FANOURIOS LAVDAS

ENGINEERING DATABASE SYSTEMS

IE 3425

MIDTERM PROJECT

NOVEMBER 5, 2014

***DB Backend***

1. **Business Plan**

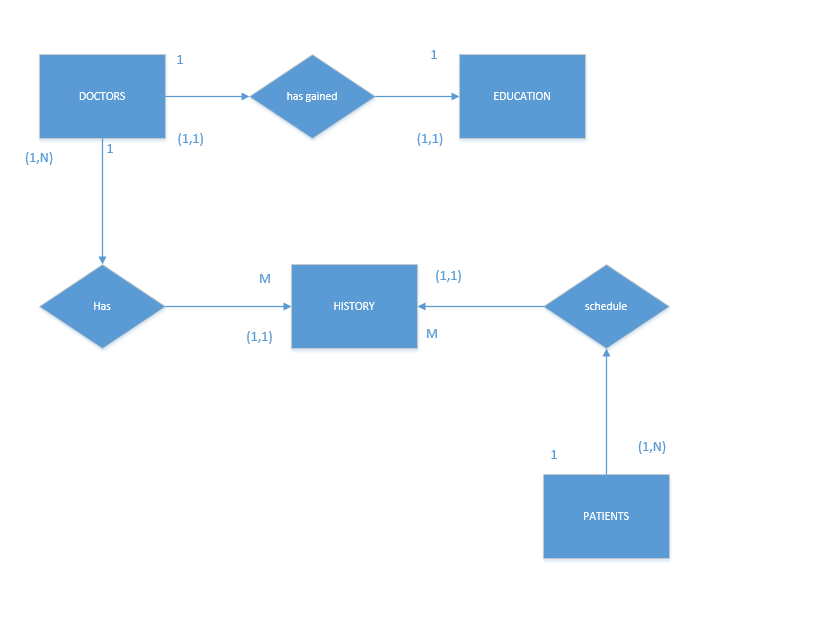
The company is a Health Clinic company in Boston, Massachusetts. A database is essential in order for the clinic to run effectively. The database keeps track of the history of appointments that patients make with doctors, as well as information for the patients, the doctors and their education. This database was created in order to keep track of all the appointments made between November 02, 2014 and November 06, 2014. It was determined that each appointment would be uniquely identified by the appointment number which was set at the FK of the appointment table (the table is labeled as “HISTORY” in the database). In addition, it was decided that the health record number of each patient, the ID of each doctor, the data and time of appointment as well as the medical room that the examination took place would also be included in the appointment history table. The clinic has 4 medical rooms (A,B,C and D). Each Health record number uniquely identifies a patient as well as the Doctor ID uniquely identifies each Doctor. It is important to notice that both a health record number and a Doctor’s ID can appear more than one times, as it is possible that a patient had more than one appointments scheduled between these dates and a Doctor might have more than one appointments during those days. Furthermore, three essential questions were included in the table; weather a blood test was performed and a prescription was given during the appointment and weather a further appointment was necessary. Finally, the Health Insurance Company of each patient was also essential to appear on the history of appointments table.

The Doctor’s information is essential to be known as well. A table labeled “DOCTORS” was created in this database, which shows all the necessary information for each Doctor. The Doctor ID uniquely identifies each Doctor. Essential information are also the First and Last Name, the address, city, state, zip code as well as email and phone. Finally, although there is a small clinic, it consists of doctors of a variety of areas of specializations: cardiologist, dermatologist, allergist, epidemiologist, endochrinologist, immunologist, orthopedist, pathologist, neurologist and oncologist. The area of specialization of each doctor is also included in the doctors’ information table.

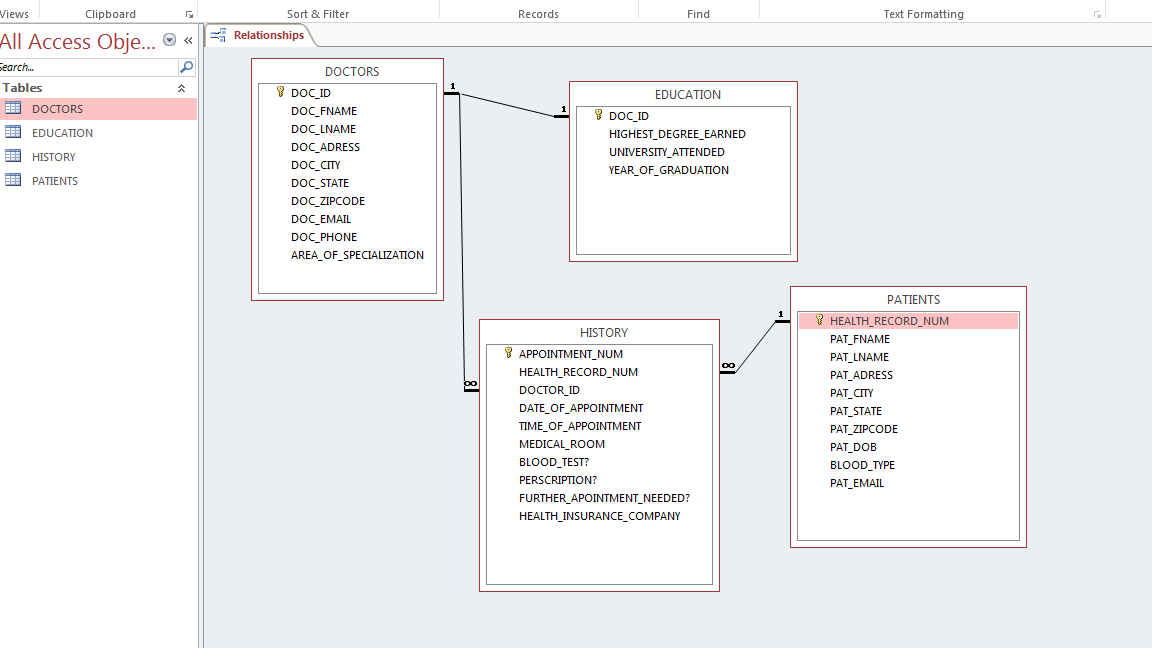
The patients’ information is also essential to show. As previously mentioned the Health Record Number uniquely identifies each patient. The patients first name, last name, address, city, state and zip code are information that are shown in the table as well as the patients date of birth, patient’s blood type and patients email. The patients shown in the tables are only the tables that had scheduled at least one appointment during the date period of November 02 to November 06.

How qualified a doctor is also a very important thing to know. That is why the Doctor’s education was also shown in the database. The Doctor’s ID, the highest degree they have earned, the university they attended and the year of graduation are shown in the doctors’ education table (labeled as “EDUCATION” in the database). The director of the clinic is very selective when hiring new doctors. He believes that his clinic should only have the most qualified doctors in the world and as he believes that Harvard University and Johns Hopkins University have the best medical schools in the world, therefore he has set a rule that new doctors would be hired only if they have graduated in one of these two universities.

1. **ERD**



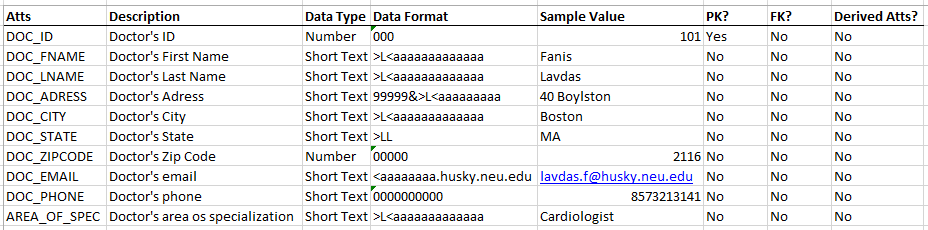
1. **Relationship Schema**



1. **Normalization**

No normalization needed. All tables are in 3NF. There is no Composite Primary Key in any of the tables, so partial dependencies. Also, the tables were created in such a way that no transitive dependencies would occur. Therefore, no normalization is needed.

