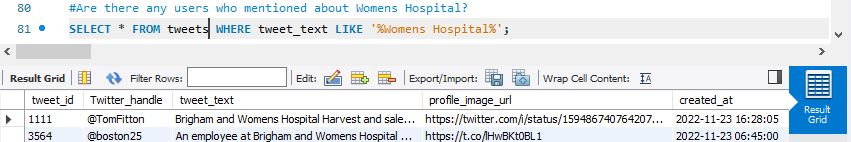
1. How many patients have booked appointment under doctor Nathan Connell?

****

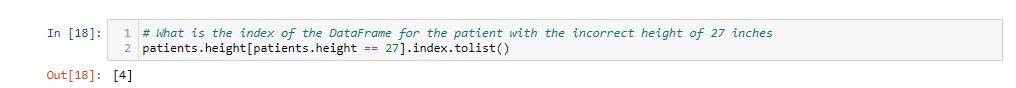
1. How many different speciality doctors are present?

****

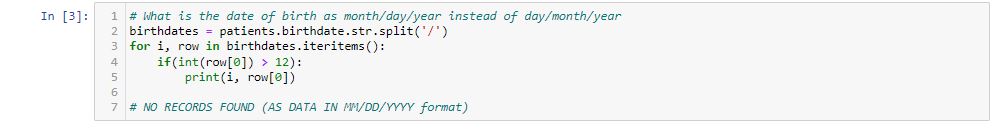
1. Are there any users who mentioned about Women Hospital?

****

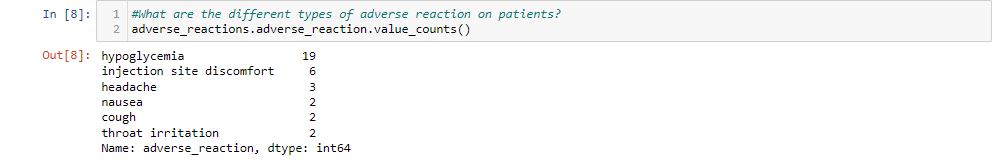
12. What is the index of the DataFrame for the patient with the incorrect height of 27 inches?



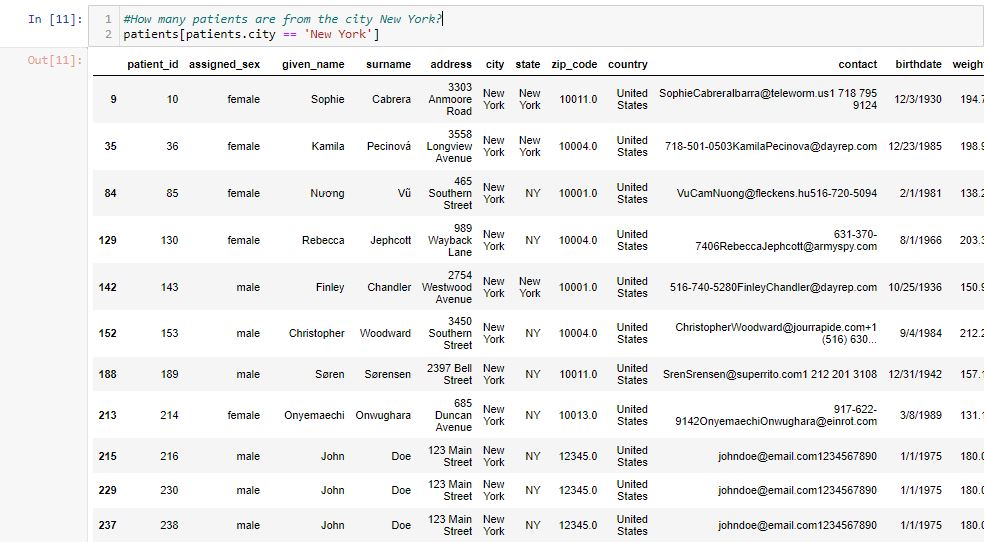
13. What is the date of birth as month/day/year instead of day/month/year?



