CONTENT

Work Annotation

Almost every organization and business need a highly maintained database in order to keep their information organized. It is very important to design the database to cover all needs of the domain because changing the format of the database later can cause a loss of information. First of all, the design should start with a description of the problem. Gathering more information about the problem would make it more specific so it will be easy to find the best solution. To specify the problem as well as possible, it is a good idea to talk with the owner of the business or the leaders of the organization. Another best way is that you can analyze similar organizations to see what are the similarities and differences that way we will also have an idea of what to include in the project. In our case, the problem is to build a database for a public hospital. Because this database would be not used in real life we will build this database based on assumptions and similarities with a real public hospital.

Secondly, it is best practice to find and write the business requirements for the project before starting designing that way we can decide easily how many entities to create, how many attributes should every entity have, and also how to build relationships between entities. Business requirements should cover all the needs and musts of the organization. Besides business requirements, the organization may also want to specify some constraints. If there are no definite requirements for something then it is good to use assumptions. They should realistic in order to be implemented in the database system but at the same time, too much complexity should be avoided.

After completing the steps which are specified above finally, we can start designing our database system. There are two different ways to design a database. The first option is that you can design a database logical model (ER diagram) and you can use the forward engineering tool of the database management software to convert the logical model to a physical model. In a second way, you have to first design a physical database and when you need an ER diagram of the database you can use the reverse engineering tool of the database management software to get a logical model. It does not matter which way will you choose to design a database system but, in both ways, you need to apply all the business requirements which were specified early. It should be not forgotten to use the proper data type for each attribute of the entity otherwise the database can take up a lot of storage space which can cause slowness in the database operations.

A hospital database system will be designed in this project. We will use the first way to design this database system. It means first a logical model will be created then forward engineering tools will be used to convert the logical model to physical. The business requirement of the project can be

found on the next pages. MySQL Workbench desktop application will be used for designing the project for a public hospital. Entities such as doctor, nurse, patient, inventory and room will be included in this project. The entities will be filled with the required attributes. There will be different relationships between the entities and the relationship will be created logically and according to a real-life situation.

It is always good to test the product yourself before transferring it to the client so, at the end of the project will enter some dummy data into our database entities for testing purposes. Several scenarios will be considered and the result will be shown with screenshots to see if the designed database is working properly and as required. The queries will be written in MySQL (SQL) language. SQL joins and functions will be used to write advanced queries. After creating the physical database from a logical model, we will also check our physical database and tables if it is the same as the logical model.

Introduction

In this project a Database system will be created for a hospital. First of all, it is very important to describe the problem in the best way possible. Because Hospital domain was selected for this project we can guess that tables like doctor, nurse, patient, inventory and room are the main tables which should be created.

The main reason for selecting hospital domain is that it can be applied easily in real life and it could be a very good example for practicing database designing and usage of MySQL functions. I will try to build the project very close to a real hospital database so we can implement the real-life scenarios. Project will be designed using MySQL Workbench design tools. We will consider to design a database for a public hospital. Hospital management will be able to store information about doctors, nurses, patients, rooms and inventory belonging to hospital so later the could retrieve that information.

First the project business requirement will be specified according to the needs of the hospital. Than according to specified business requirements the database will be designed. I will prefer to design the logical model for the database first, later the logical model will be converted to physical model. however, it is possible to do it the opposite way (design the database if you need use reversing engineering to get ER diagram later).

Designing the database is the most important part of every database project but checking the database if it works as required is also as important as designing. In order to be able to check if the database is working properly you will need some dummy data. Fill the table with some dummy data and start write some real life and advanced query and inspect the result for the correctness.

1 Business Requirements

This is the business requirements to the hospital database which has doctor and nurse serving patients to treat them. In this hospital they need to keep information about every patient, doctor, nurse and items which belongs to hospital. This hospital is considered as public hospital so nobody needs to pay for their treatment that is why there is no need for any payment information. Any new graduated doctors or nurses are welcome to work in this hospital so there is not specific work experience requirement for them. Every patient needs to registered while entering the hospital. First name, surname, entrance date, disease and a room number are required to register a patient. A patient ID will be attached to every patient by computer automatically. Exit date information can be updated whenever the patient is leaving hospital.

Information regarding a doctor in hospital must include an id, first name, surname, salary, phone number and the section that he works in. Id information will be added automatically when doctor is employed in hospital. A doctor can treat zero or many patients at the time. Doctors are able to work with nurses; one doctor have to work at least with one nurse but he/she can work with more than one nurse at the same time.

Than we have nurses which work in the hospital, id, first name, surname, salary, phone number and the section information must be also included for each nurse. One nurse can work with only one doctor at the time but a nurse can take care of more than one patient at the same time. Because nurse have to work with one doctor the doctor ID information will be also included for nurses.

There are rooms in the hospital where patient will stay while they are treated. Room may have more than one bed so it shared with several patient. There could be empty rooms in hospital where no patient is staying. Information such as ID, floor and number of beds must be stored for the rooms.