1. **Data Dictionary for DOCTOR table**



1. **Tables**

**DOCTORS TABLE**

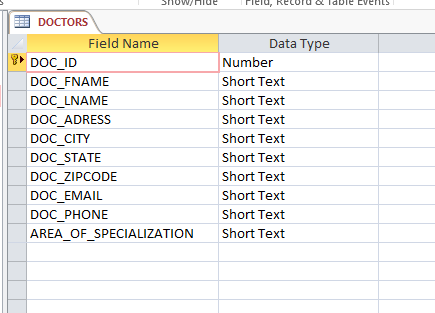


Figure 1: DOCTORS table design view

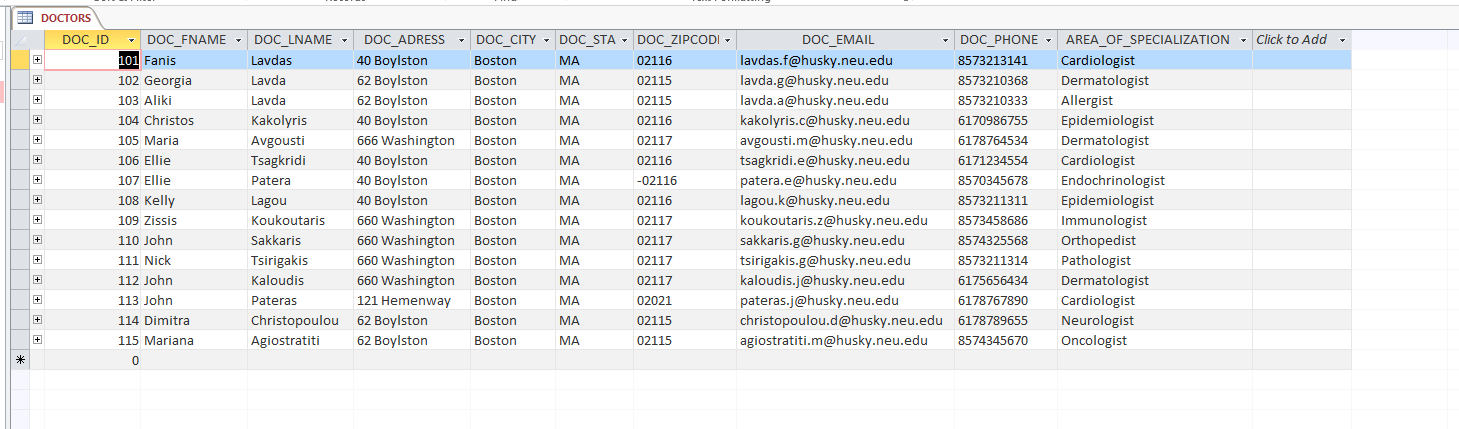


Figure 2: DOCTOR table Datasheet view

**PATIENTS TABLE**

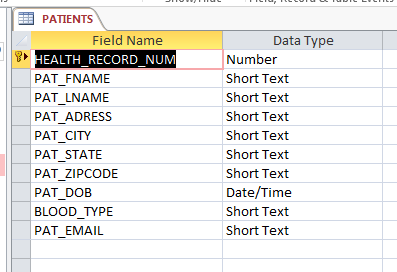


Figure 3: PATIENTS table design view

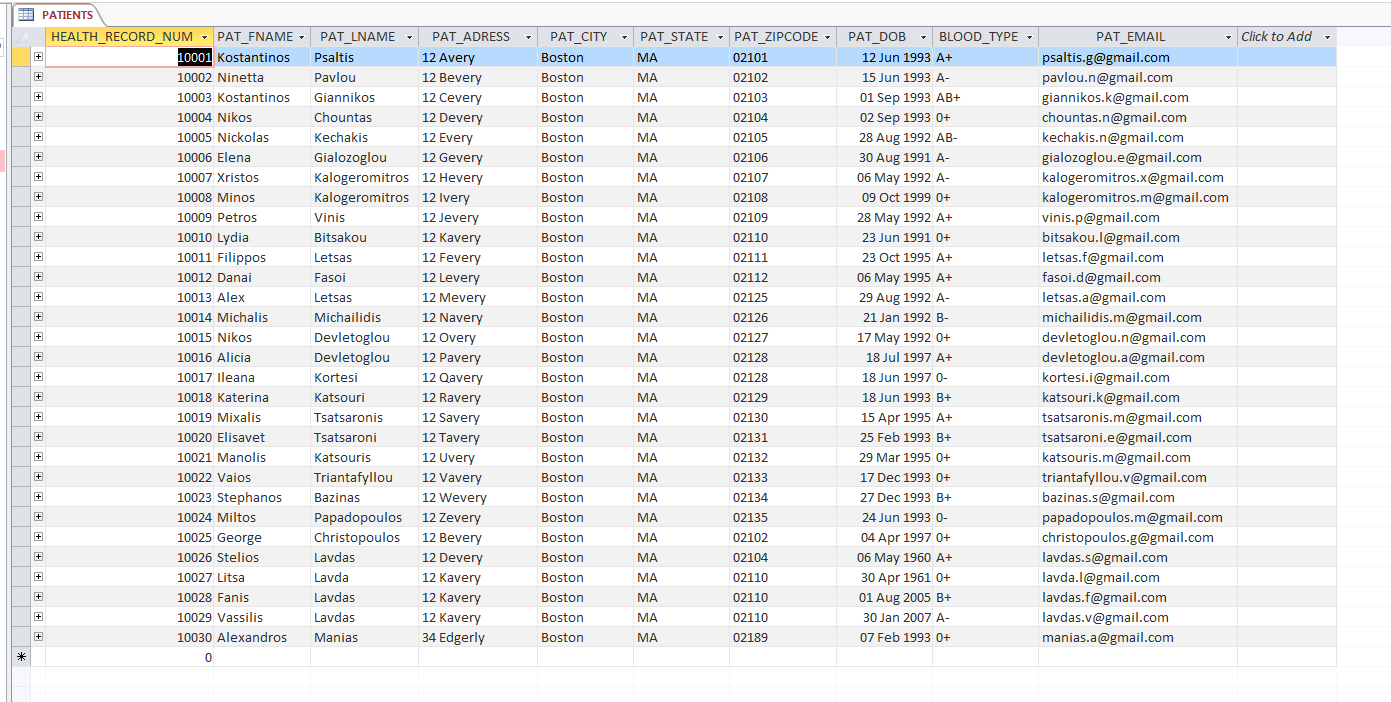


Figure 4: PATIENTS table datasheet view

**HISTORY TABLE**

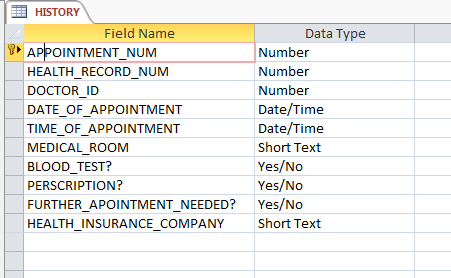


Figure 5: HISTORY table design view

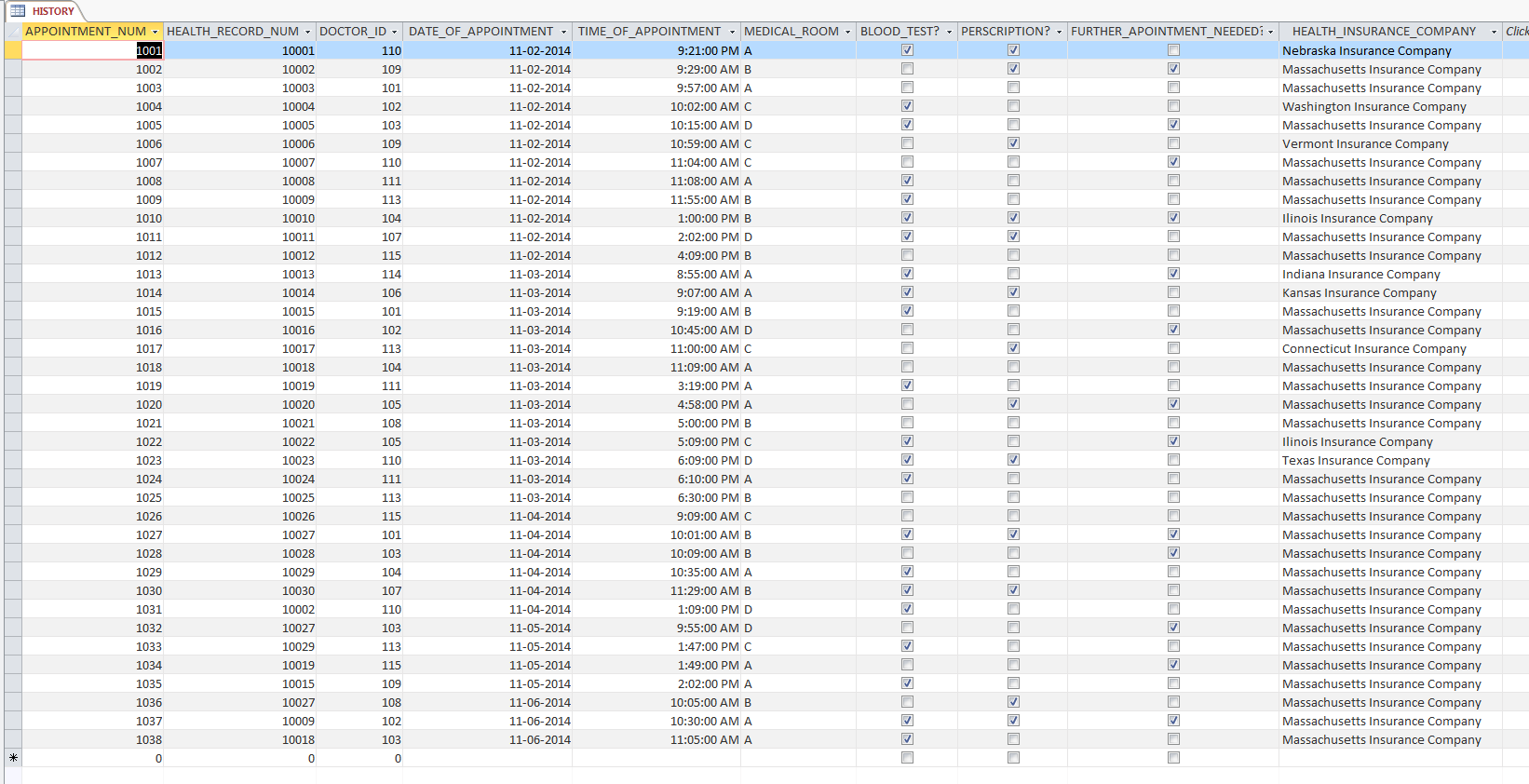


Figure 6: HISTORY table datasheet view

**EDUCATION TABLE**

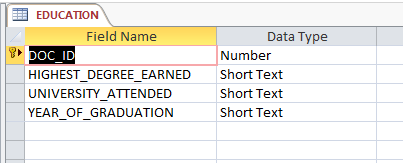


Figure 7: EDUCATION table design view

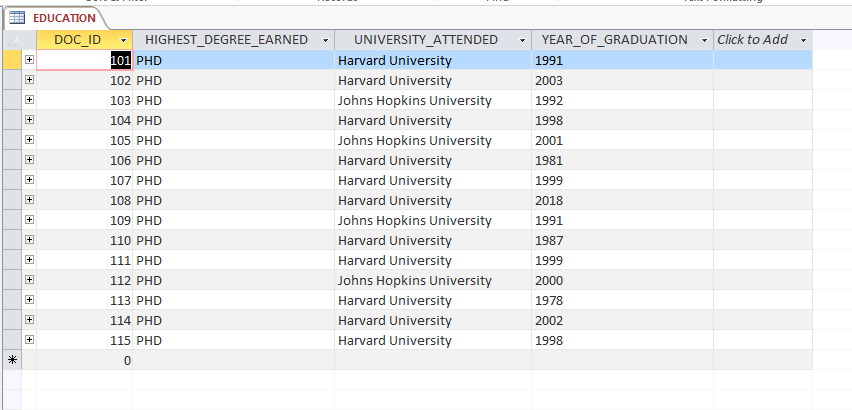


Figure 8: EDUCATION table datasheet view

1. **Queries**

**Sorting Query: sort Patients by ascending last name**

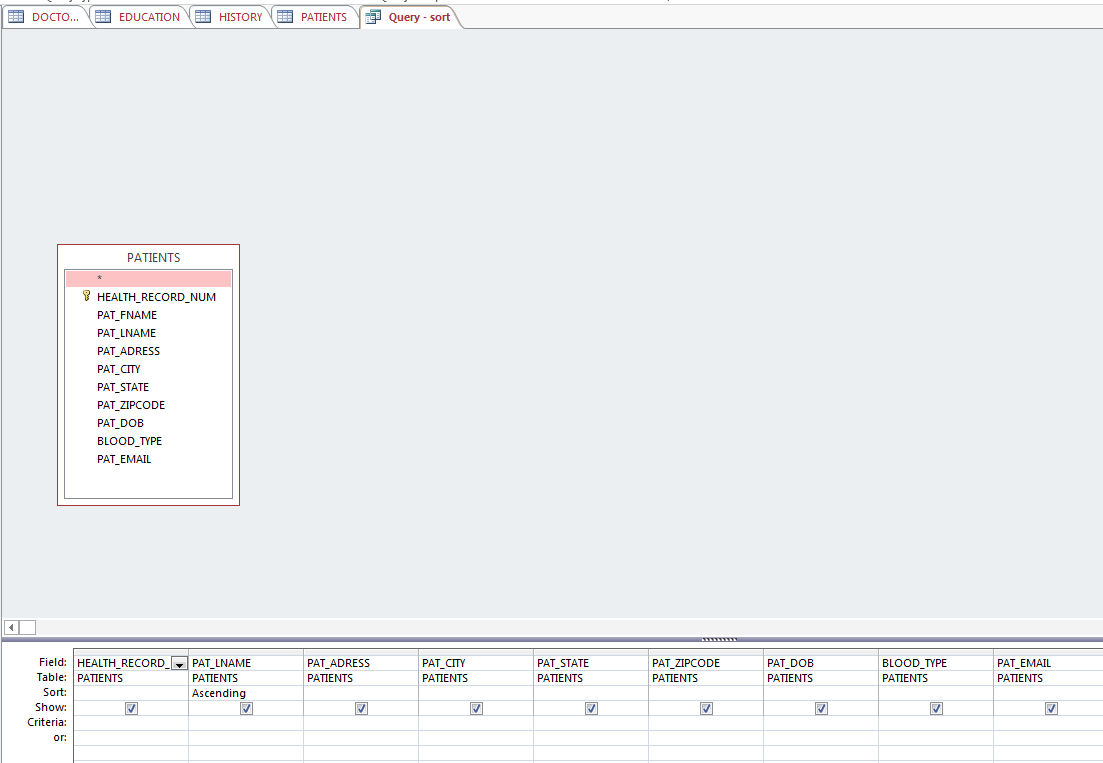


Figure 9: Sort query Design View

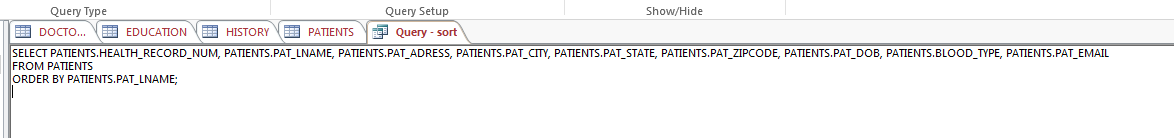


Figure 10: Sort Query SQL View

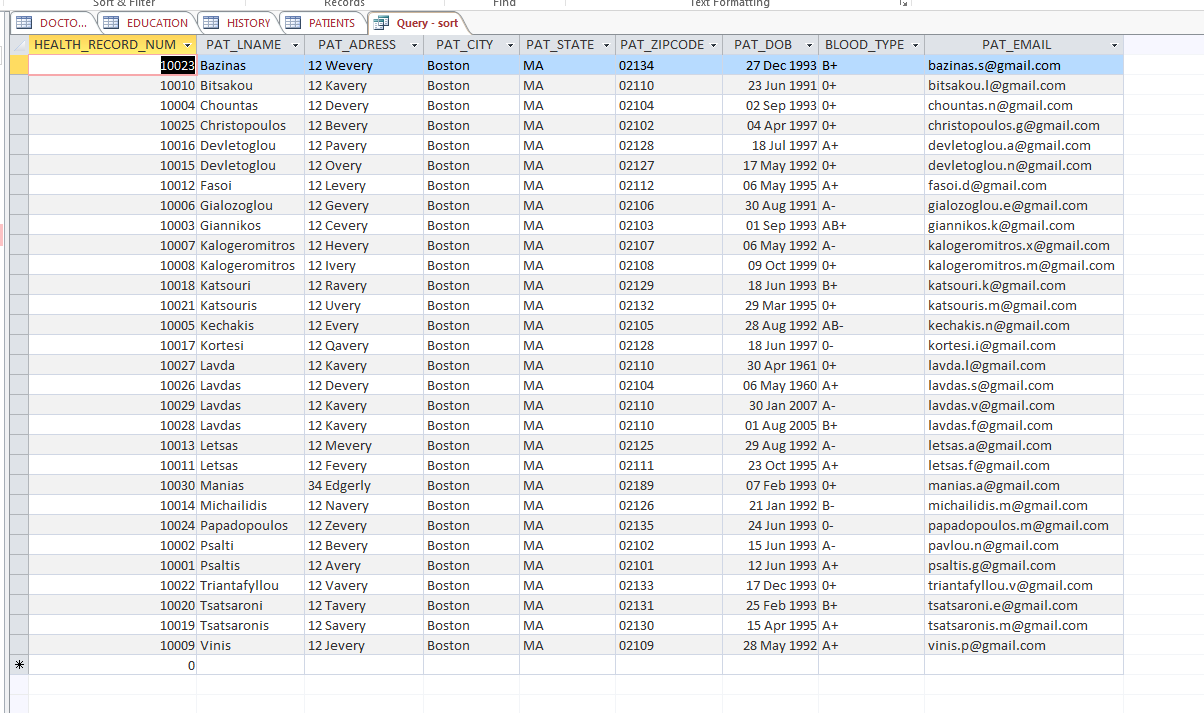


Figure 11: Sort Query Datasheet View

**Concatenation Query: show health record number and formatted date:**

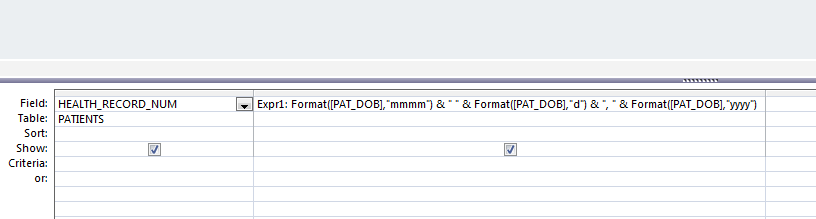


Figure 12: Concatenation Query Design View

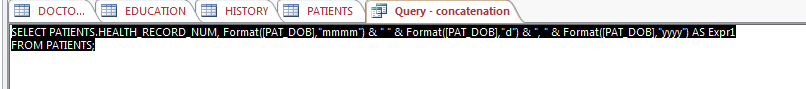


Figure 13: Concatenation Query SQL View

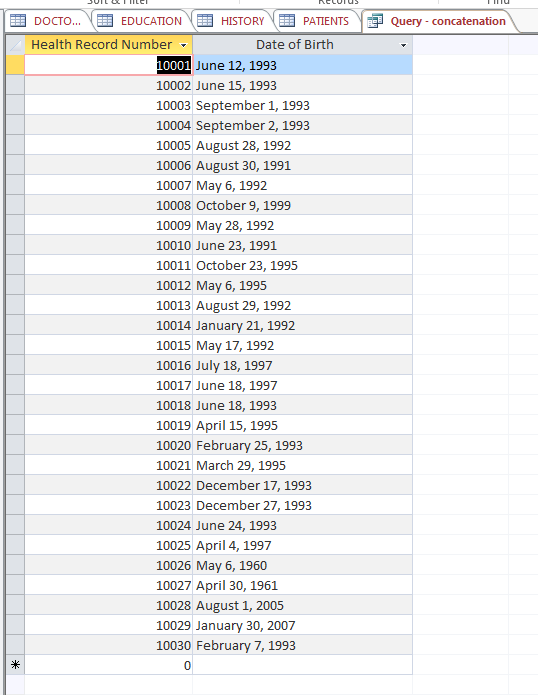


Figure 14: Concatenation Query Datasheet View

**Left Outer Join query**

**Doctors who do not have any scheduled appointments**

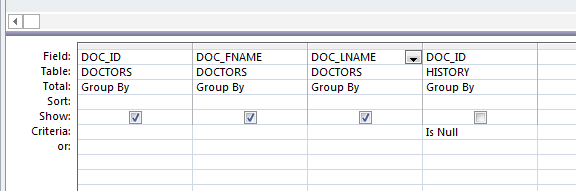


Figure 15: Design View

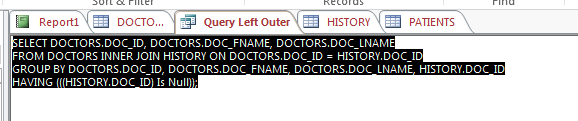


Figure 16: SQL View

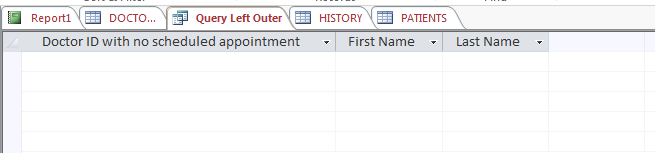


Figure 17: Datasheet View

As it can be noticed, there are no registered doctors with no scheduled appointments.

**Right Outer Join query**

**Appointments with Doctors who are not registered as Doctors**

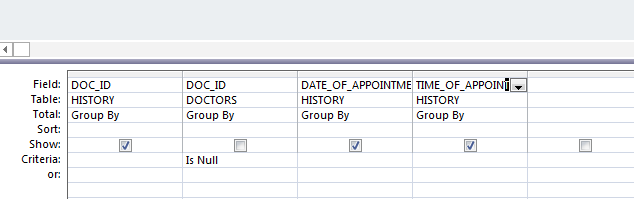


Figure 18: Design View

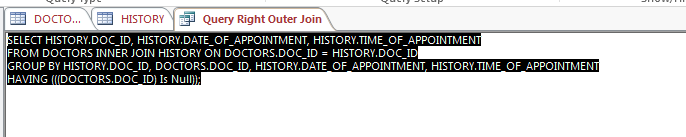


Figure 19: SQL View

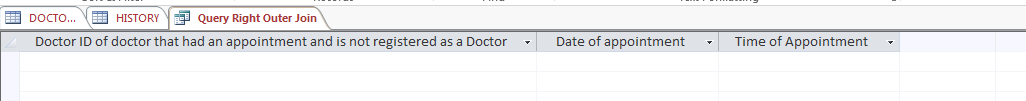


Figure 20: Datasheet View

As it can be noticed, there is no Doctor who had an appointment in the clinic, who is not a registered Doctor of the clinic.

**Update Query: Ninetta Pavlou married Constantinos Psaltis and changed her last name from Pavlou to Psalti:**

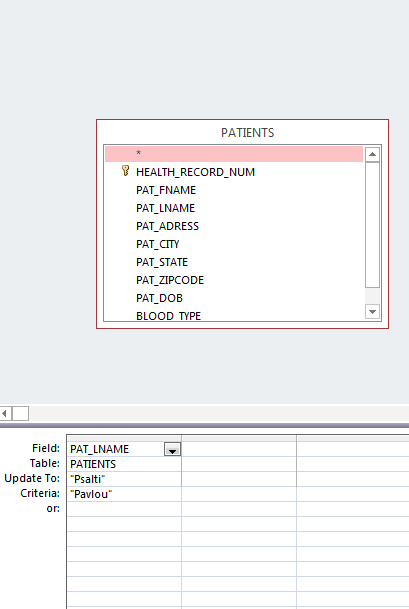


Figure 21: Update Query Design View

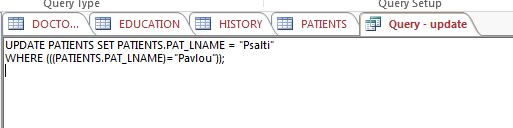


Figure 22: Update Query SQL View

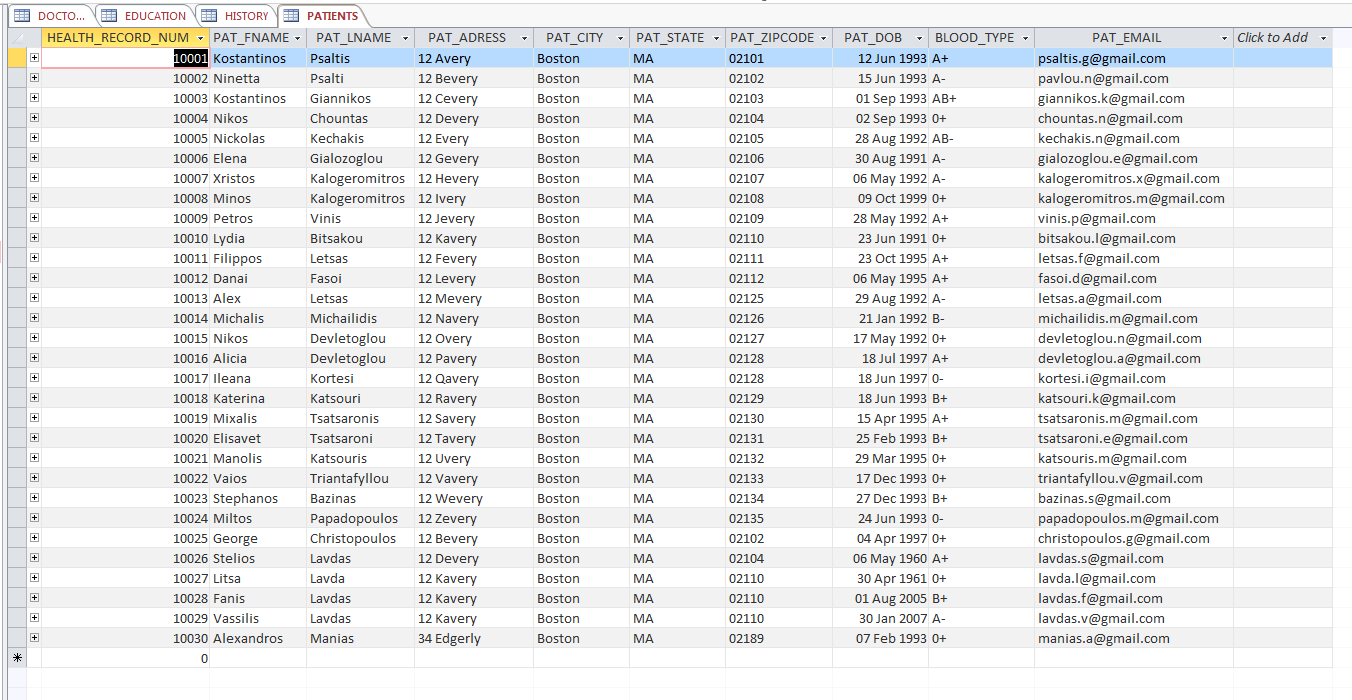


Figure 23: Updated PATIENTS table with updated last name "Psalti'