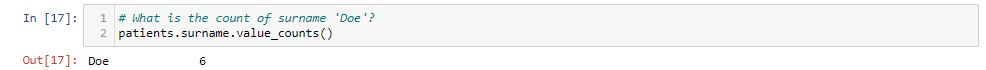
14. What are the different types of adverse reaction on patients?



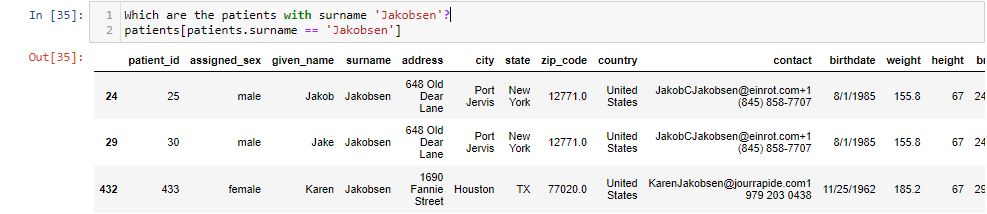
15. How many patients are from the city New York?



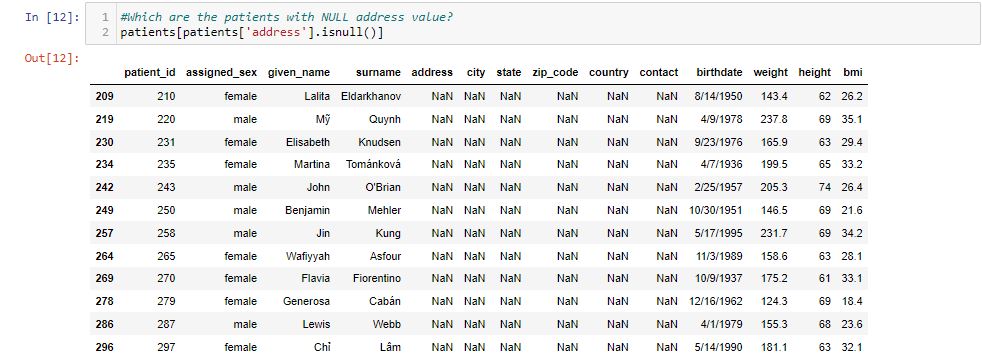
16. What is the count of Surname ‘Doe’?



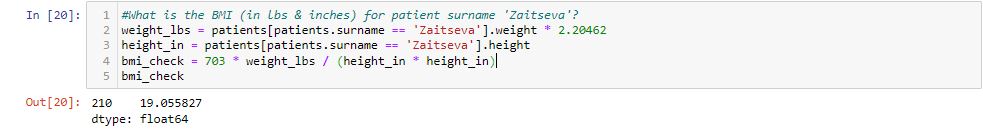
17. Which are the patients with surname 'Jakobsen'?



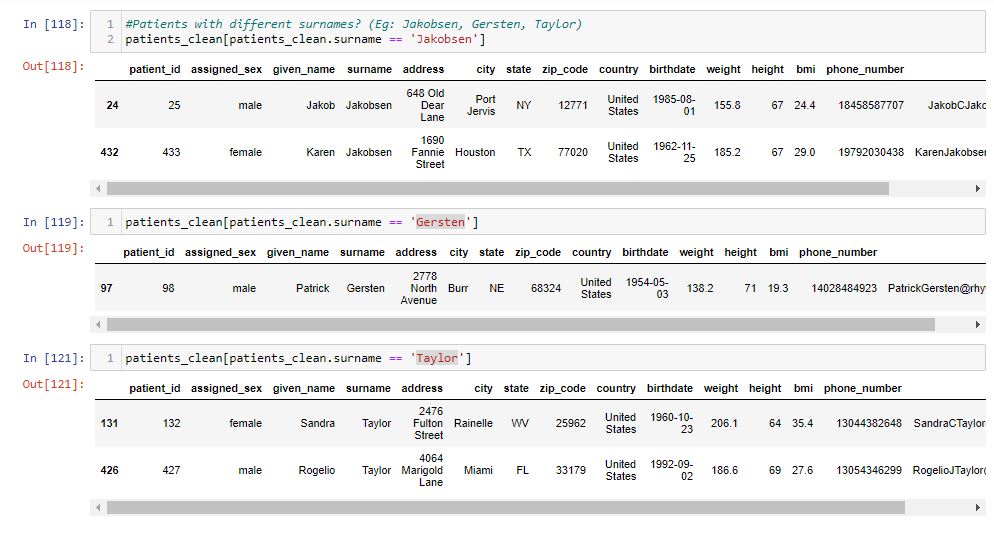
18. Which are the patients with NULL address value?