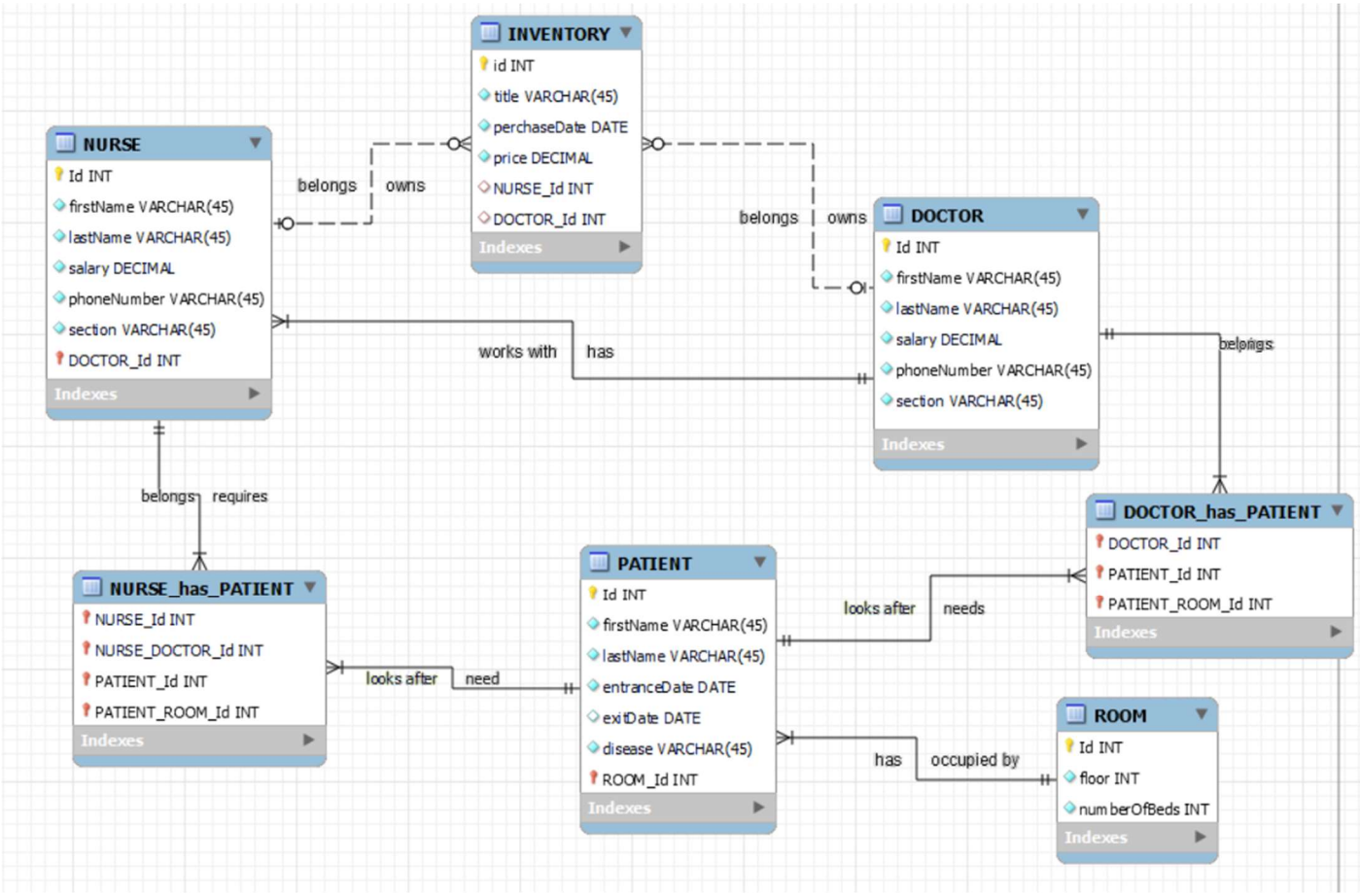
Several items for instance chairs, computers, machine and other medical instruments belong to hospital is also needed to be track down. ID, title, purchase date and price information must be registered for each item in hospital. Some items such as computer, office tables or chairs can belong to a specific doctor or nurse however other item could not belong to any one and they could use just for hospital.

To build a database for this selected domain (Hospital) we need entities such as doctor, nurse, inventory, room and patient. These table should be connected with each other logically. It is good to start building tables and relationship in given order because so we will not end up in a mess.

2 Logical Model Design

In this section the designed database will be described with phots. First database will be described generally than each table will be described by itself. The hospital domain was selected for this project so we included entities such as doctor, nurse, patient, room and inventory. Of course, two entities (Nurse_has_Patient & Patient_has_doctor) has been created automatically because of the relationship between the entities. Relationships between tables are selected logical and almost every relationship contains a caption from both sides to be more explanatory.