**Append Query: Insert all information for a new patient in the PATIENT table:**

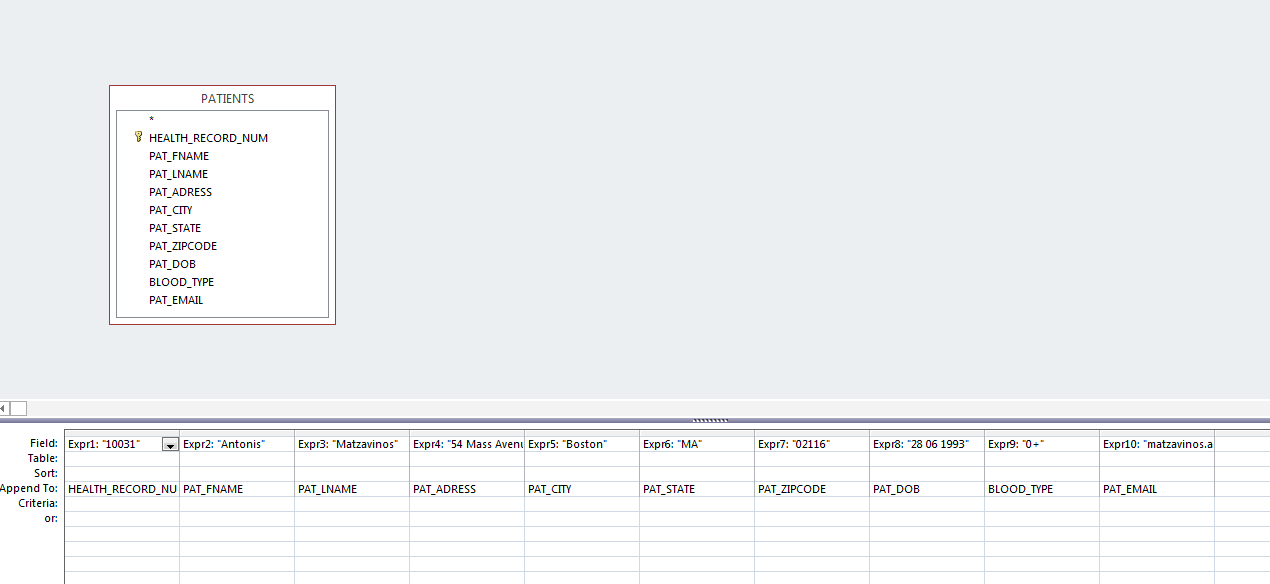


Figure 24: Append Query Design View

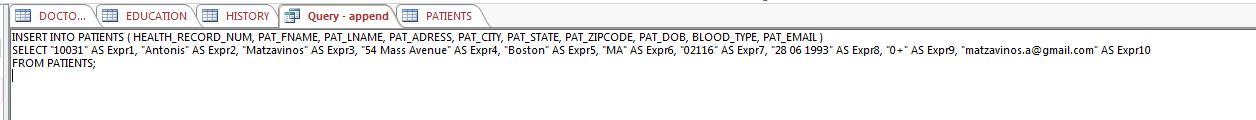


Figure 25: Append Query SQL View

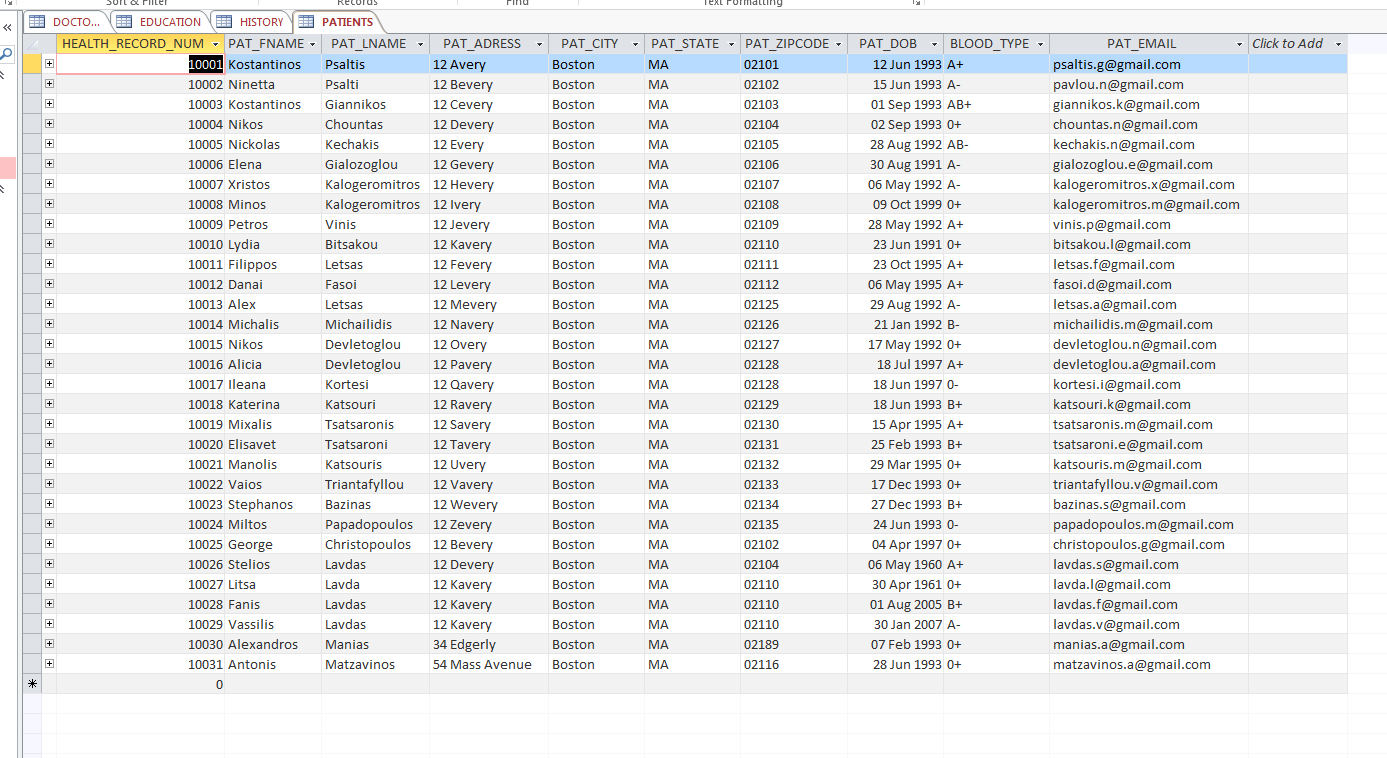


Figure 26: PATIENTS table with inserted new patient

**Delete Query: delete the last appointment in the HISTORY table.**

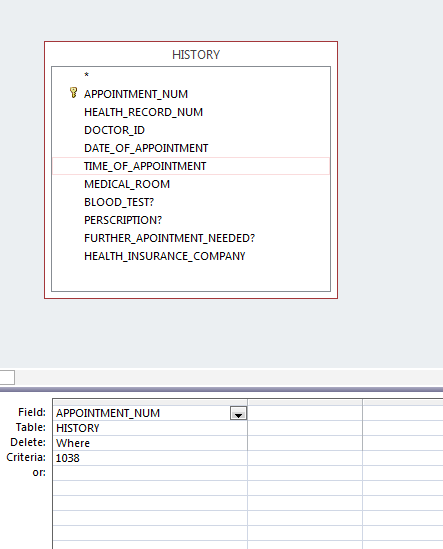


Figure 27: Delete Query Design View

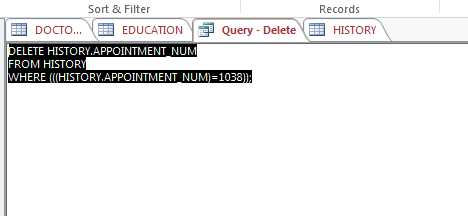


Figure 28: Delete Query SQL View

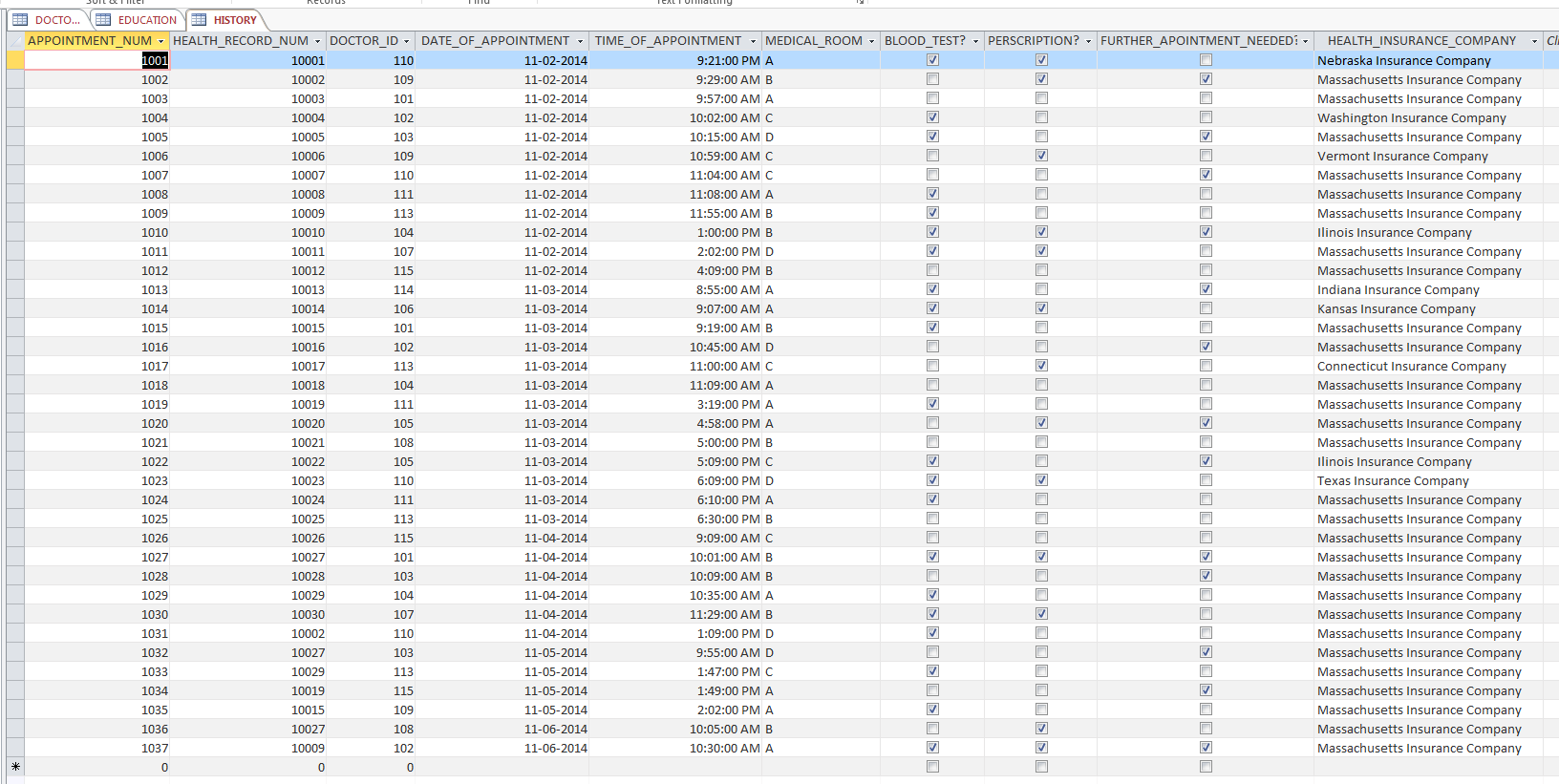


Figure 29: Updated HISTORY table after deletion of last row

1. **Input masks, validation rules and validation text**

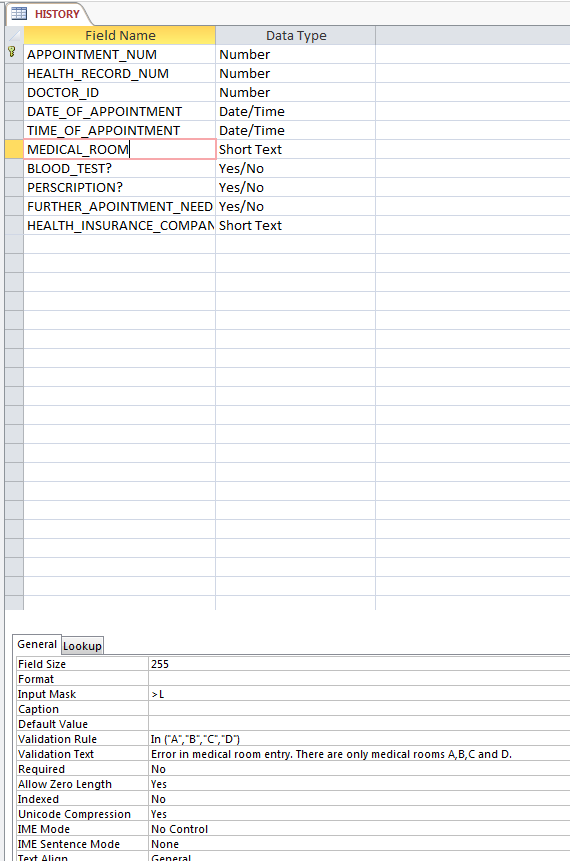


Figure 30: Input mask, validation rule and validation text for MEDICAL\_ROOM

**No default value was needed**

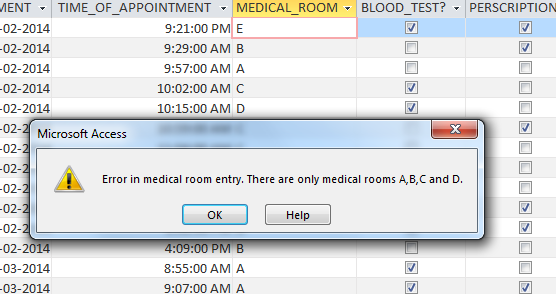


Figure 31: Validation text shown when entry violates Validation rule

***Front End Access GUI:***

1. **Form with three different tabs. Each tab adds a new row in the three tables (3 forms to add data to the DB).**