19. What is the BMI (in lbs & inches) for patient surname 'Zaitseva'?



20. Patients with different surnames? (Eg: Jakobsen, Gersten, Taylor)



**UPDATED CONCEPTUAL MODEL (SQL STATEMENTS)**

CREATE DATABASE hospital;

**Twitter Appointment**

CREATE TABLE `Twitter\_Appointment` ( `appointment\_id` INT NOT NULL AUTO\_INCREMENT, `appointment\_reason` VARCHAR(100), `appointment\_date` date, `Twitter\_handle` VARCHAR(50) NOT NULL, `doctor\_id` INT NOT NULL, patient\_id INT, PRIMARY KEY (`appointment\_id`));

INSERT INTO twitter\_appointment (appointment\_id, appointment\_reason, appointment\_date, Twitter\_handle, doctor\_id, patient\_id) VALUES (10011, 'Fever', '2022-10-22', '@myelessar', 701, 801), (10022, 'Stomach flu', '2022-11-17', '@LiamPieper', 702, 802), (10033, 'Common cold', '2022-09-10', '@EmilySlosberg', 703, 804), (10044, 'Hepatitis', '2022-08-19', '@cubedweller', 704, 805);

UPDATE twitter\_appointment SET doctor\_id = 702 WHERE appointment\_id = 1…

[4:57 pm, 05/12/2022] anuja kale: CREATE TABLE `User` (`Twitter\_handle` VARCHAR(50), `name` VARCHAR(50), `profile\_image\_url` VARCHAR(200), `description` VARCHAR(200), `followers\_count` INT, `following\_count` INT, PRIMARY KEY (`Twitter\_handle`));

INSERT INTO user VALUES ('@TomFitton', 'Tom Fitton', 'https://twitter.com/TomFitton/photo', 'President, Judicial Watch. Fact checker. "Expert" (These are my personal views only!) LATEST BEST SELLER BOOK: A Republic Under Assault: http://judicialwatchbook.com', 200000000, 3610);

UPDATE user SET followers\_count = 3320060 WHERE Twitter\_handle = '@TomFitton';

**Patients**

CREATE TABLE `patients`(`patient\_id` INT NOT NULL AUTO\_INCREMENT, `assigned\_sex` VARCHAR(50), `given\_name` VARCHAR(50), `surname` VARCHAR(50), `address` VARCHAR(100), `city` VARCHAR(50), `state` VARCHAR(20), `zip\_code` INT, `country` VARCHAR(20), `contact` VARCHAR(100), `birthdate` VARCHAR(20), `weight` FLOAT, `height` INT, `bmi` FLOAT, PRIMARY KEY (`patient\_id`));