Picture bellow shows all of the table created.

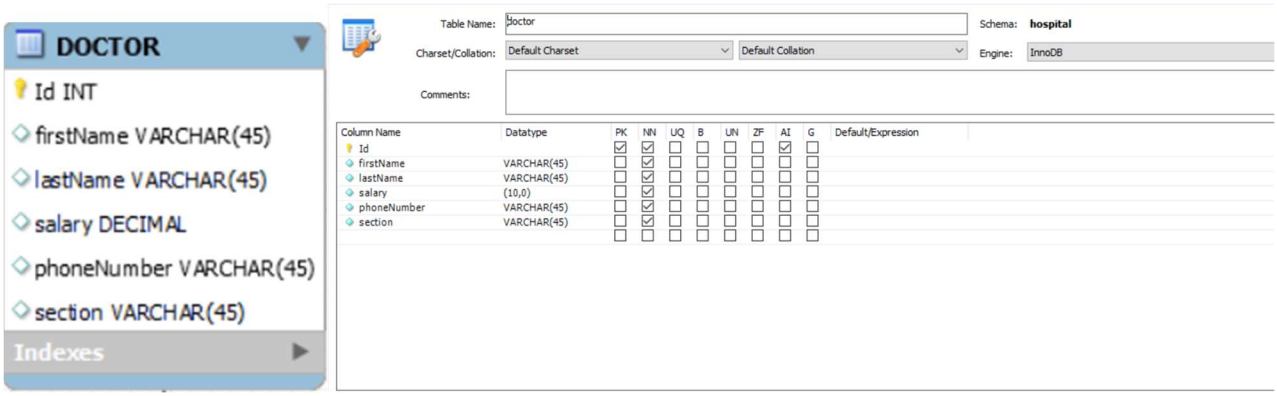


1 Figure. Hospital database logical model

We can see a logical model of the designed database. Later we will use the forward engineering tool of the MySQL Workbench to convert this logical model to physical. Every table has some attributes which are used to collect need information for every entry in the database. The proper data type was selected for each column in the table. Every table also contain an ID column which will give as ability to specify every record in the table with a unique ID number. Because of the relationships between tables each table also contain one or more foreign key attributes.

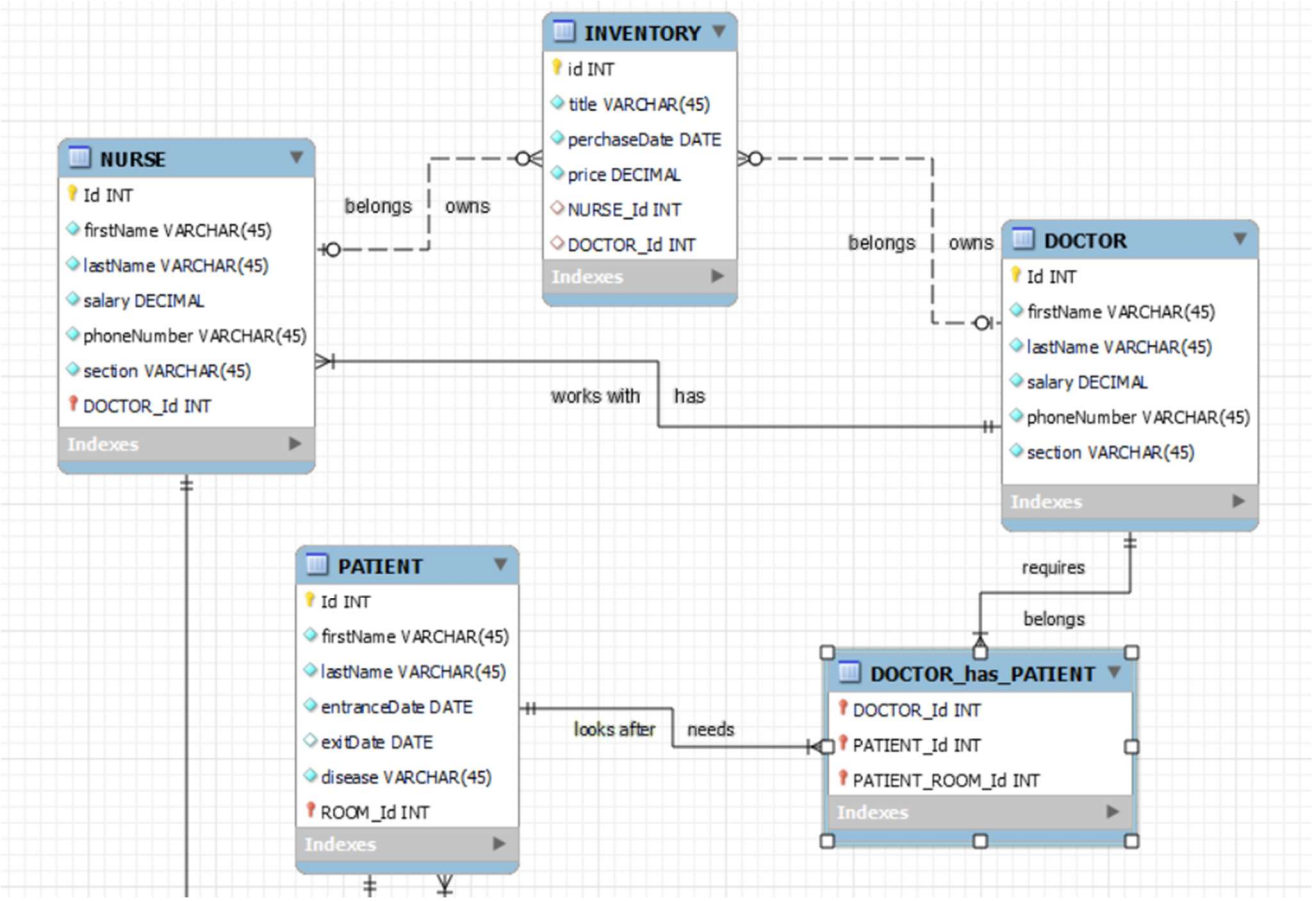
Doctor Table:

It is clear from the name of the table that it stores information regarding doctors working in the hospital. This table has (ID, first name, surname, salary, phone number and section) attributes.



2 Figure. Doctor table | 3 Figure. structure of Doctor table

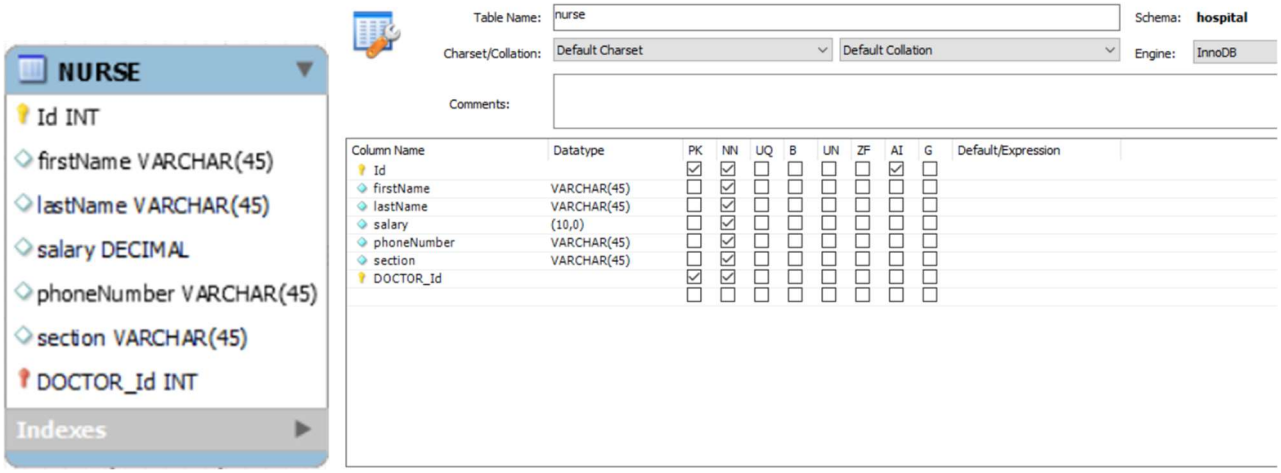
This Database also has other tables which are somehow related with to doctor table. For example, a doctor can work with one or more than one nurse, some items like computers, chairs and medical instruments can also belong to a particular doctor, and most important one doctors work to treat patients, so a doctor can have one or more than one patient at the time. The relationship of doctor table with other tables are shown in the picture bellow.



4 Figure. Doctor table's relationships

Nurse Table:

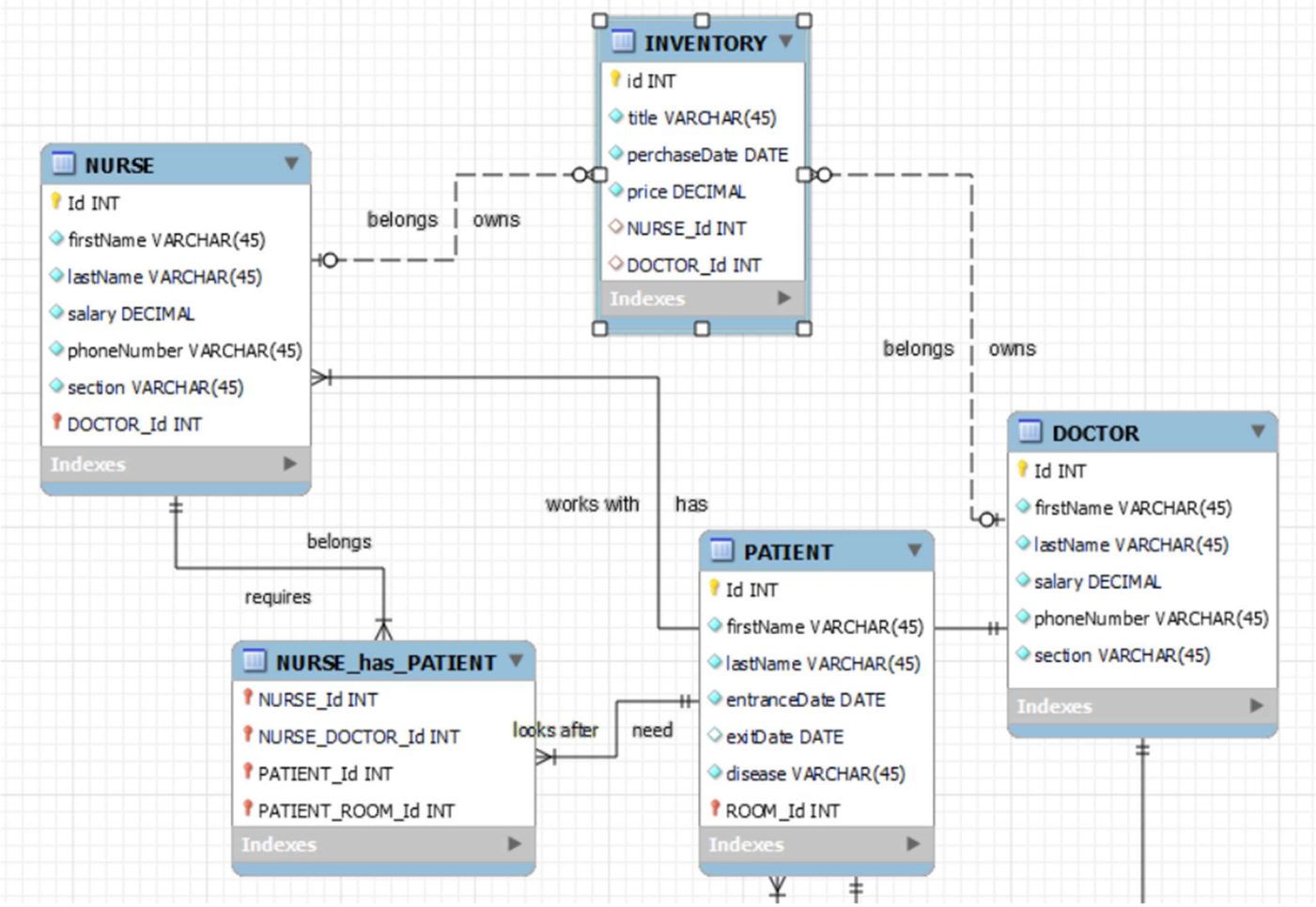
The second main table of this database is called nurse which store information about each nurse working in the hospital. This table is pretty similar with doctor table. Nurse table contains attributes such as (ID, first name, surname, salary, phone number, section and also a foreign key doctor_Id).



5 Figure. Nurse table | 6 Figure. structure of Nurse table

This table have relationships with doctor, patient and inventory table. A nurse can only work with on doctor at the same time. Some items such as medical instruments and computer may belong to nurse but it is not mandatory. A nurse can take care of one or more patient at the same time also

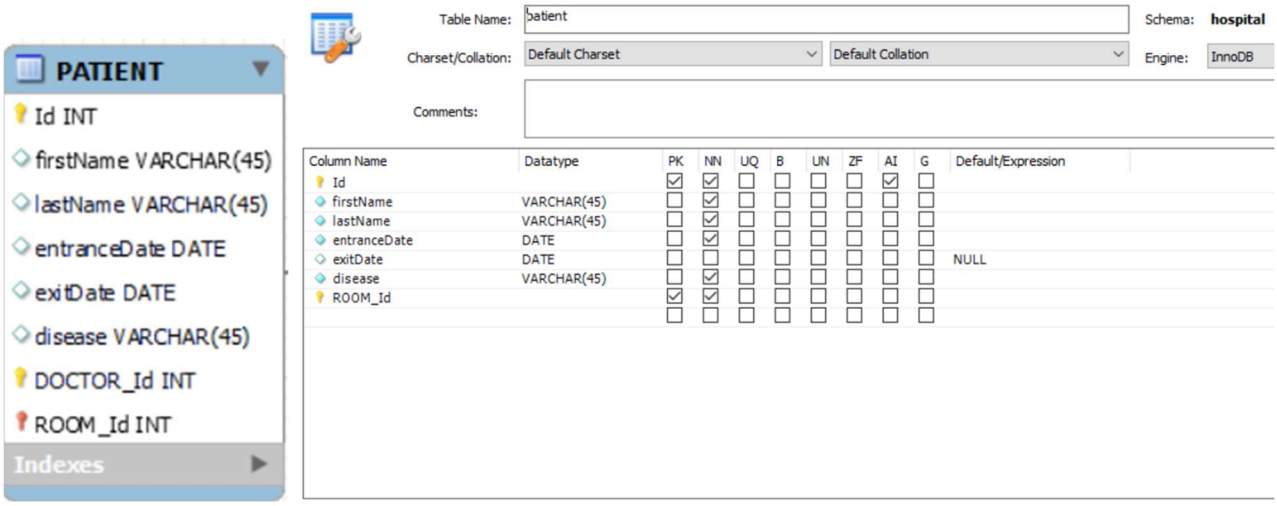
more than one nurse can take care of the same patient. because of the last relationship (many to many) between nurse and patient table a new table named Nurse_has_Patient has been created automatically. To be clearer these relationships are shown below.