**Add Doctor Form**

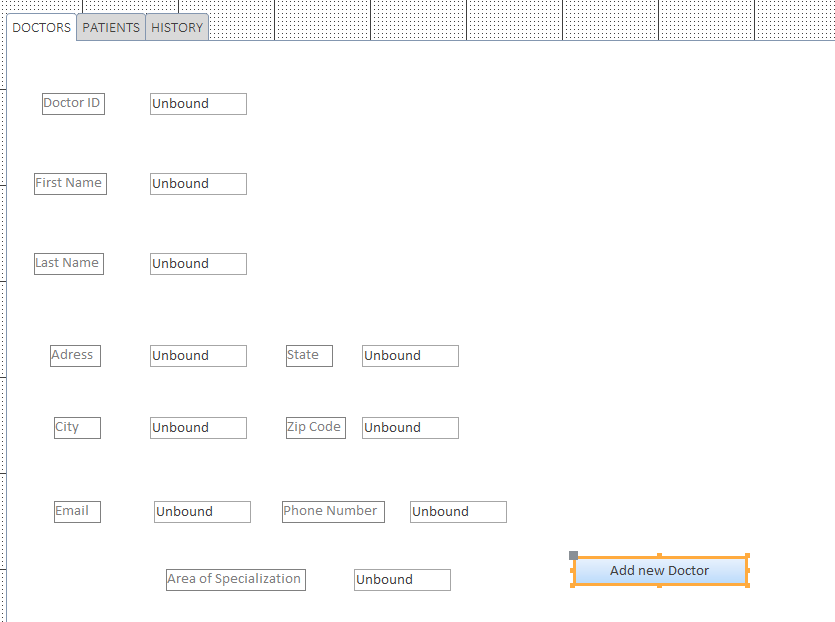


Figure 32: Add new doctor form Design View

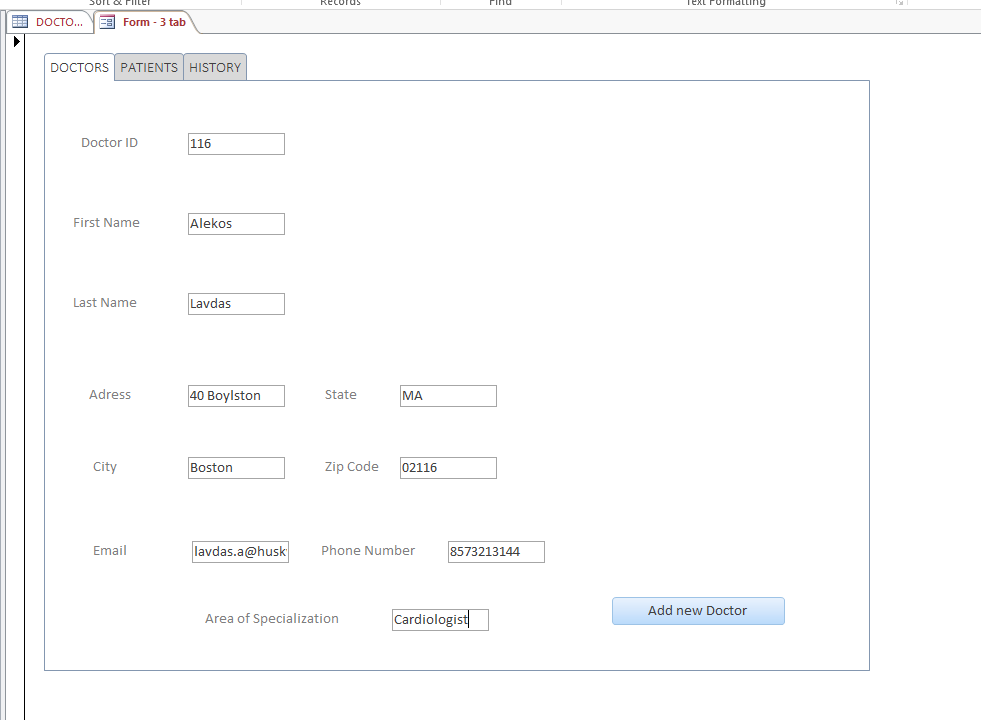


Figure 33: Add new doctor Form View

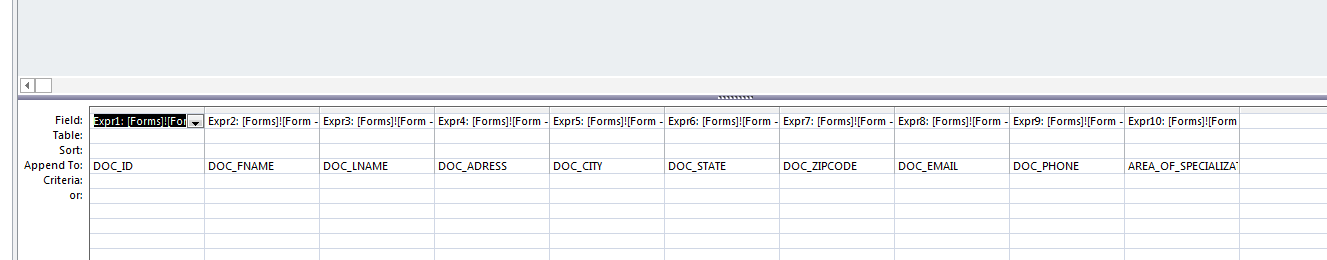


Figure 34: ADD DOCTOR query Design View

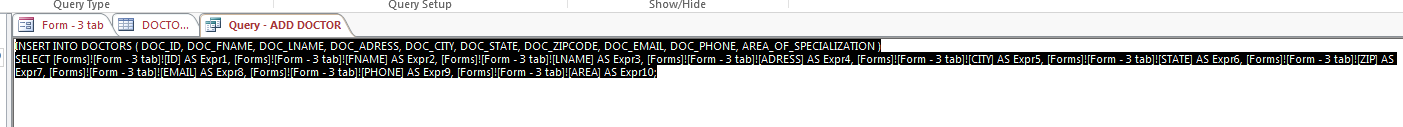


Figure 35: ADD DOCTOR query SQL View

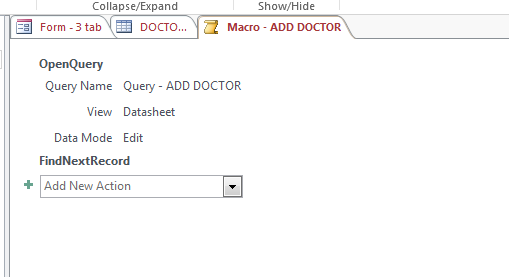


Figure 36: ADD DOCTOR macro

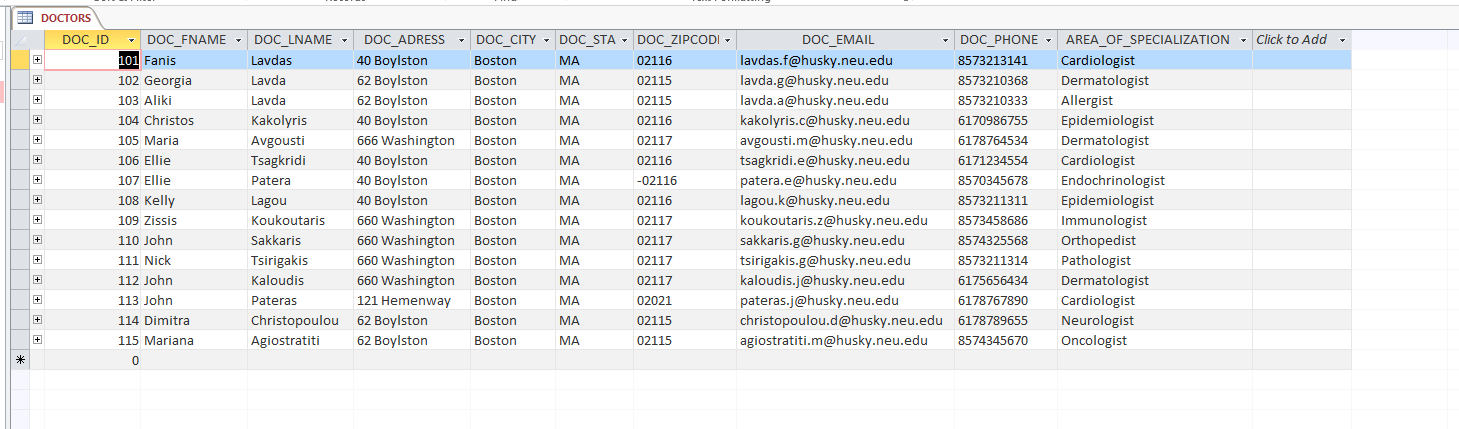


Figure 37: DOCTORS table before

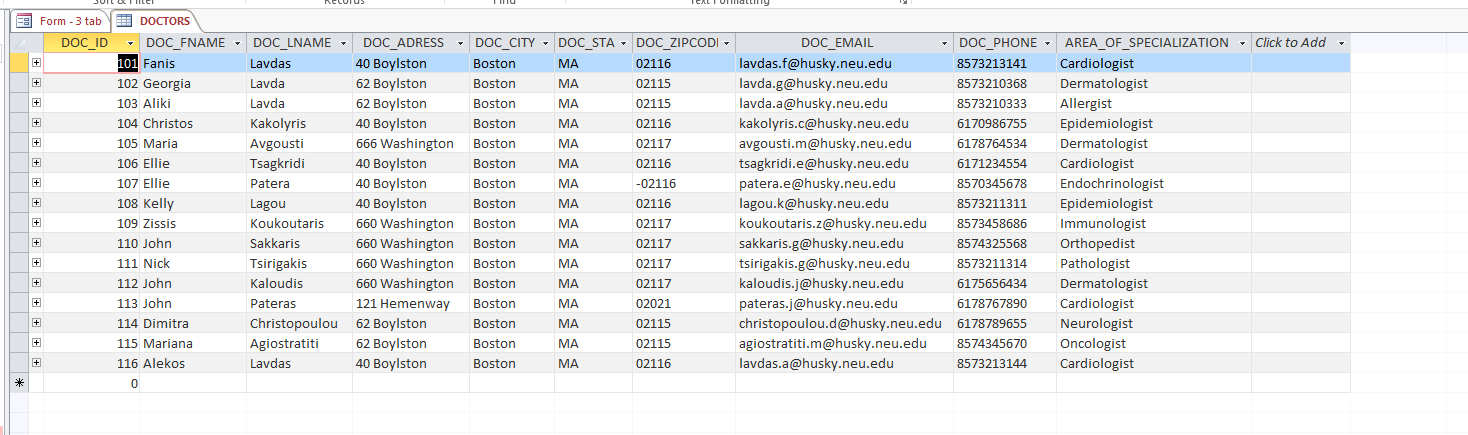


Figure 38: DOCTORS table after addition of new doctor

**Add Patient Form**

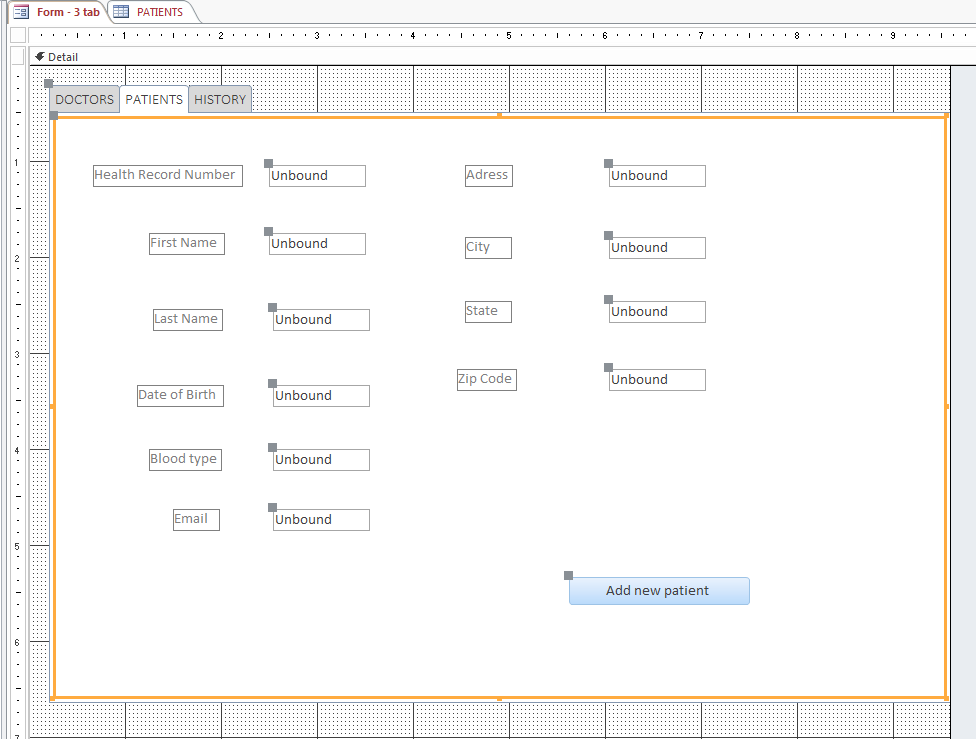


Figure 39: Add new patient Design View

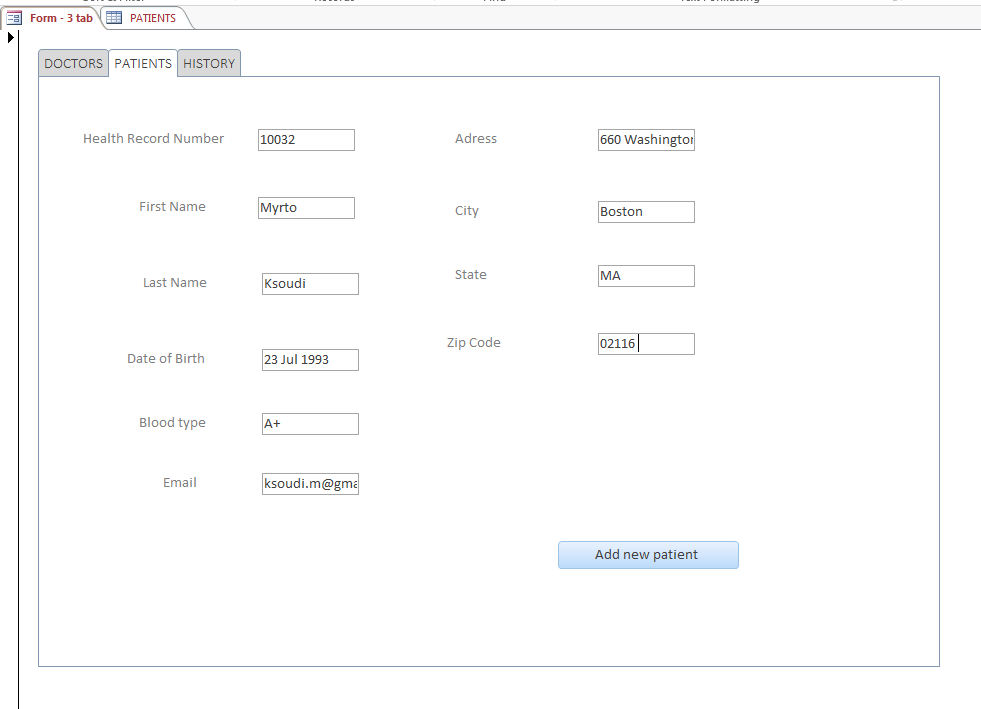


Figure 40: Add new patient Form View

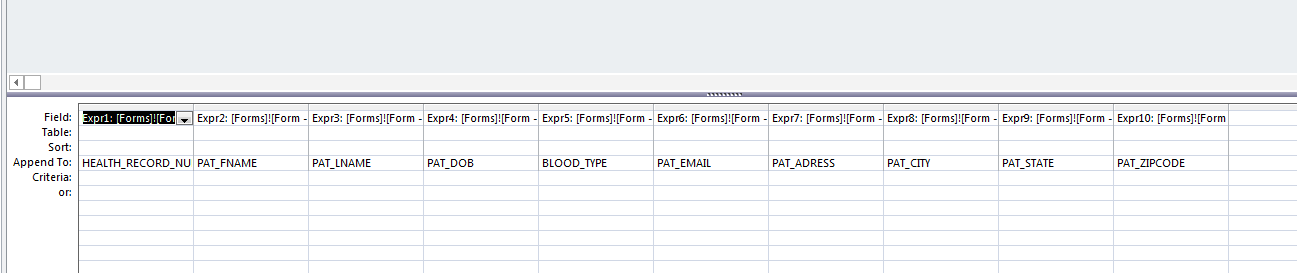


Figure 41: ADD PATIENT query Design View

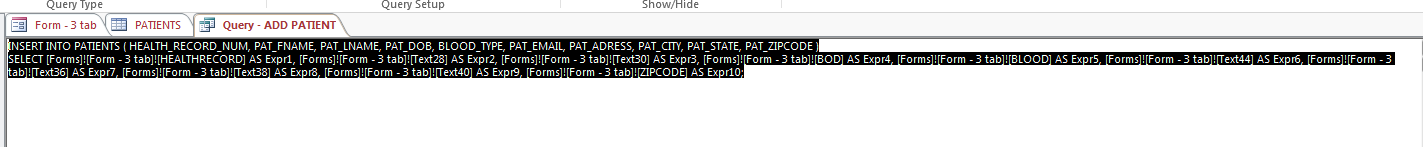


Figure 42: ADD PATIENT query SQL View

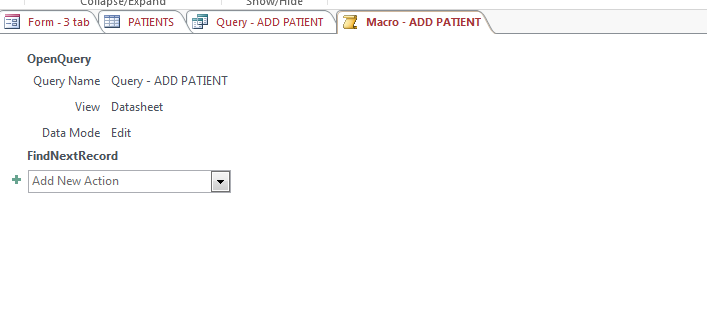


Figure 43: ADD PATIENT macro