INSERT INTO patients(patient\_id, assigned\_sex, given\_name, surname, address, city, state, zip\_code, country, contact, birthdate, weight, height, bmi) VALUES(225, 'female', 'Veronika', 'Jindrová', '3641 Selah Way', 'Brattleboro', 'VT', 5301, 'United States', '802-614-0812VeronikaJindrova@jourrapide.com', '1/15/1981', 171.4, 62, 31.3), (242, 'female', 'Skye', 'Gormanston', '936 Lightning Point Drive', 'Arlington', 'TN', 38002, 'United States', '901-484-5225SkyeGormanston@dayrep.com', '1/20/1934', 139.7, 60, 27.3), (345, 'female', 'Sophia', 'Haugen', '4178 Despard Street', 'Atlanta', 'GA', 30303, 'United States', '404-713-3641SophiaHaugen@dayrep.com', '6/4/1939', 181.1, 63, 32.1), (276, 'male', 'Eddie', 'Archer', '2043 Jadewood Drive', 'Lombard', 'Illinois', 60148, 'United States', 'EddieAArcher@gustr.com+1 (224) 305-6805', '7/17/1982', 158.6, 69, 23.4), (15, 'female', 'Asia', 'Woźniak', '4970 Heather Sees Way', 'Tulsa', 'OK', 74105, 'United States', 'AsiaWozniak@rhyta.com918-712-3469', '8/15/1997', 112, 65, 18.6),(128, 'female', 'Lena', 'Baer','272 Boone Crockett Lane', 'Elma', 'WA', 98541, 'United States', '360-482-2553LenaBaer@rhyta.com', '11/10/1977', 170.7, 61, 32.2);

**Treatments**

CREATE TABLE `treatments`(`treatment\_id` INT , `given\_name` VARCHAR(20), `surname` VARCHAR(20), `hba1c\_start` FLOAT, `hba1c\_end` FLOAT, `hba1c\_change` FLOAT , `treatment` VARCHAR(20), `dose` VARCHAR(20), `dose\_start` VARCHAR(20), `dose\_end` VARCHAR(20), patient\_id INT, PRIMARY KEY (`treatment\_id`) );

INSERT INTO treatments(treatment\_id, given\_name, surname, hba1c\_start, hba1c\_end, hba1c\_change, treatment, dose,dose\_start, dose\_end, patient\_id) VALUES(900,'veronika', 'jindrová', 7.63, 7.20, 0.43, 'auralin', '41u-48u', '41u', '48u', 225), (901,'skye', 'gormanston', 7.97, 7.62, 0.35, 'auralin', '33u-36u', '33u', '36u', 242), (902,'sophia', 'haugen', 7.65, 7.27, 0.38, 'auralin', '37u-42u', '37u', '42u', 345),(903,'eddie', 'archer', 7.89, 7.55, 0.34, 'auralin', '31u-38u', '31u', '38u', 276), (904,'asia', 'woźniak', 7.76, 7.37, 0.39, 'auralin', '30u-36u', '30u', '36u', 15),(905, 'lena', 'baer', 7.7, 7.4, 0.3, 'auralin', '41u-38u', '41u', '38u',128);