7 Figure. Nurse table's relationships

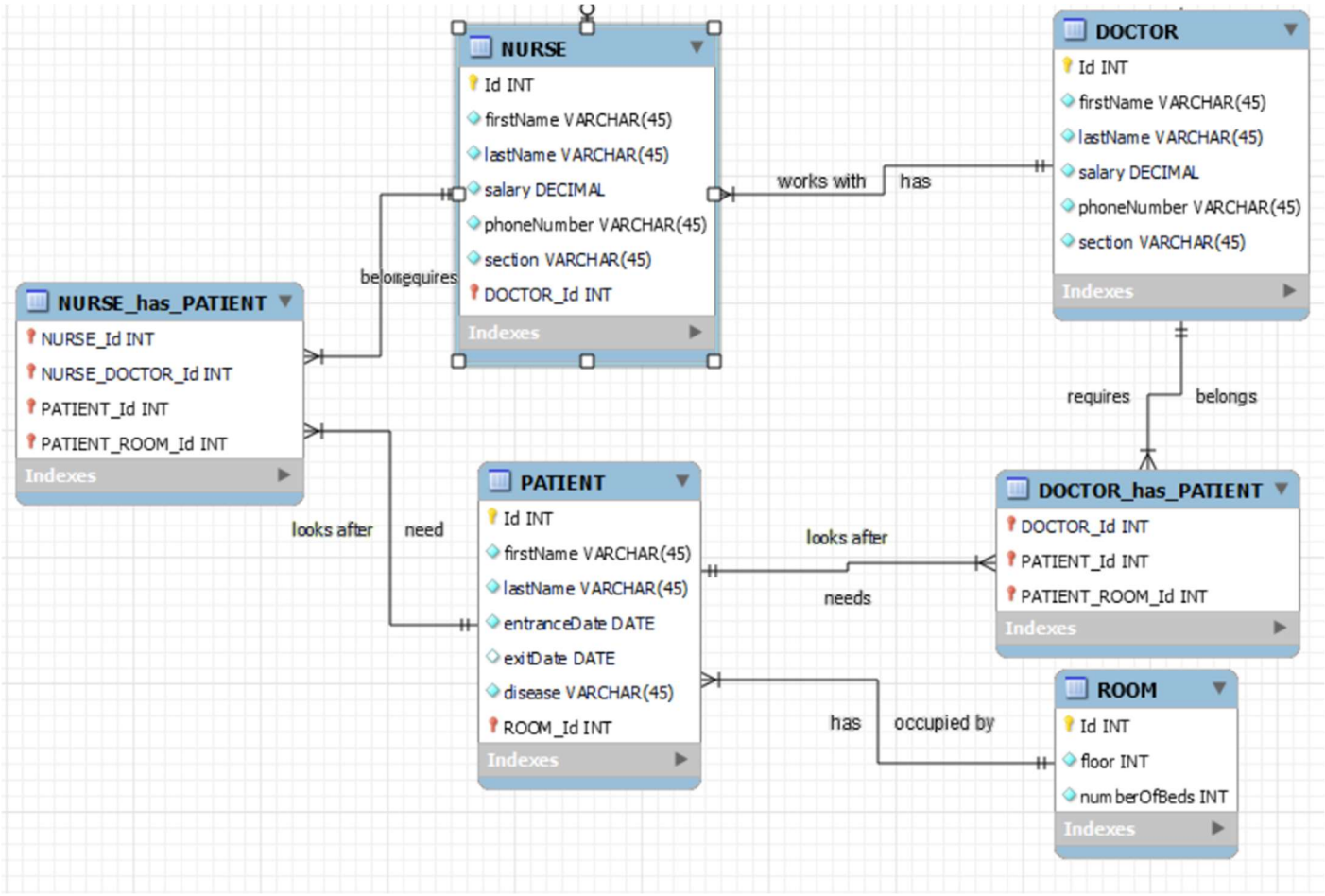
Patient Table:

Patient table is also one of the main tables in this project. Patient table has (ID, first name, surname, entrance date, exit date, disease and foreign keys doctor_ID, room_ID) attributes. Information behalf of the patient will be stored in this particular table.