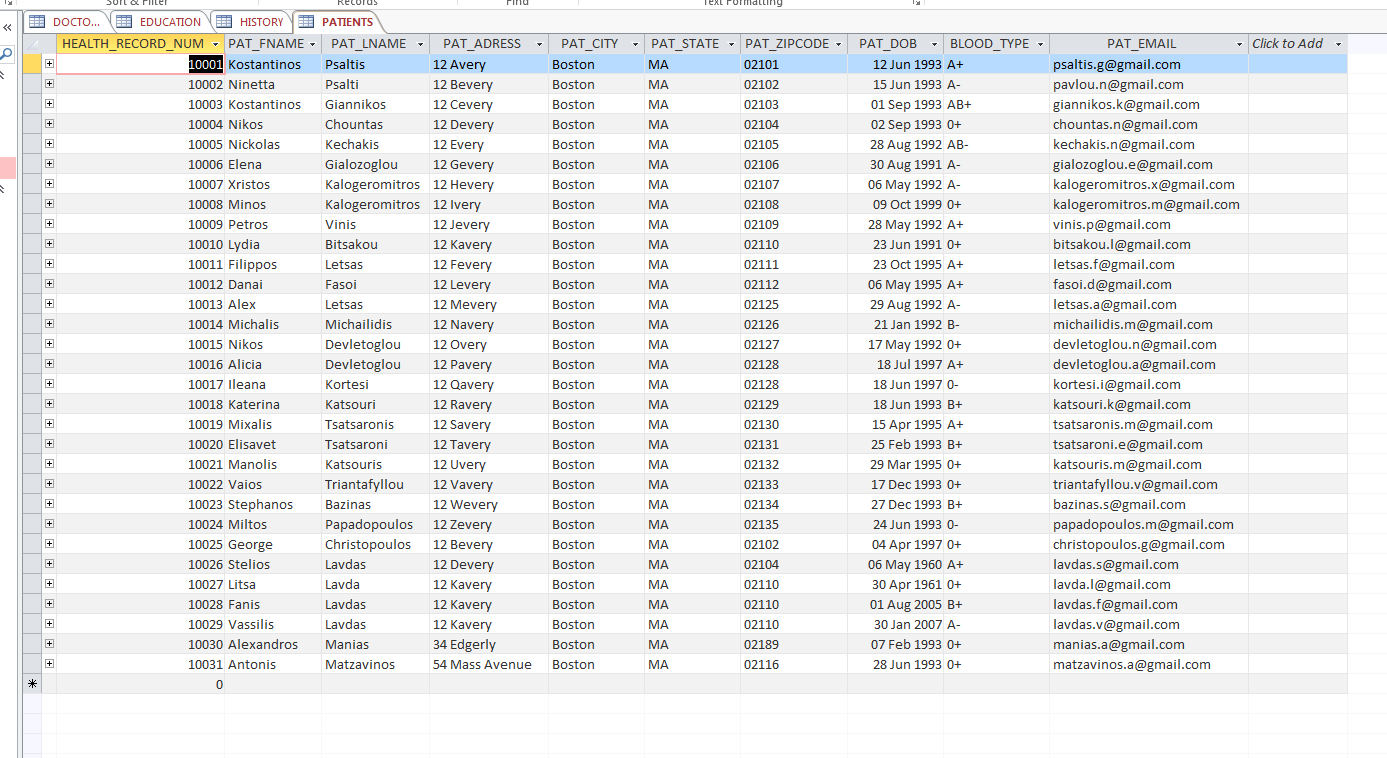


Figure 44: PATIENTS table before

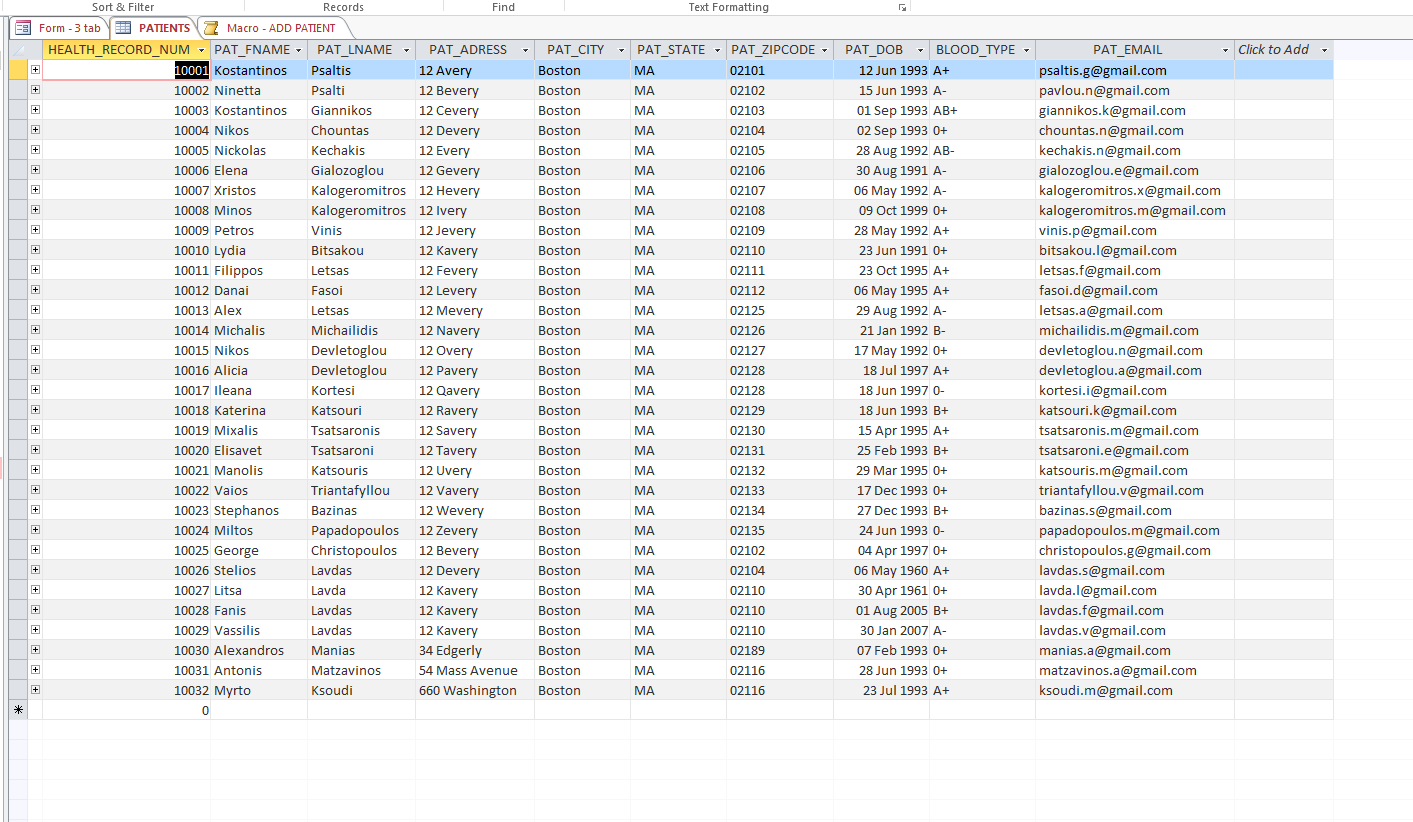


Figure 45: PATIENTS table after addition of new patient

**Add Education Information for new Doctor**

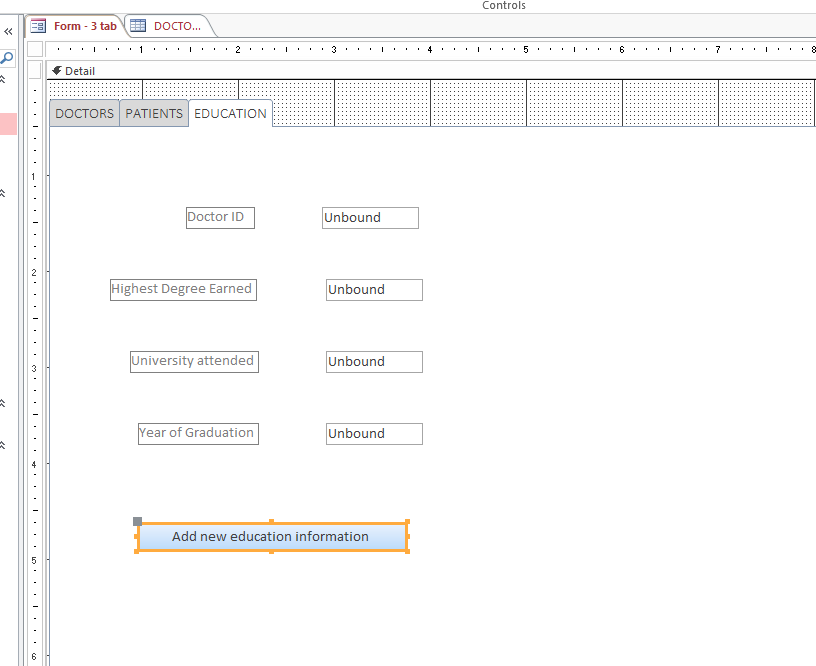


Figure 46: Add new education information for new doctor Design View

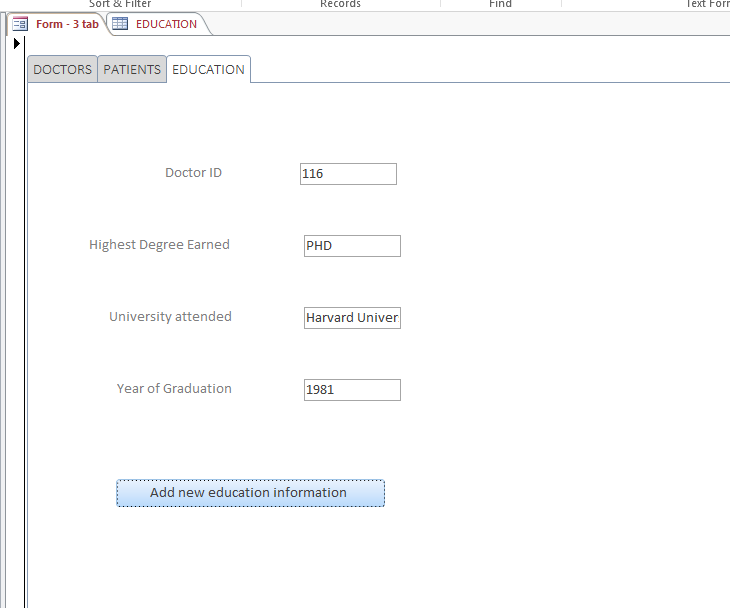


Figure 47: Add new education information for new doctor Form View

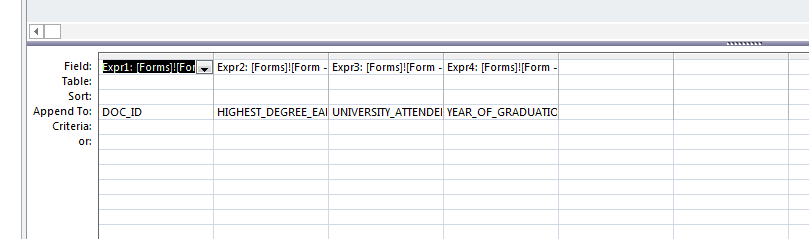


Figure 48: ADD EDUCATION query Design View

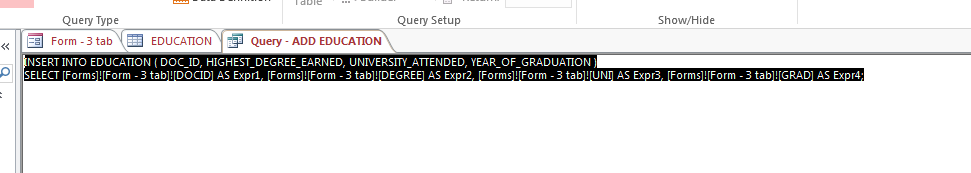


Figure 49: ADD EDUCATION query SQL View

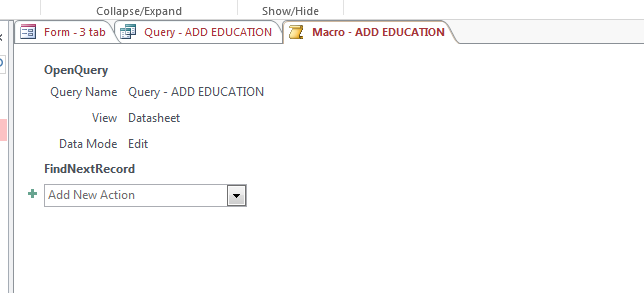


Figure 50: ADD EDUCATION Macro

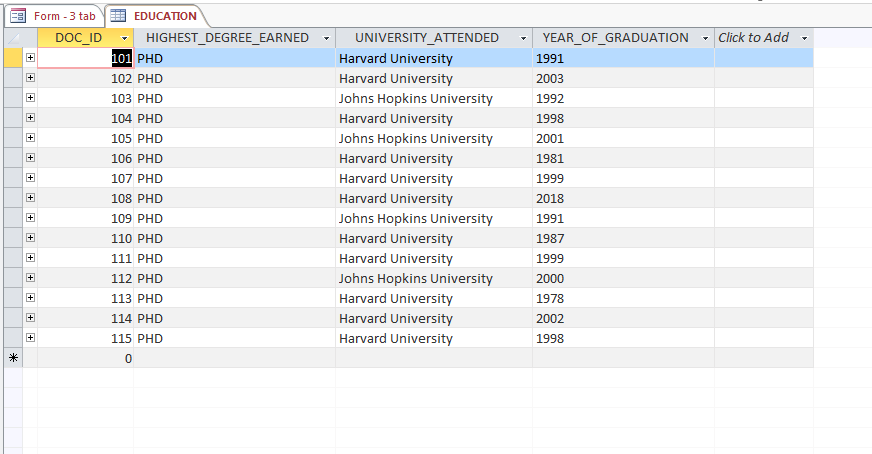


Figure 51: EDUCATION table before

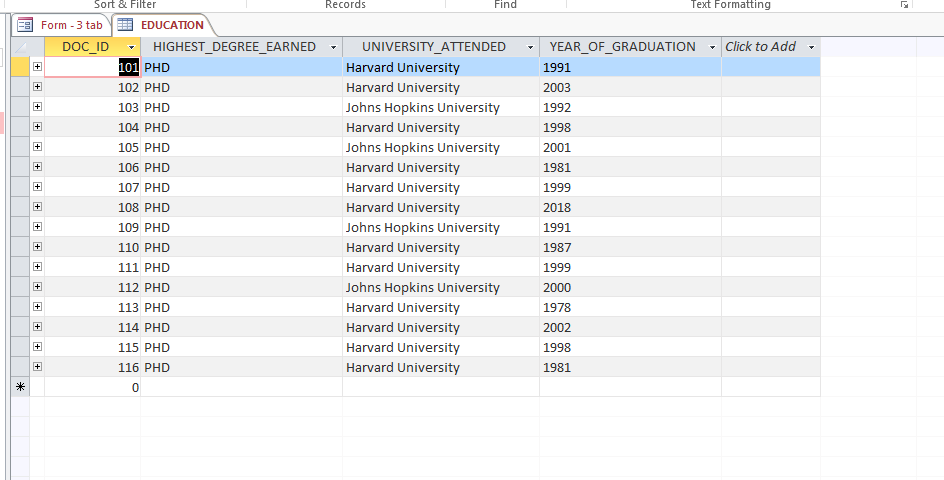


Figure 52: EDUCATION table after education information for new doctor is added

1. **All Access controls were added: text boxes and buttons were added in the above 3 forms to add data. Forms using combo box and list box are shown below**