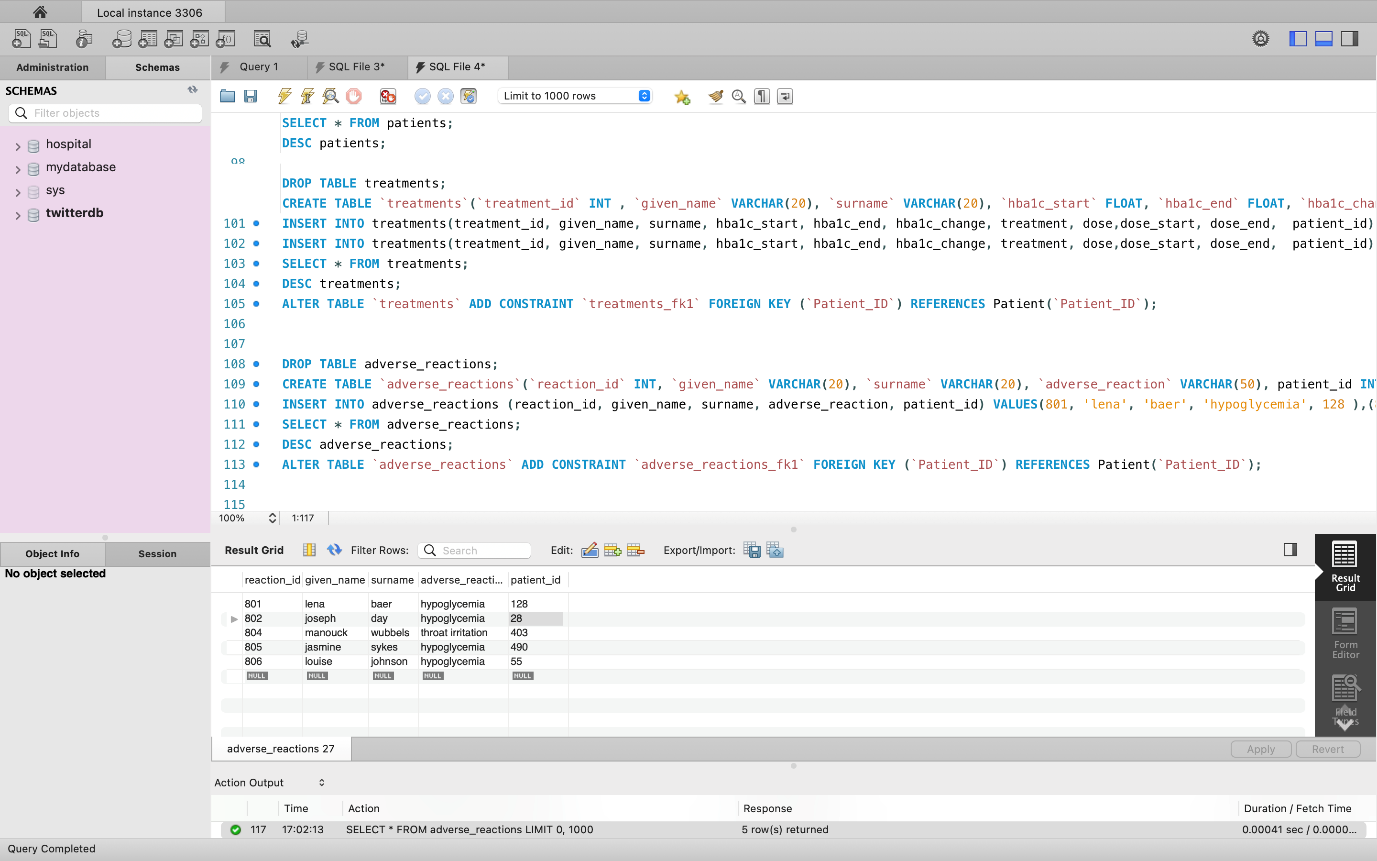
ALTER TABLE `treatments` ADD CONSTRAINT `treatments\_fk1` FOREIGN KEY (`Patient\_ID`) REFERENCES Patient(`Patient\_ID`);