**Find Doctor Information using their Email Address COMBO BOX**

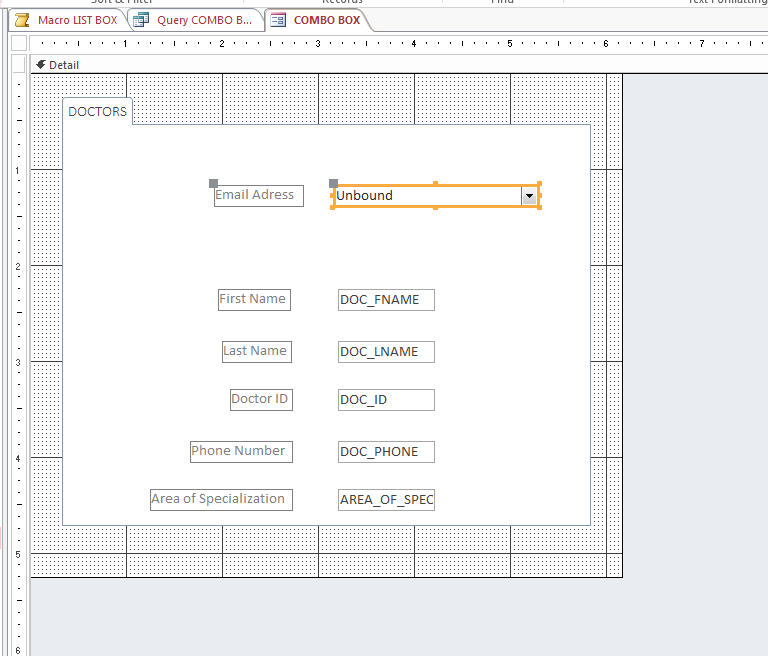


Figure 53: Combo box form Design View



Figure 54: COMBO BOX macro

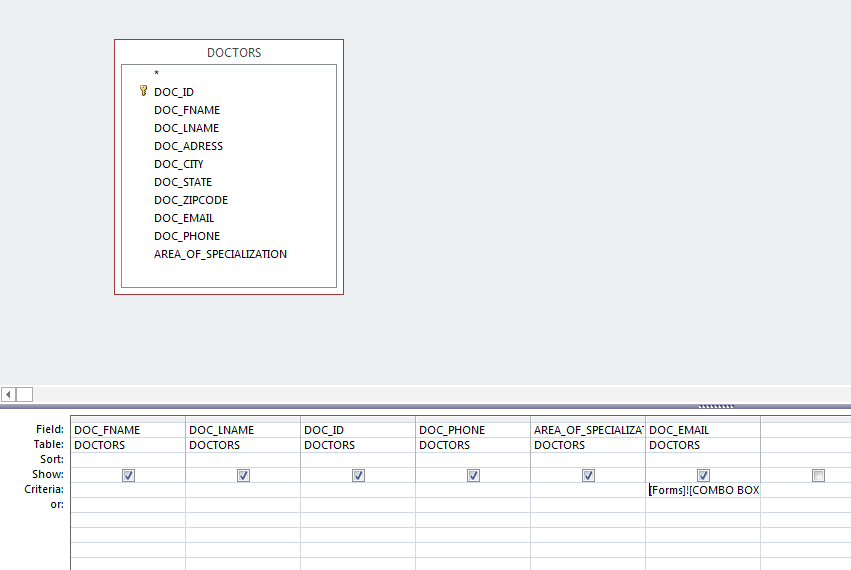


Figure 55: COMBO BOX query Design View

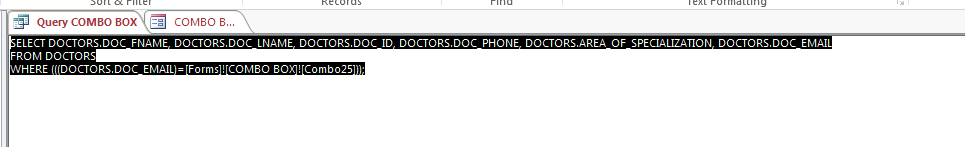


Figure 56: COMBO BOX query SQL View

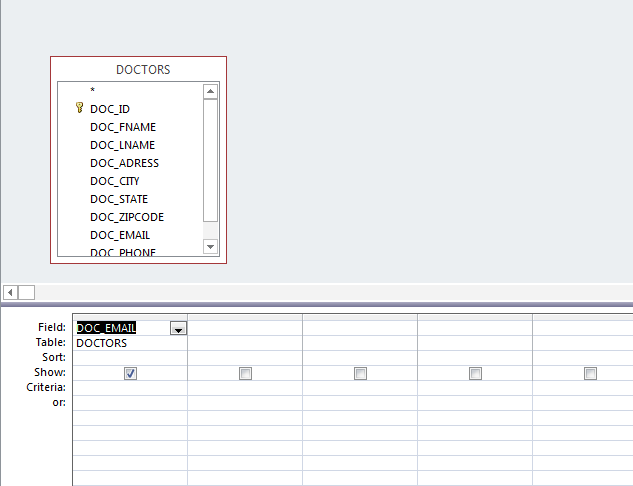


Figure 57 COMBO BOX\* query that is attached to the combo box Design View

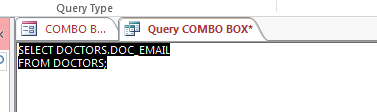


Figure 58: COMBO BOX\* query that is attached to the combo box SQL View

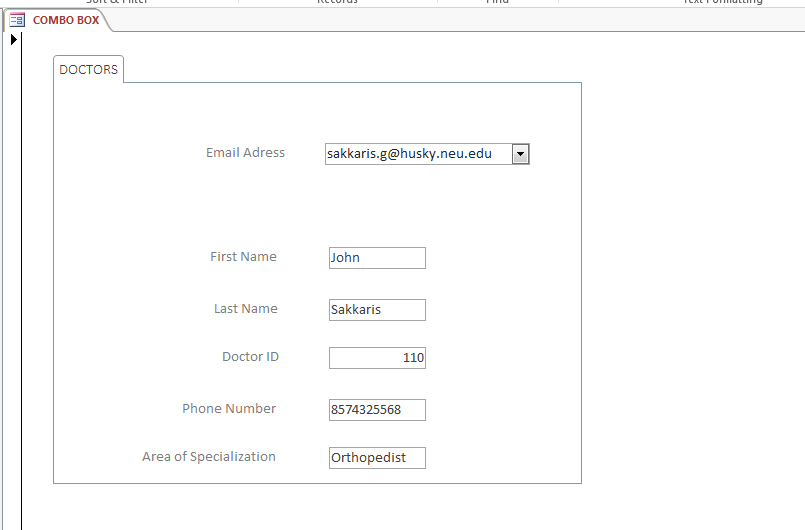


Figure 59: Form View

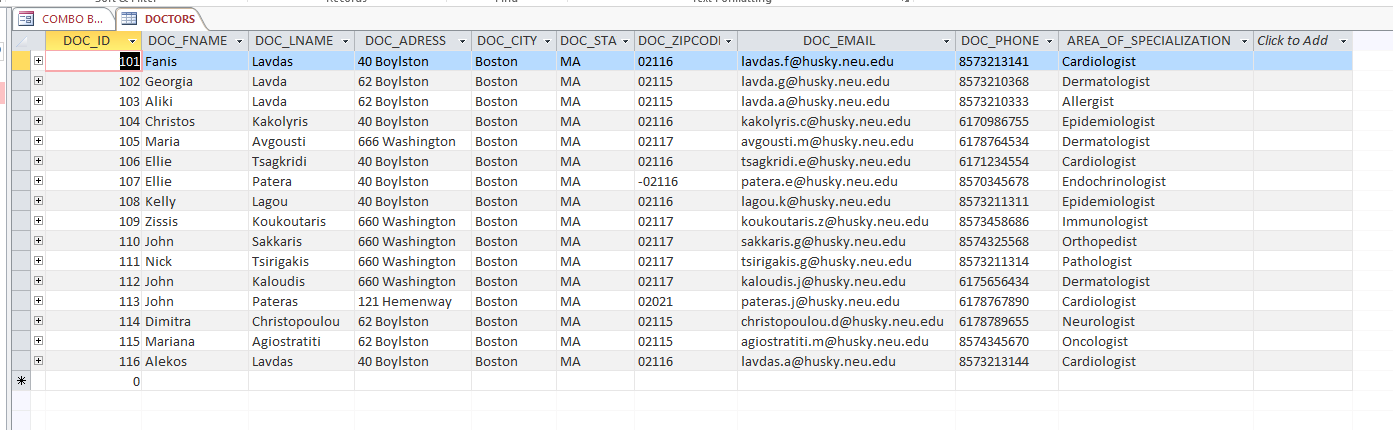


Figure 60: DOCTORS table

**Find Doctor Information using their Email Address LIST BOX**

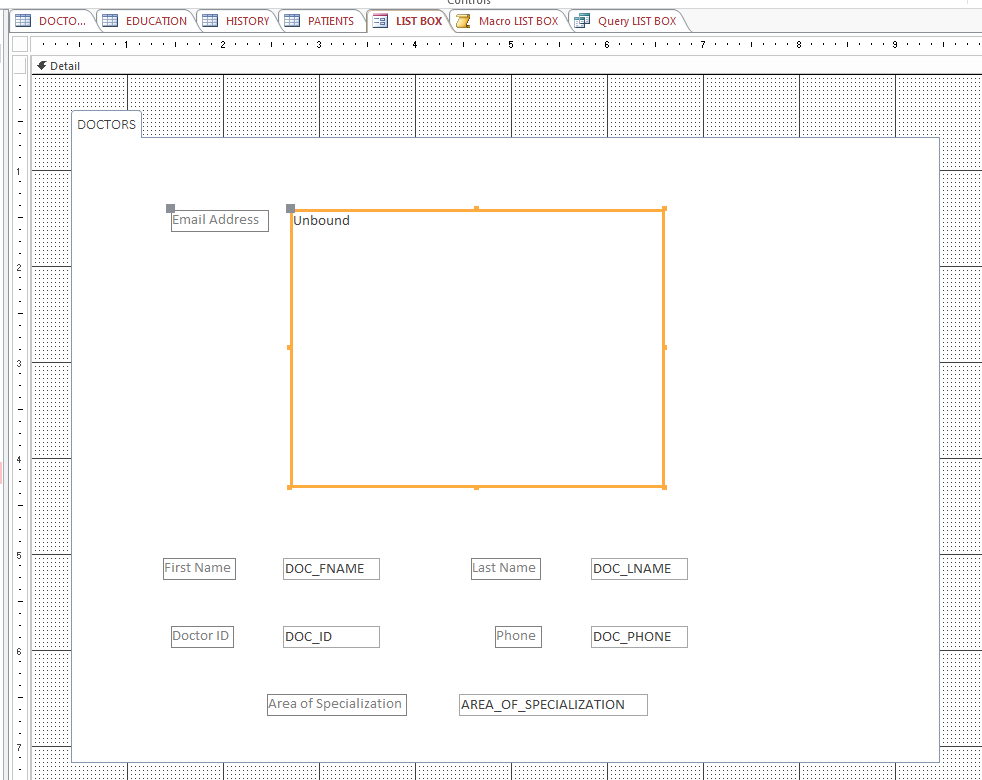


Figure 61: LIST BOX Form Design View

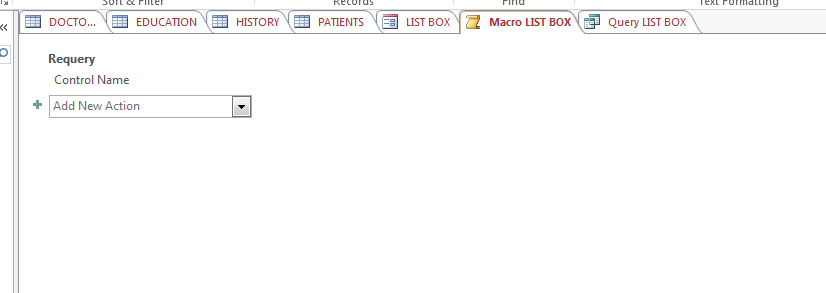


Figure 62: LIST BOX Macro

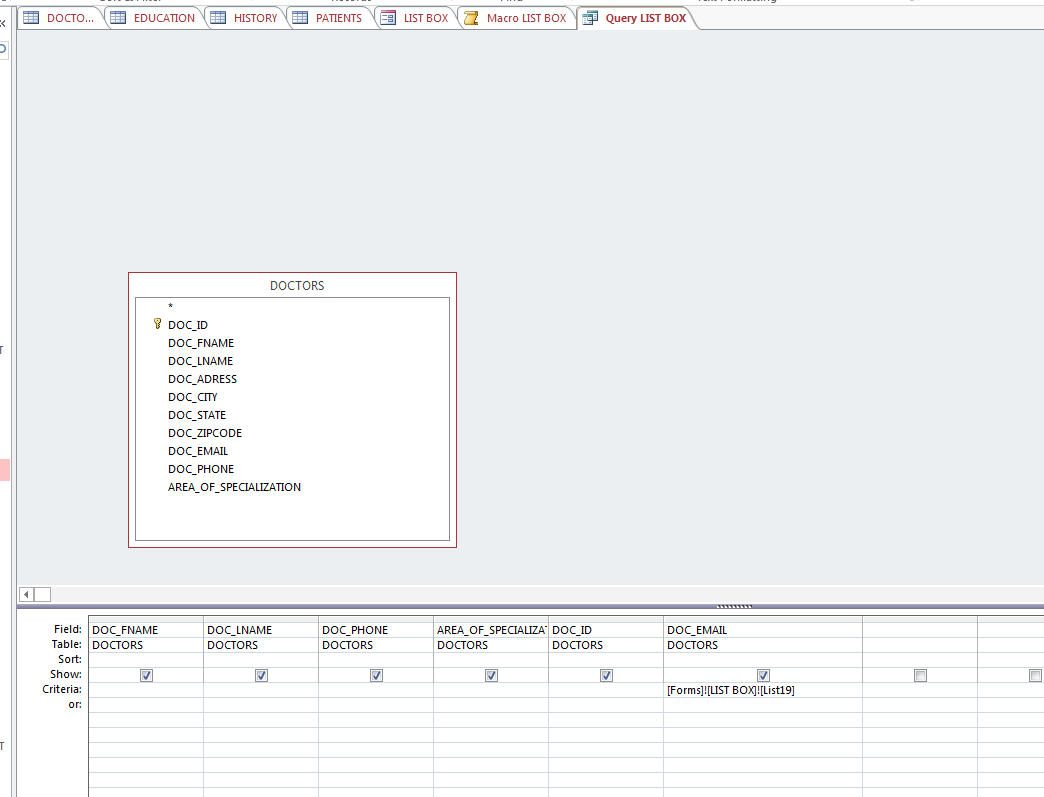


Figure 63: LIST BOX query Design View

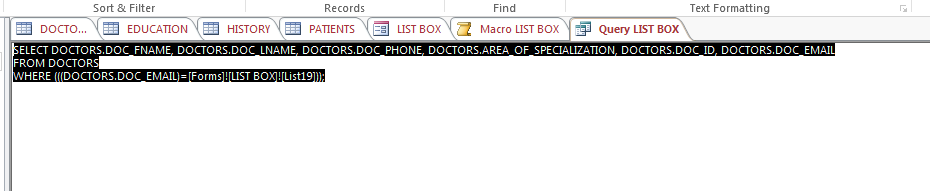


Figure 64: LIST BOX query SQL View

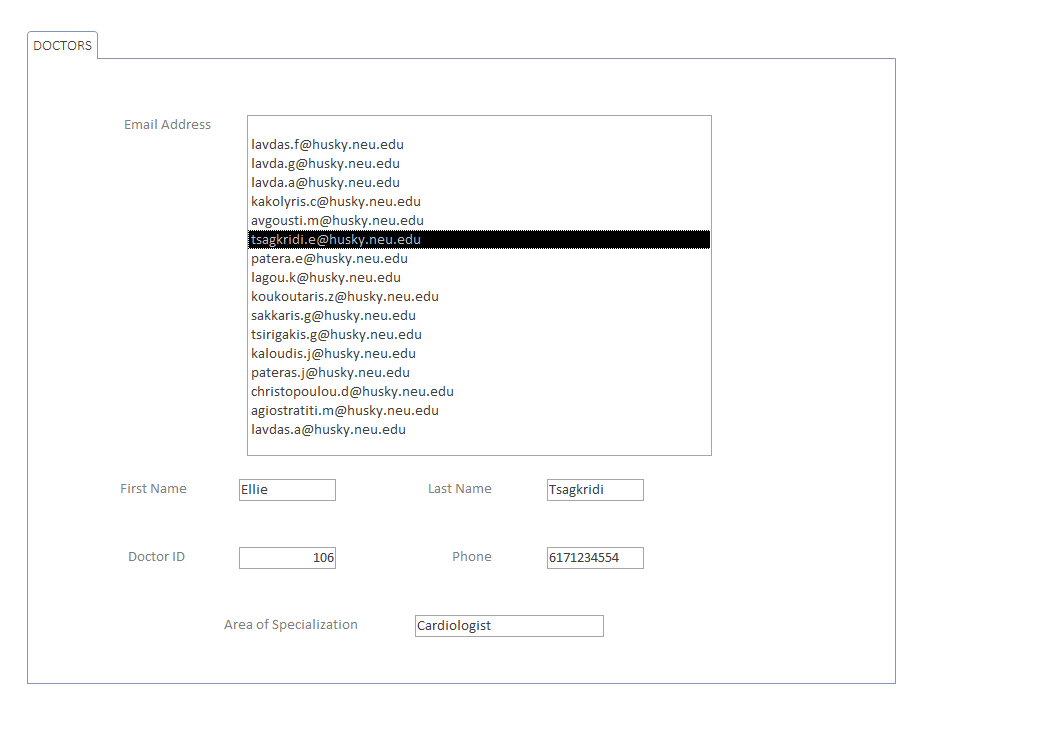


Figure 65: Form View

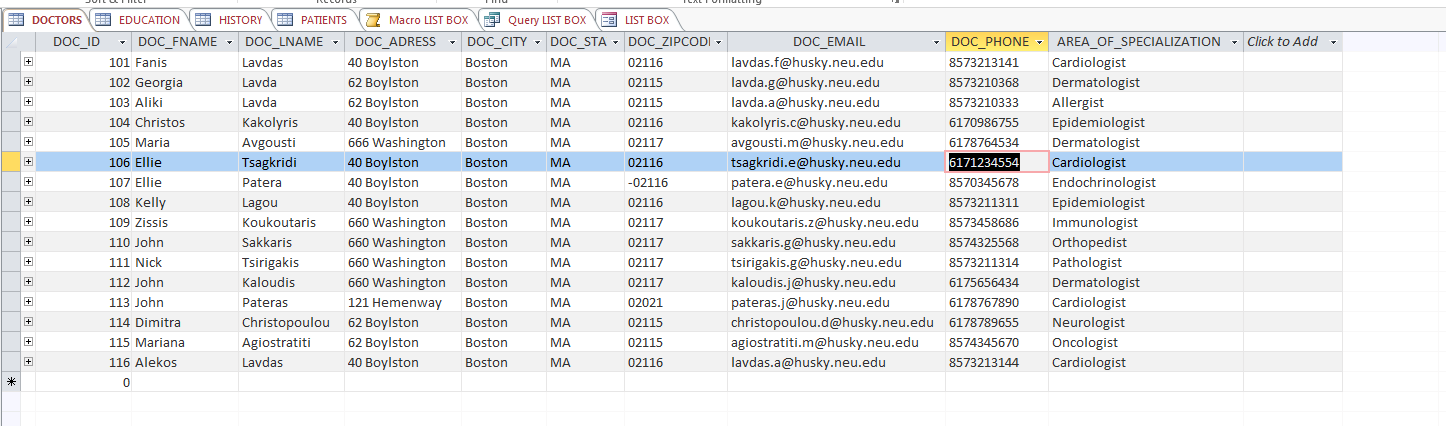


Figure 66: DOCTORS table

1. **Two forms to add data to the DB: three forms were created to add data to the DB and are shown previously in question 1**
2. **Two forms to extract information from the DB**

**Extract Appointment Information from HISTORY Form**

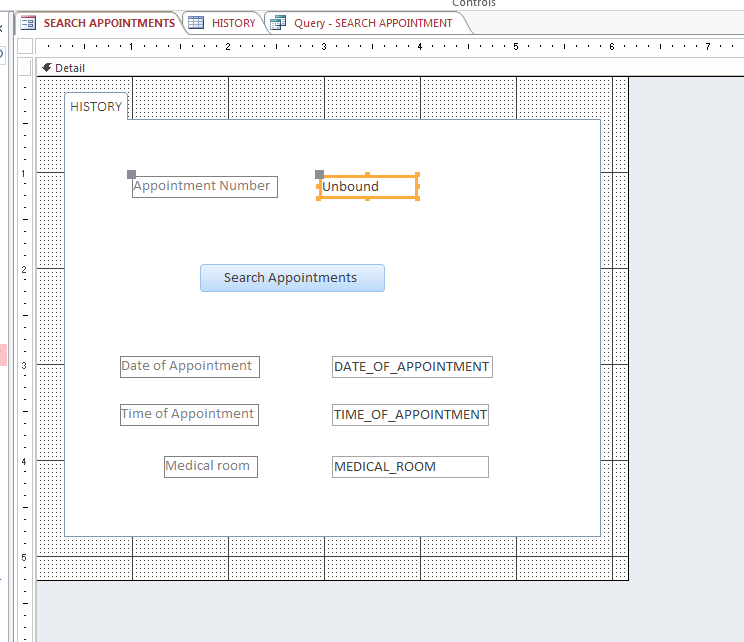


Figure 67: Design View

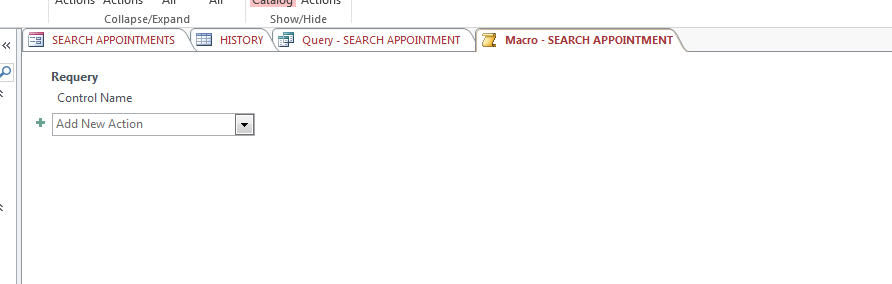


Figure 68: Macro

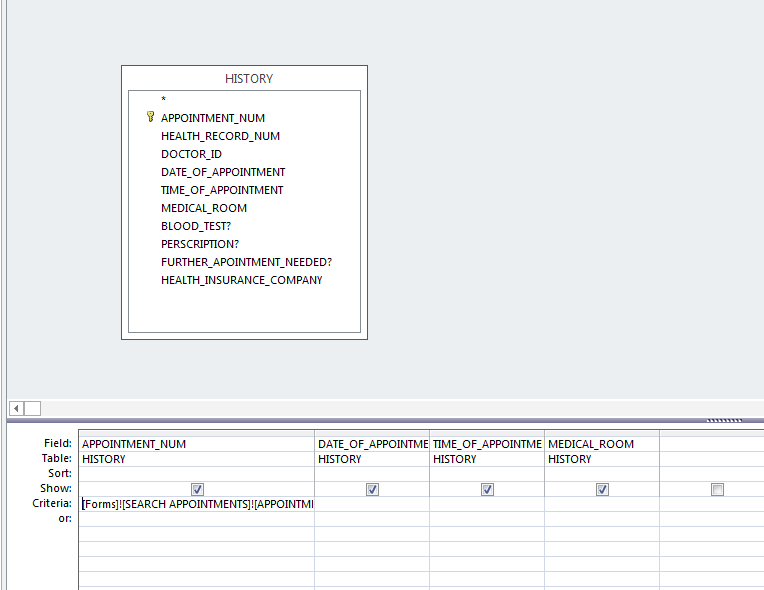


Figure 69: SEARCH APPOINTMENT query Design View

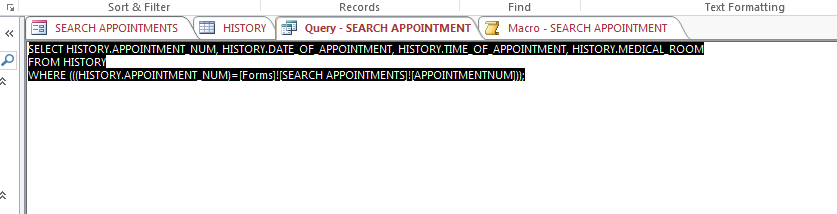


Figure 70: SEARCH APPOINTMENT query SQL view

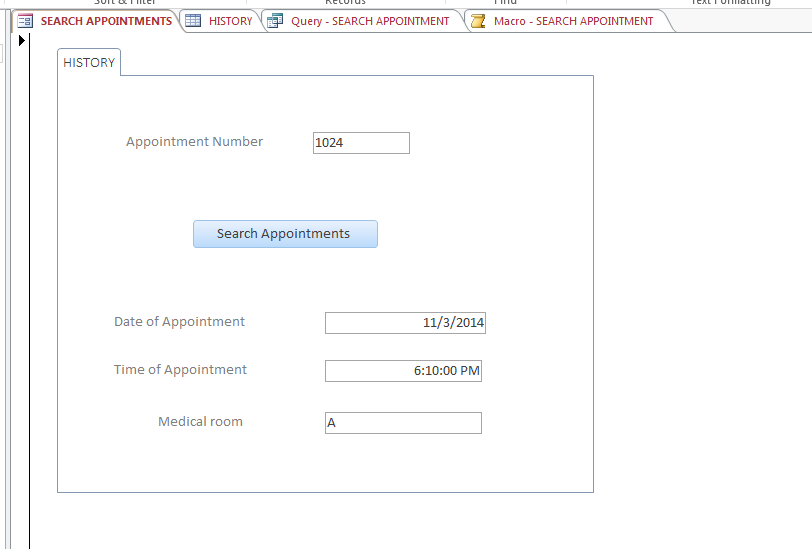


Figure 71: Search Appointment Information Form View

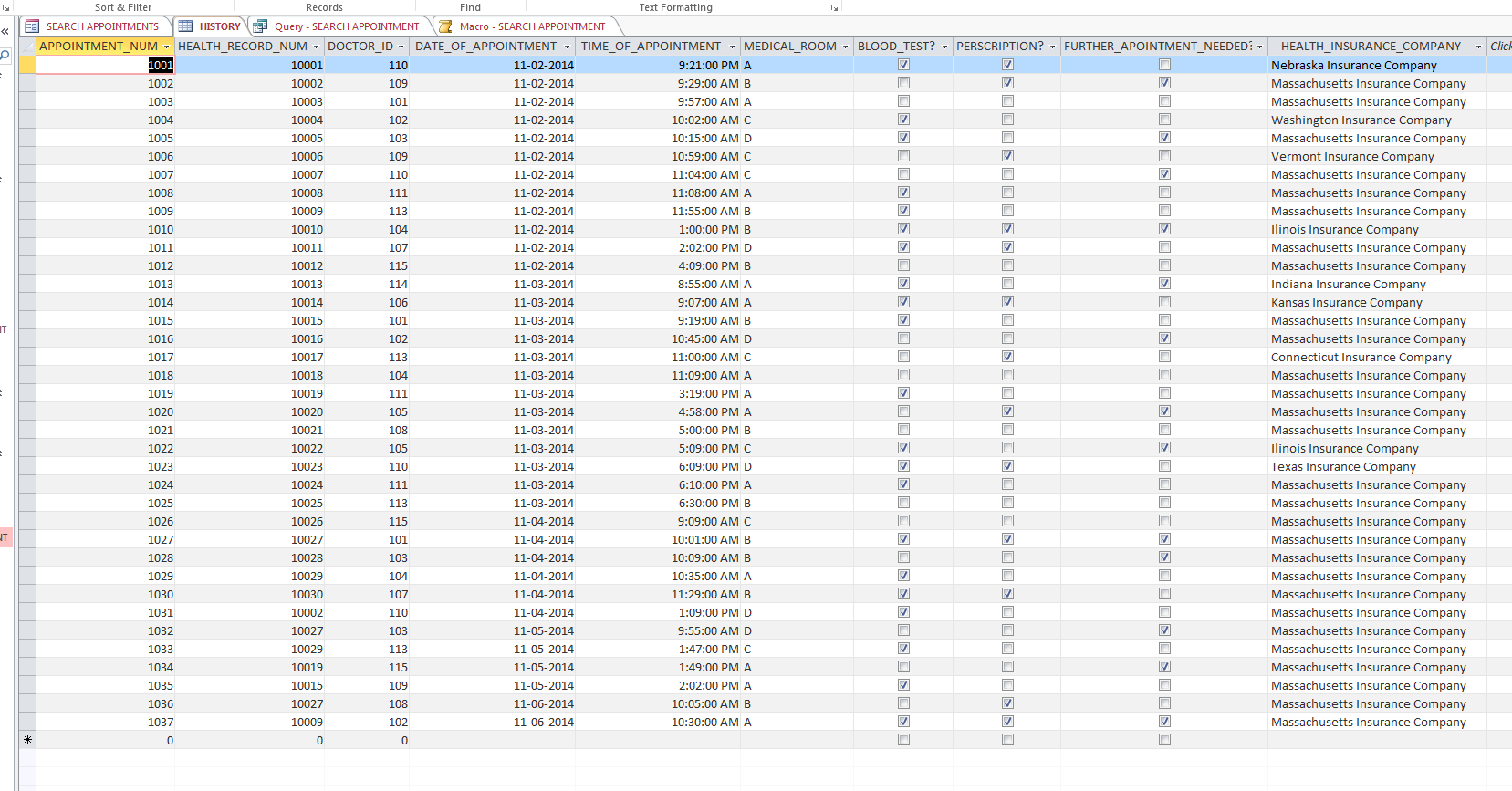


Figure 72: HISTORY table

**Extract Doctors Information from DOCTORS Form**

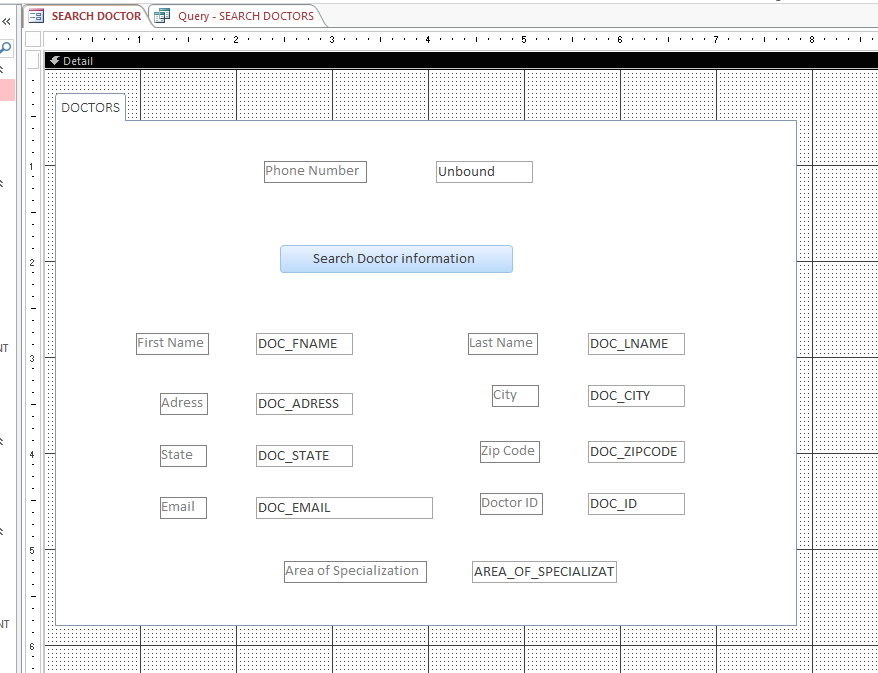


Figure 73: Search Doctor Information Design View

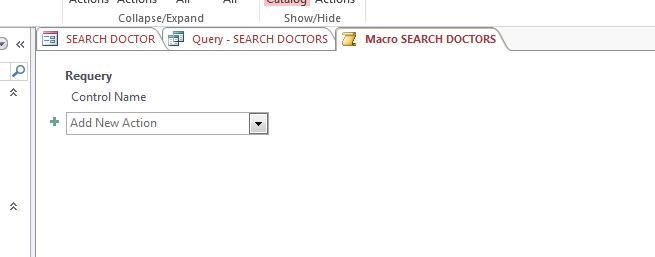


Figure 74: SEARCH DOCTORS macro

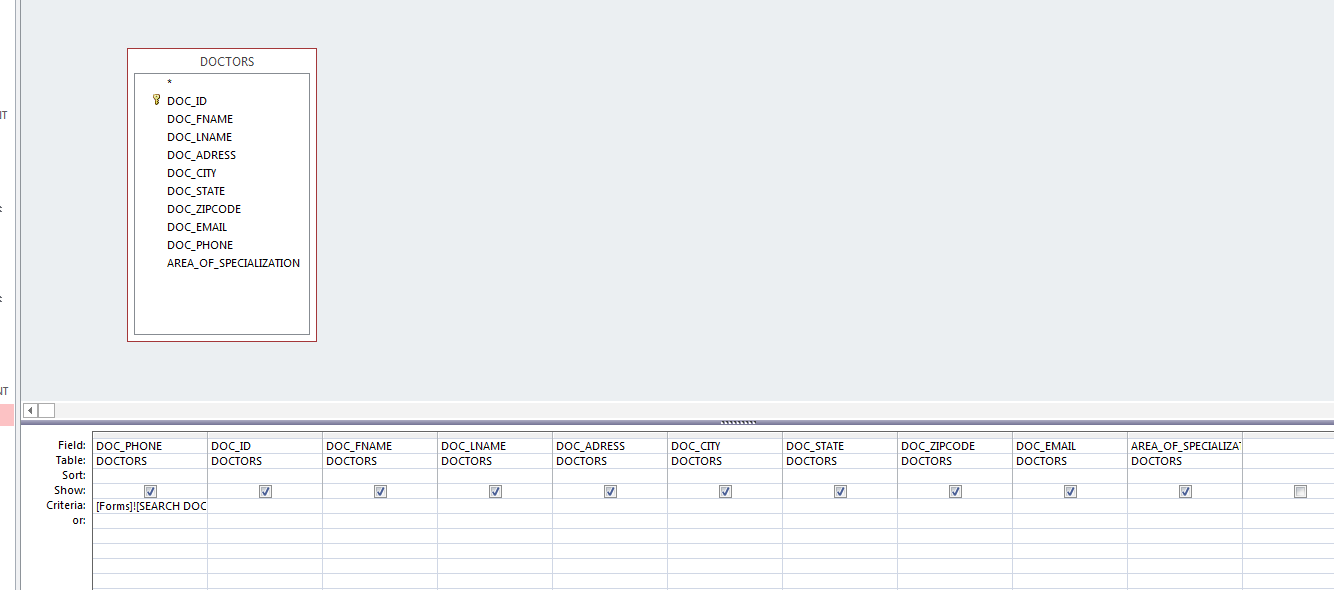


Figure 75: SEARCH DOCTORS query Design View

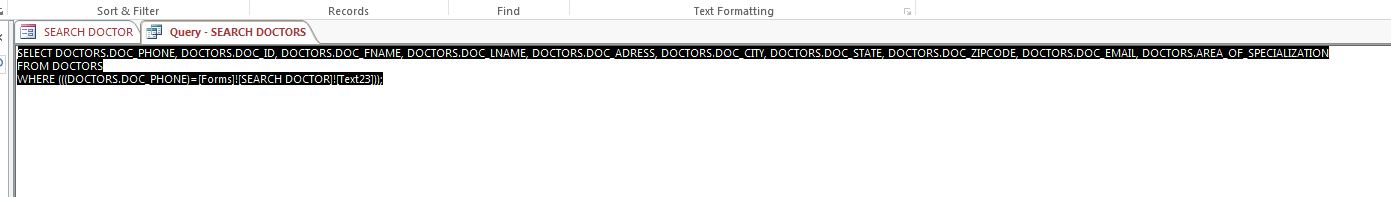


Figure 76: SEARCH DOCTORS query SQL View

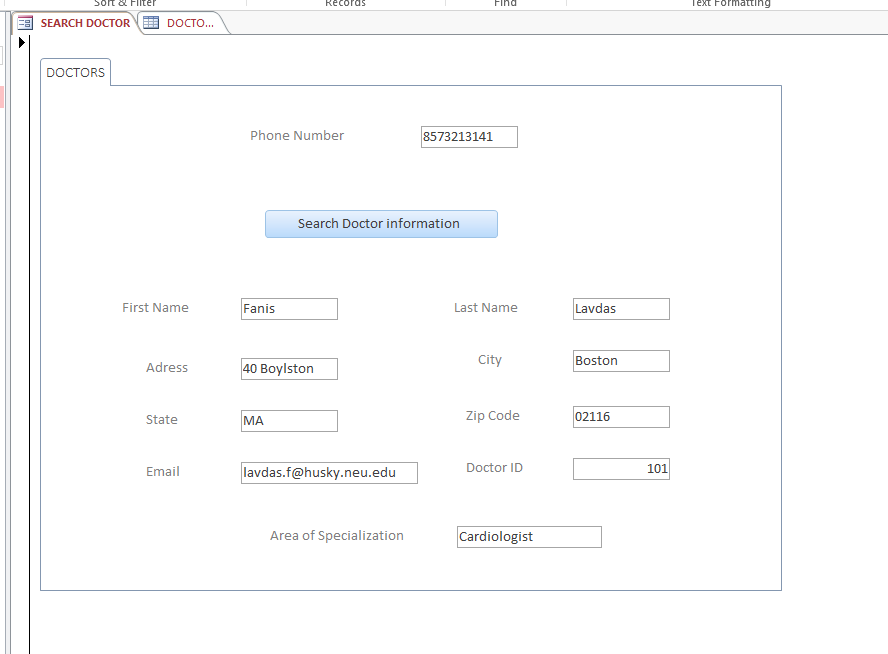


Figure 77: Extract Doctor's Information using Phone Number Form View

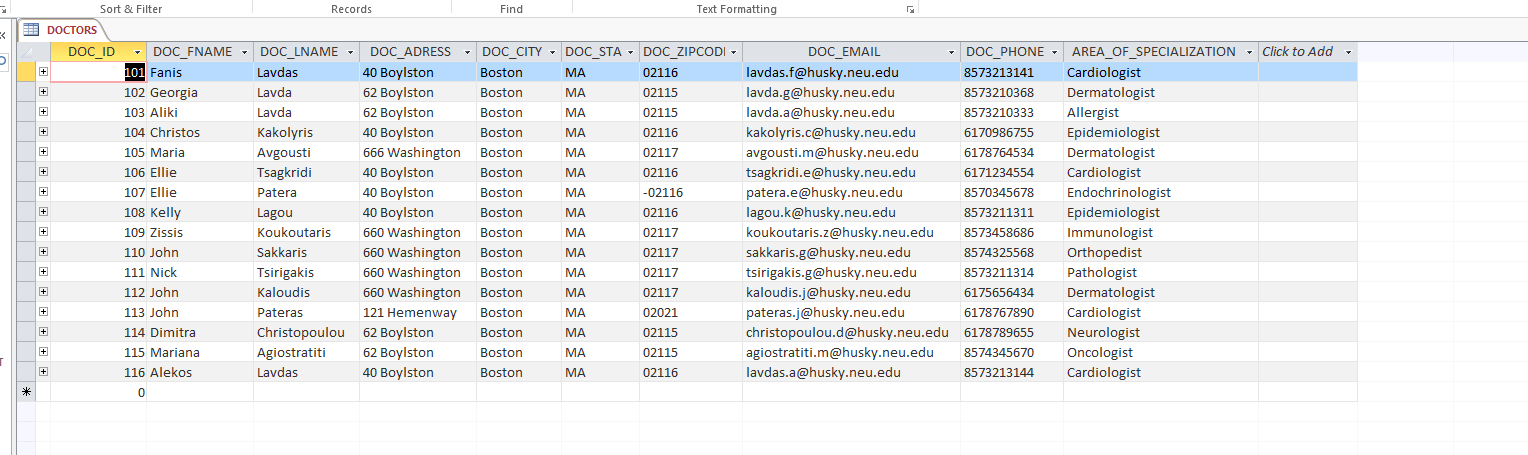


Figure 78: DOCTORS table

1. **Two reports**

**Report that shows appointment History**

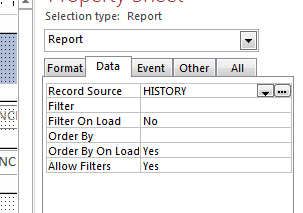


Figure 79: The report takes info from the HISTORY table

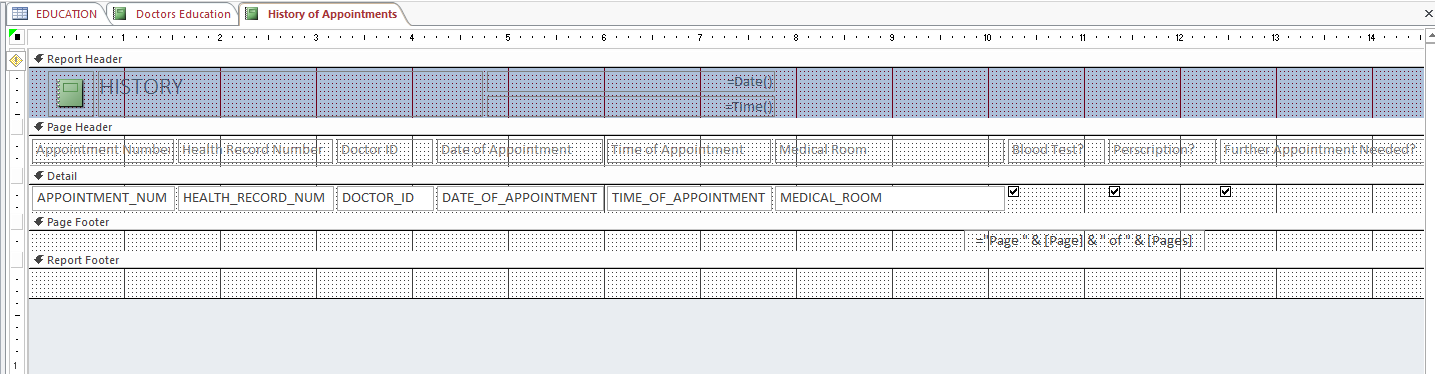


Figure 80: Report Design View

Report is labeled “History of Appointments” on Database file. The whole report is too big (37 records) to fit in one page therefore it is better to be seen directly from the DB.