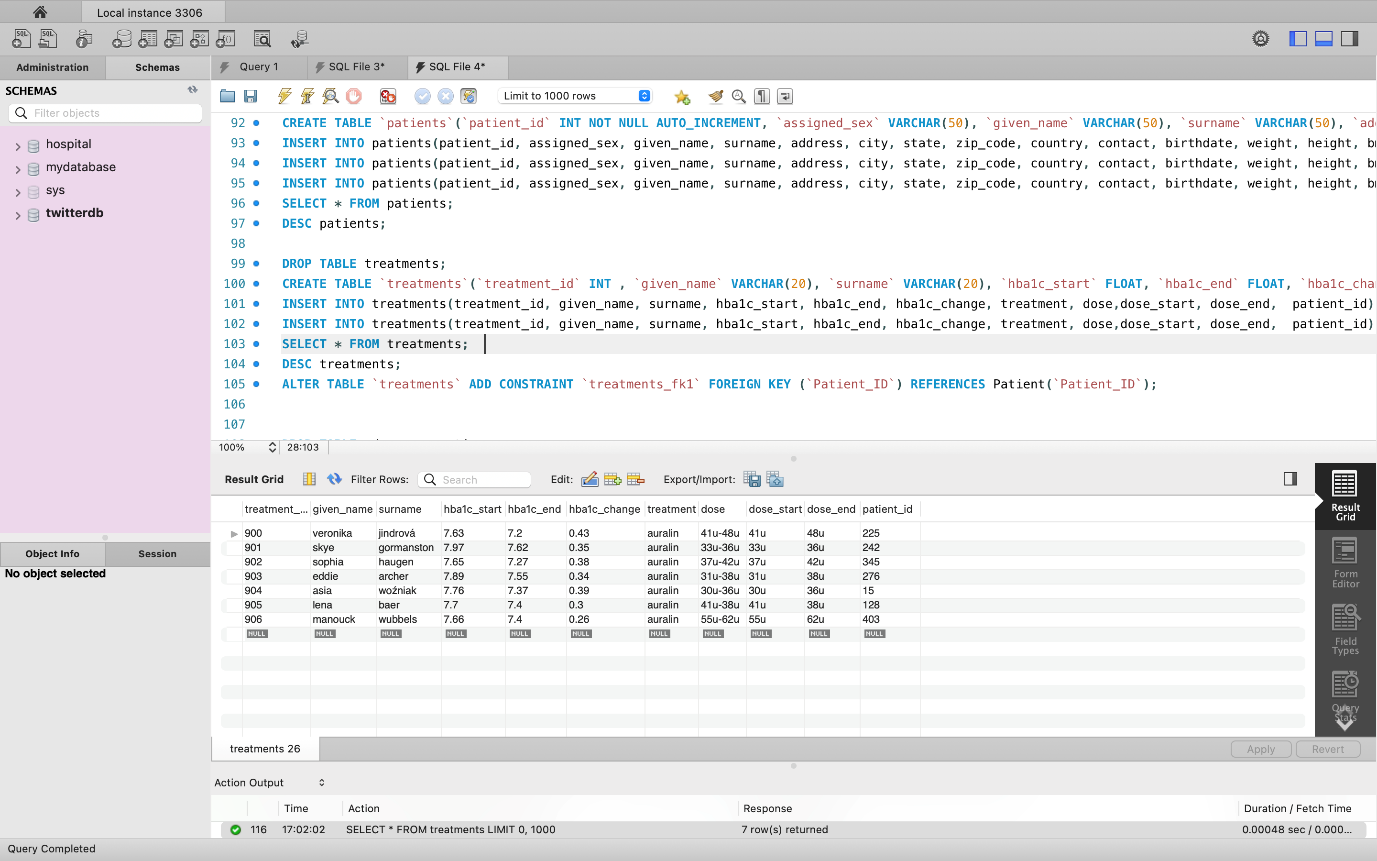
**Adverse Reactions**

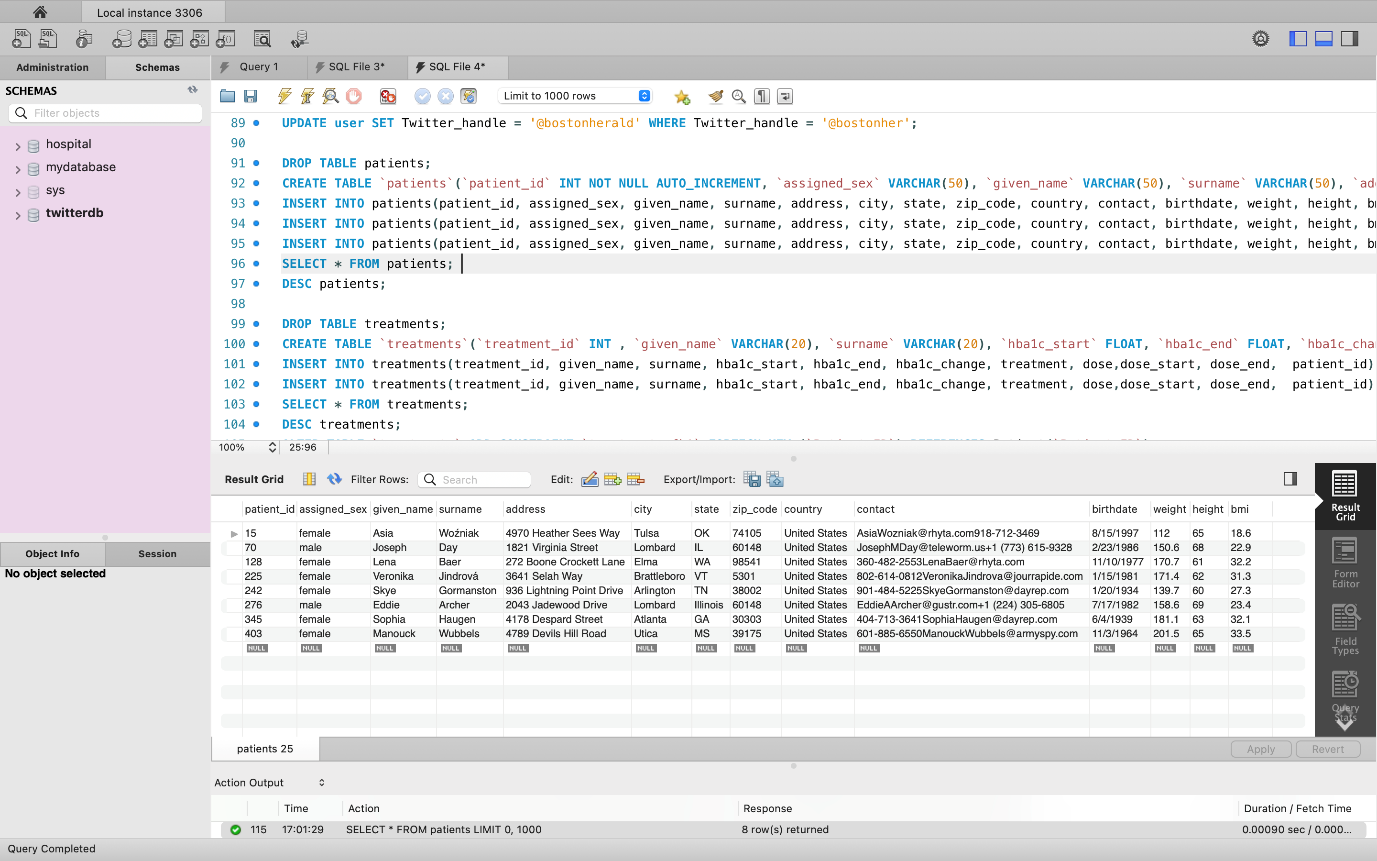
CREATE TABLE `adverse\_reactions`(`reaction\_id` INT, `given\_name` VARCHAR(20), `surname` VARCHAR(20), `adverse\_reaction` VARCHAR(50), patient\_id INT, PRIMARY KEY (reaction\_id) );

INSERT INTO adverse\_reactions (reaction\_id, given\_name, surname, adverse\_reaction, patient\_id) VALUES(801, 'lena', 'baer', 'hypoglycemia', 128 ),(802, 'joseph', 'day', 'hypoglycemia', 70), (804, 'manouck', 'wubbels', 'throat irritation', 403), (805, 'jasmine', 'sykes', 'hypoglycemia', 490),(806, 'louise', 'johnson', 'hypoglycemia', 55 );

**DATABASE SCHEMA**

****

****

****