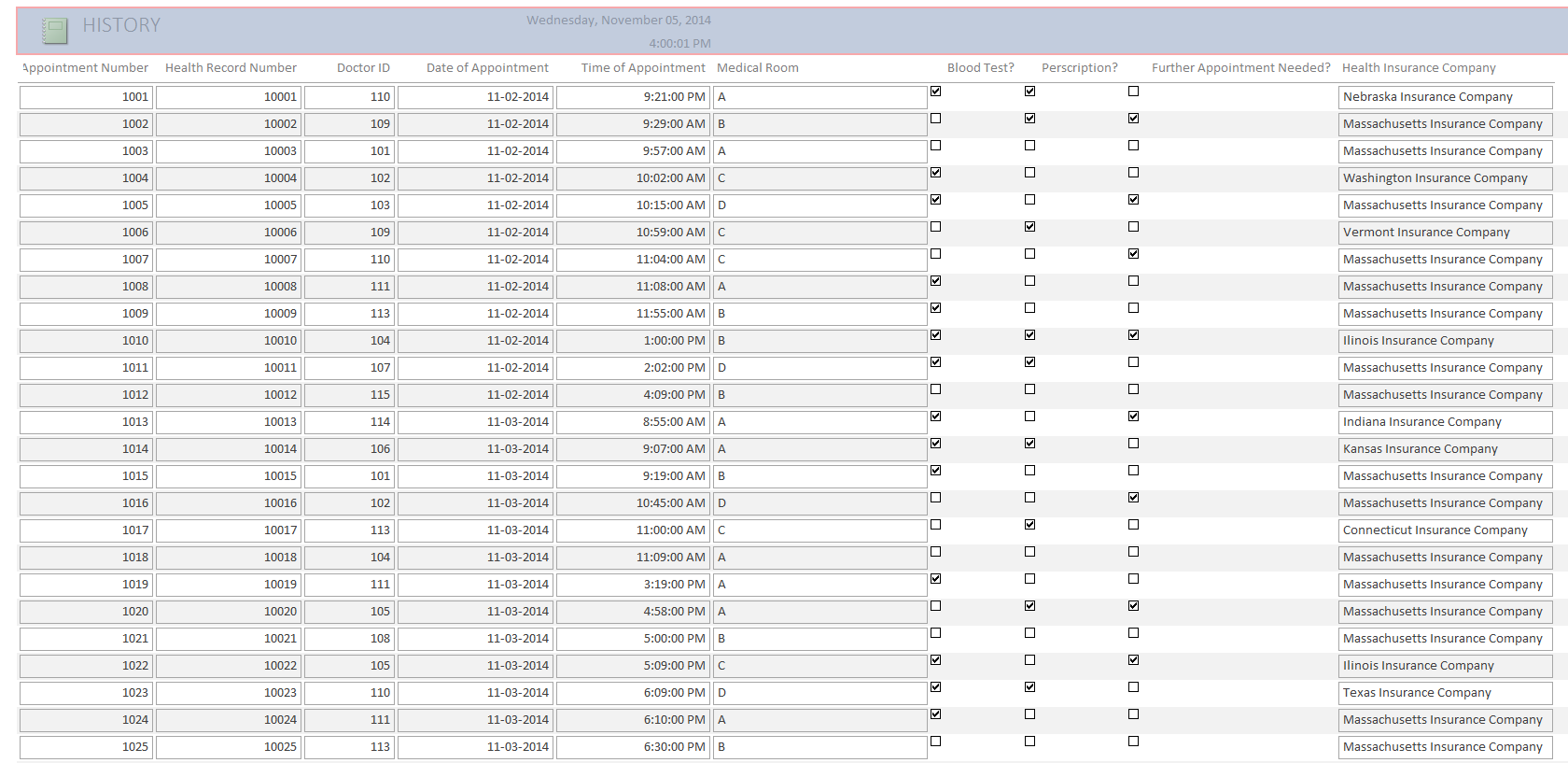


Figure 81: Report for appointments 1001-1025



Figure 82: Report for the remaining appointments 1026-1037

**Report that shows Education information for Doctors**

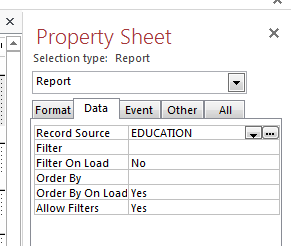


Figure 83: report retrieves info from EDUCATION table

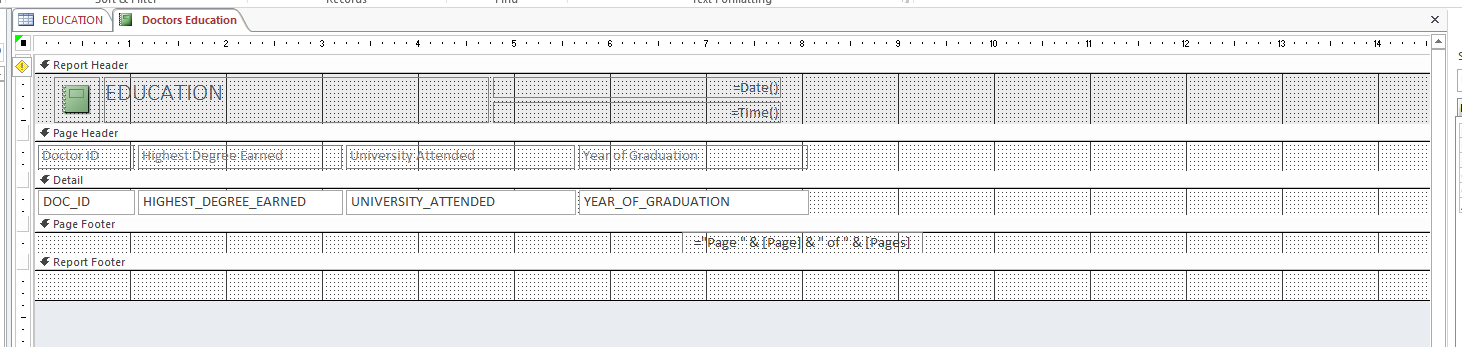


Figure 84: Report Design View

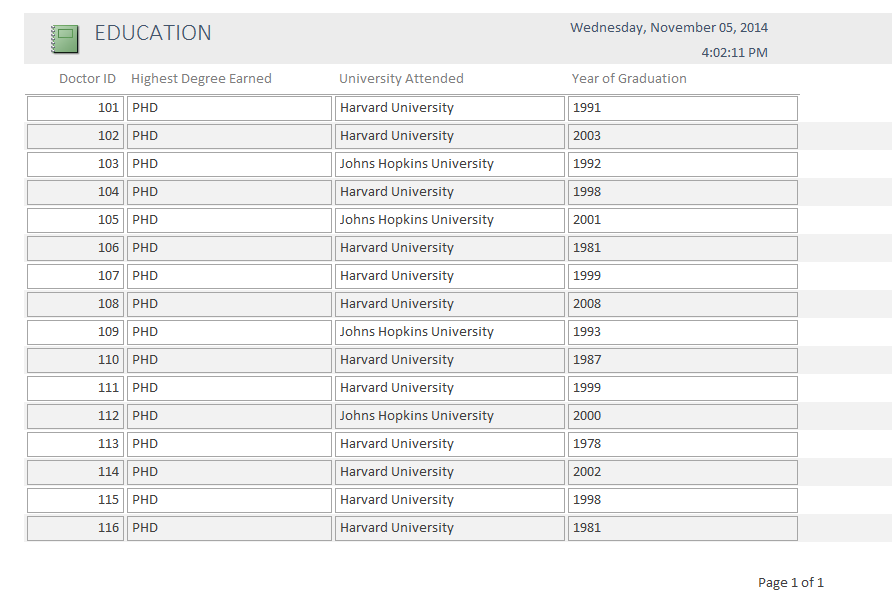


Figure 85: Report for Doctors Education

1. **One column chart and one line chart**

**Column Chart**

**Graduation Year of each Doctor**



Figure 86: Chart Design View

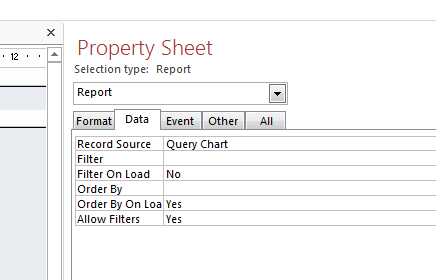


Figure 87: The chart retrieves information from the Chart query

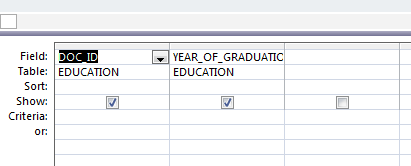


Figure 88: Chart Query Design View

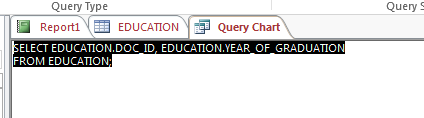


Figure 89: Chart Query SQL View

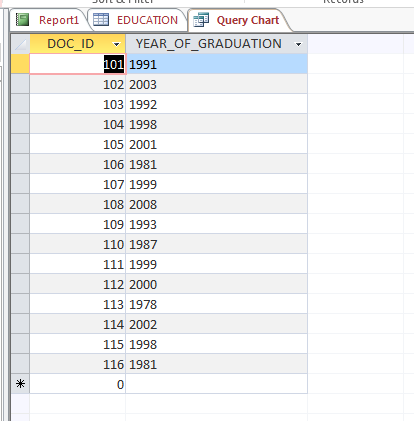


Figure 90: Chart Query Datasheet View

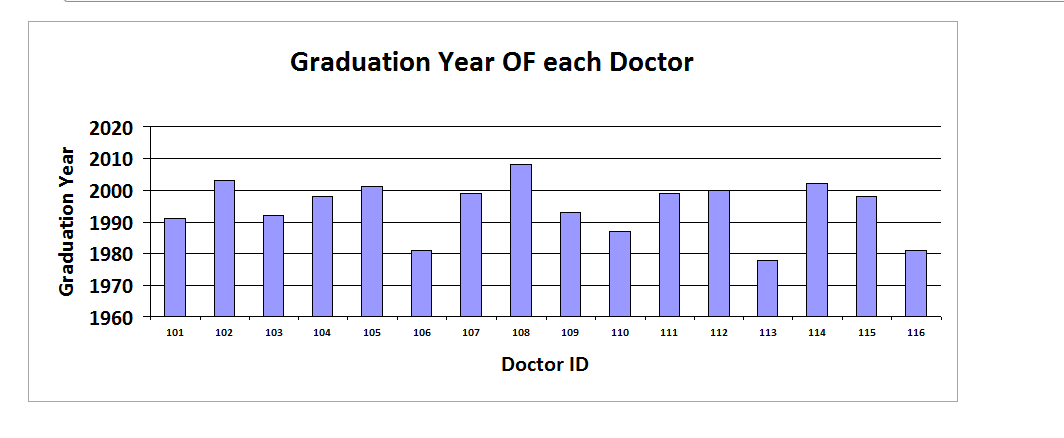


Figure 91: Column Chart Report View

**Line Chart**

**Graduation Year of each Doctor**

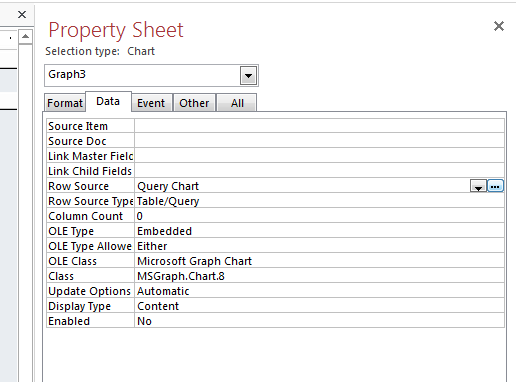


Figure 92: Chart retrieves info from Chart Query

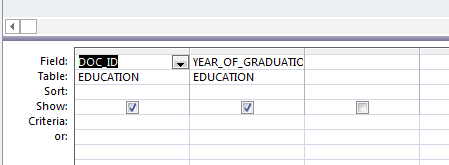


Figure 93: Chart Query Design View

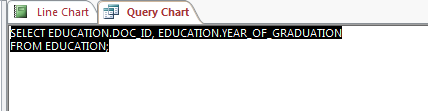


Figure 94: Chart Query SQL View

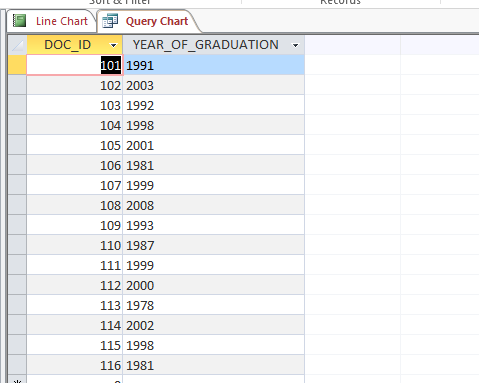


Figure 95: Chart Query Datasheet View

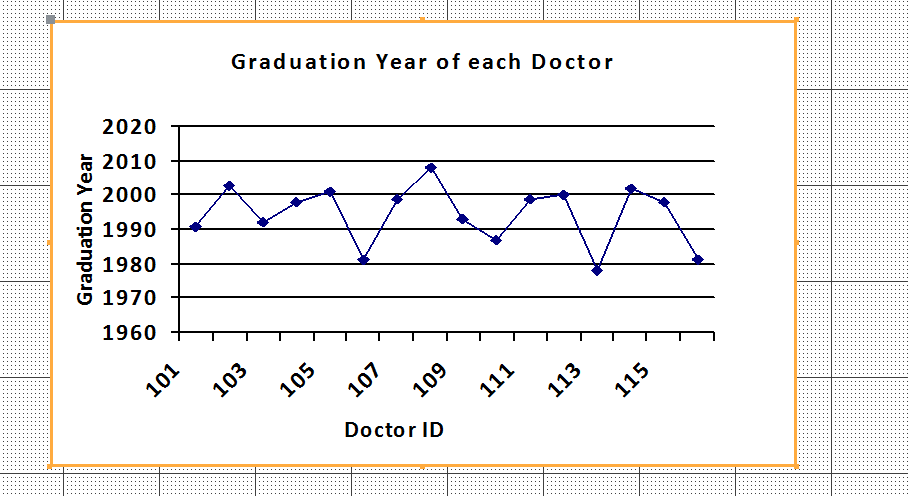


Figure 96: Line Chart Design View

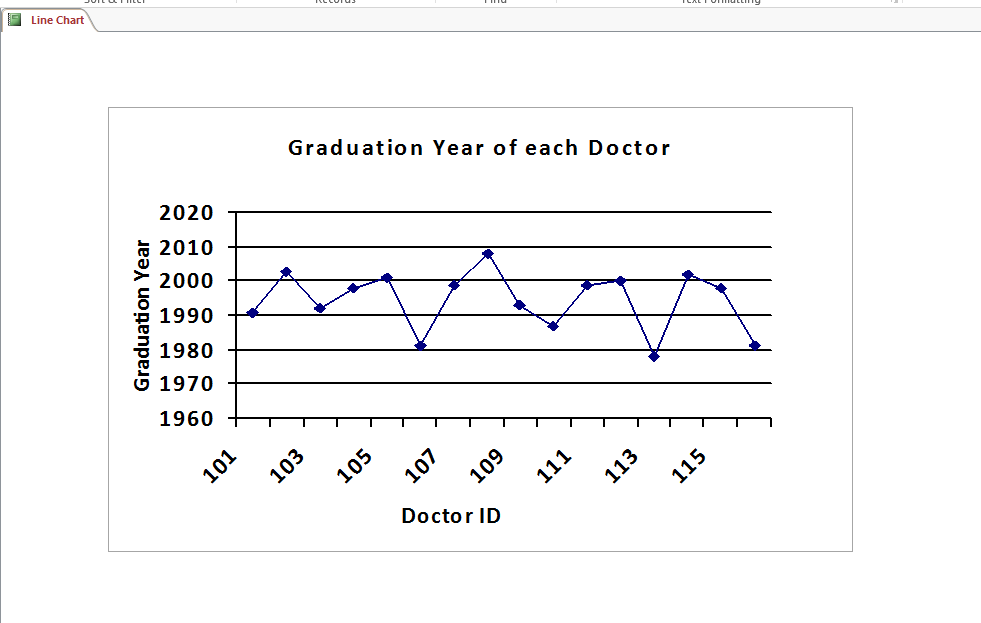
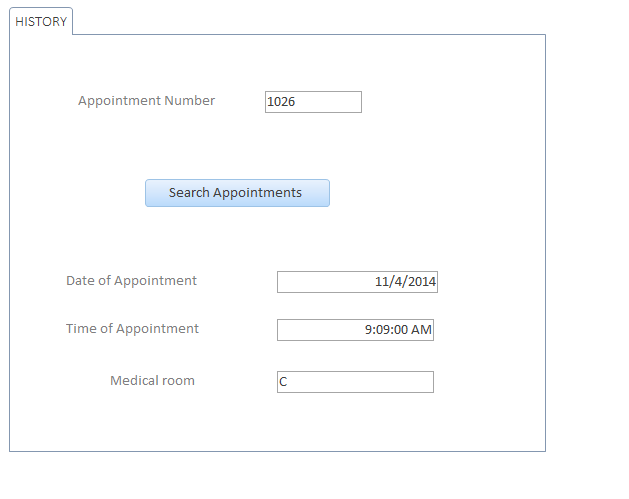


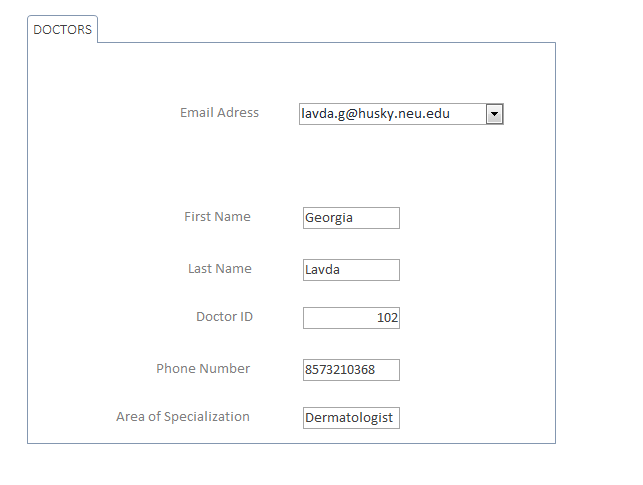
Figure 97: Line Chart Report View

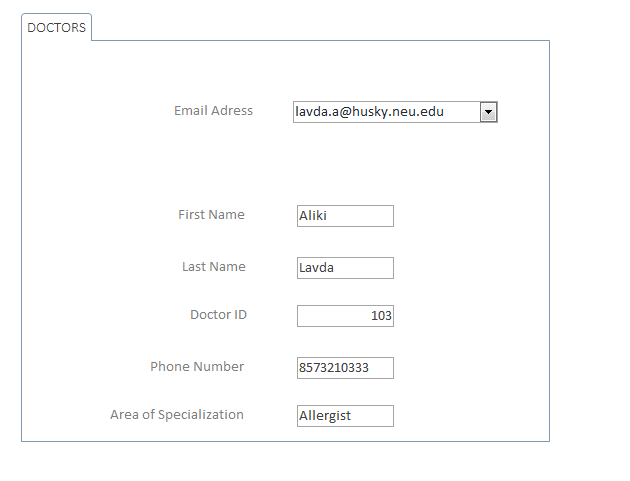
1. **Testing and Use Cases: Include sample cases that illustrate the use of your DB application. Describe at least three cases (description and necessary screenshots)**

**Case 1**: The receptionist responsible for confirming appointments calls the patient to verify an upcoming appointment. At the time when the appointment was originally scheduled, a unique appointment number had been assigned to the appointment. The receptionist wants to verify all scheduled appointments for the next day, so she makes a list of all appointment numbers and calls the number that corresponds to the Health Record Number that has scheduled the appointment. Every time she call a patient, she inputs his/her unique appointment number and by using the SEARCH APPOINTMENTS form, she can retrieve the information about the time, date and place of the scheduled appointment. She can then ask the patient to verify the appointment based on the information extracted. For example, the receptionist want to verify the upcoming appointment with Stelios Lavdas (Health Record Number 10026 and Appointment Number 1026). She calls Mr. Stelios Lavdas and asks him to verify the following time and date of this appointment, as well as reminds him at which medical rooms he has to go for his appointment.



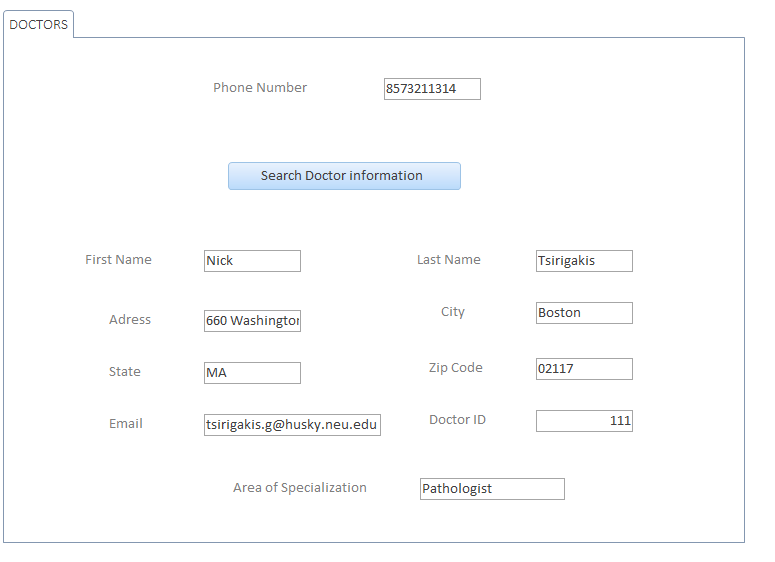
**Case 2:** Doctor’s usually give their business cards to patients. A patient has received a business card from a doctor, however the rain almost destroyed it. Neither the first nor the last name can be read from the card and some phone digits cannot be read as well from the card. The only thing that the patient can read from the card is the email address. The patient then calls the clinic where the doctors work and asks whether they can provide him with the information he wants concerning one of the doctors that work there based only on the email address he knows. The receptionist is pleased to assist him and pulls out the COMBO BOX form, where she picks from the combo box the email address and she can retrieve the doctor’s information based on the email. The patient says to the receptionist that she can read [lavda.g@husky.neu.edu](mailto:lavda.g@husky.neu.edu) from the card but he is not sure whether the g letter is a g or an a. The receptionist says to the patient that there is no problem and goes ahead to retrieve the information for both emails:





From the information retrieved she can know ask some questions to the patient to determine which one of the 2 is the doctor she is looking for. For example, she asks the patient if he remembers what the doctor’s area of specialization was: allergist or dermatologist. She can also tell her both full names in case the patient remembers it. Both are ways that the receptionist can use to give to the patient the information he/she requires.

**Case 3:** The person responsible for the clinics accountings wants to send a bonus to one of the doctors working there as he has recently completed 15 years working for the clinic. He/she can use the SEARCH DOCTOR form to input the Doctor’s phone number and retrieve all the information that are on the system for him. To be more specific, Nick Tsirigkakis, who graduated from Harvard University in 1999 and started working for the clinic immediately after has just completed his 15th year in the clinic. The accountant can use the doctor’s phone number from the DOCTORS table (8573211314) and find all the information for him:



The accountant is only interested about the address, city, state and zip code. The accountant then uses the above information to send to Nick Tsirigakis home an envelope containing his bonus accompanied by a trip.

***DB Theory Questions***

1. This DB design can easily expand to include all the appointments ever made in the clinic. This database design only included the appointment information for all appointments made between November 02, 2014 and November 06, 2014 as well as information for patients, doctors and their education. The number of doctors would stay constant in case of expansion of the DB as the DOCTORS table shows all doctors that currently work there, regardless of the dates that we focus on. As it can be noticed all doctors have had at least one appointment in their schedule during this period of days. Growing the database to include a broader period of dates would also increase the registered number of patients. The table PATIENT only shows the patients that had at least one scheduled appointment during the specified dates, therefore growing the database would mean that more patients would be added in the PATIENT table. Adding patients can be easily supported by the Add Patient Form. Finally, as the period of dates broadens, the number of scheduled appointments will increase. This DB growth can be supported by adding one more form for adding information for a new scheduled appointment.
2. Most of this question was included in the previous questions. 3 forms already exist to cover adding data to the 3 tables. Therefore, if further growth would be implemented in the DB, a new addition form should be added in order to add appointment information into the HISTORY table. Some more forms for extracting information are also needed to be added. The more extraction forms exist the better, as there are so many the different combinations of information you need to extract based on the information you already possess. For example, when a the receptionist has to arrange a further appointment with another doctor, she might need to extract some of this appointments information, such as the Health Record Number and the Health Insurance Company that the patient has given their appointment number or their first and last name or the combination of time, day and medical room, and many other combinations.
3. The best way to make the front-end idiot-proof is to design a very solid back end. Input masks, validation rules and validation texts definitely help in making the DB idiot-proof. This specific database is designed to be used by an experienced person with good knowledge in how the database works as well as how to solve any problems. In my clinic, the receptionists will operate the DB, therefore this will be the only job they have; input information to the system and retrieving information to arrange appointments. Therefore, I expect my receptionist to be experienced and trained in Databases and to be able to identify easily and quickly any possible mistake and correct it. Although the database is designed well and it is as simple as possible, there are always possible mistakes that can be made and it is essential that the DB is designed in such a way as to be convenient to correct a mistake.
4. Every query I wrote I used the Access Design View as I find it much more convenient, easy and faster to use. However, there were a lot of times when I looked at the SQL code to determine what was going wrong with my query. I find the SQL view more convenient when it comes to figuring out why your query doesn’t work the way it was supposed to.
5. From this project I took a nice idea of how much work an actual database of a big company needs, how many people should work on it, how many hours they should spend building both the back end and the front end. I took a good idea of how much work is needed for companies with thousands of employees and millions of transactions and invoices to keep track of everything and store all the information.

For this project I spend approximately 18-20 hours. I devoted a lot of time in building the back end, adding 38, 30, 15 and 15 records respectively in my 4 tables, amongst which 3 of them had 10 columns and one of the 4 columns. I wanted to make it as realistic as possible, that is why I added that 30 different patients visited the clinic between November 02 and November 06, 2014 and these 30 patients scheduled 38 appointments.

I also spent a lot of time organizing the database before even start building the back end of the database. I also wrote the business plan and the business rules in the beginning and then building the back end of the DB accordingly. Therefore, after allocating more than two hours in just planning the backend and writing the business plan, I never had to revise anything in my DB before achieving the desired objectives. I didn’t even have to perform normalization, as I created the tables in such a way that no normalization would be needed.