8 Figure. Patient table | 9 Figure. structure of Patient table

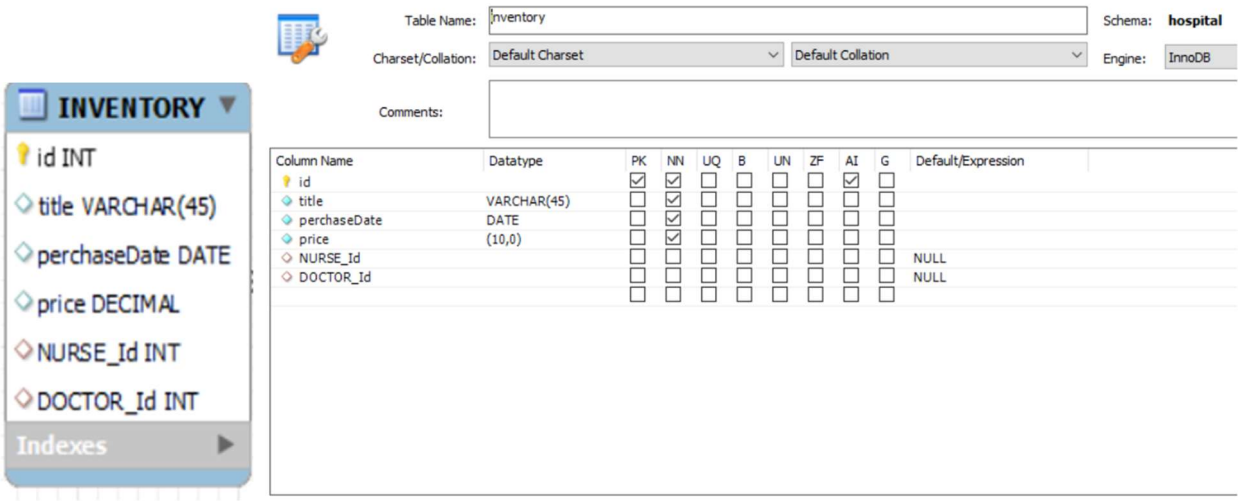
It is logical that patient table will have relationships with nurse and doctor table. However other than that it has relationship with room table. One or more doctor or nurse can take care of the same patient at the time. Also, every patient must have a room while staying in the hospital for treatments.



10 Figure. Patient table's relationships

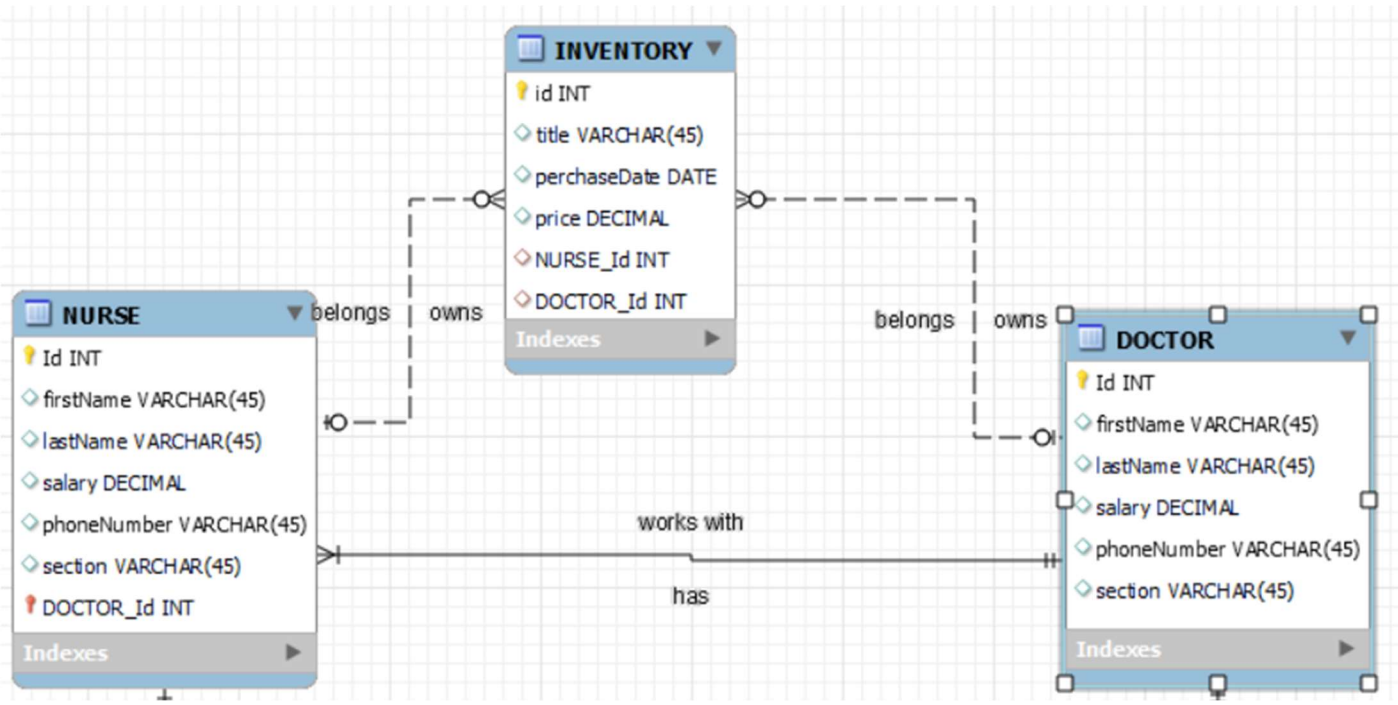
Inventory table:

Every hospital has some items that must be registered. The inventory table is responsible for storing this information. These items can for public usage such chair or patient beds or they could specifically belong to a nurse or doctor. (ID, title, purchase date, price and foreign keys doctor_Id and nurse_Id) are the attributes of inventory table.



11 Figure. Inventory table | 12 Figure. Structure of Inventory table

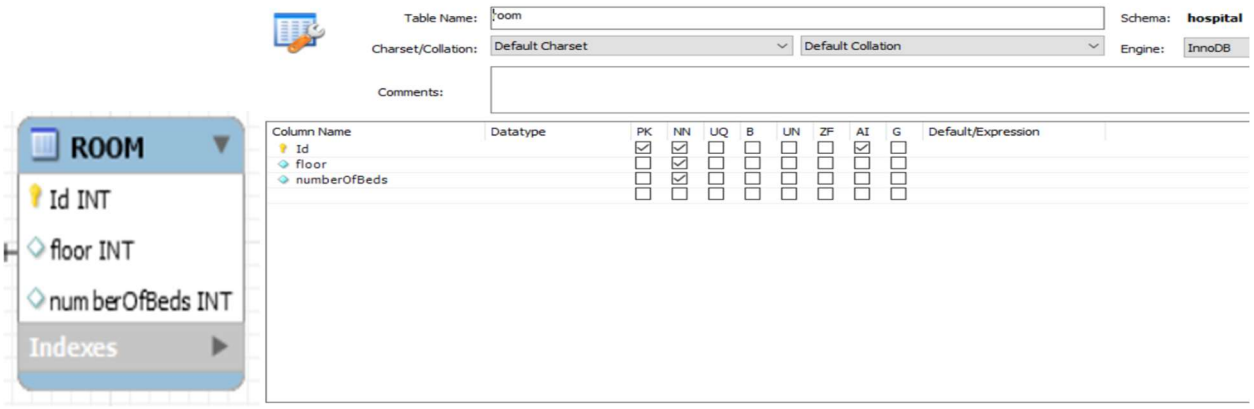
Because inventory can be specific items such as computers or medical instrument and they can be used just by doctor or nurse that they belong to this table has relationship with nurse and doctor table but it is not mandatory.



13 Figure Inventory table's relationships

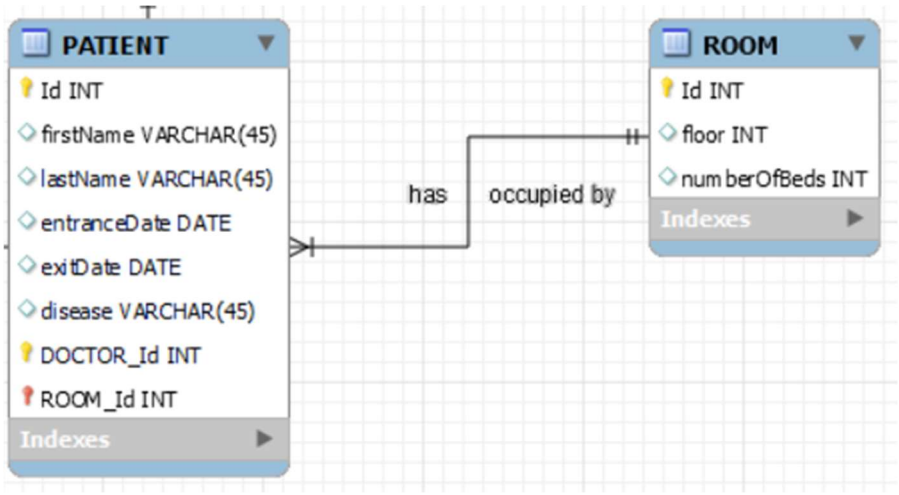
Room table:

Last table of our database is room table which collects data regarding rooms in the hospital. This is a small table so it just has (ID, floor and number of beds) attributes.



14 Figure. Room table | 15 Figure. Structure of room table

This table just have relationship with patient table because patient is staying in the hospital room. More than one patient can stay at the same room.

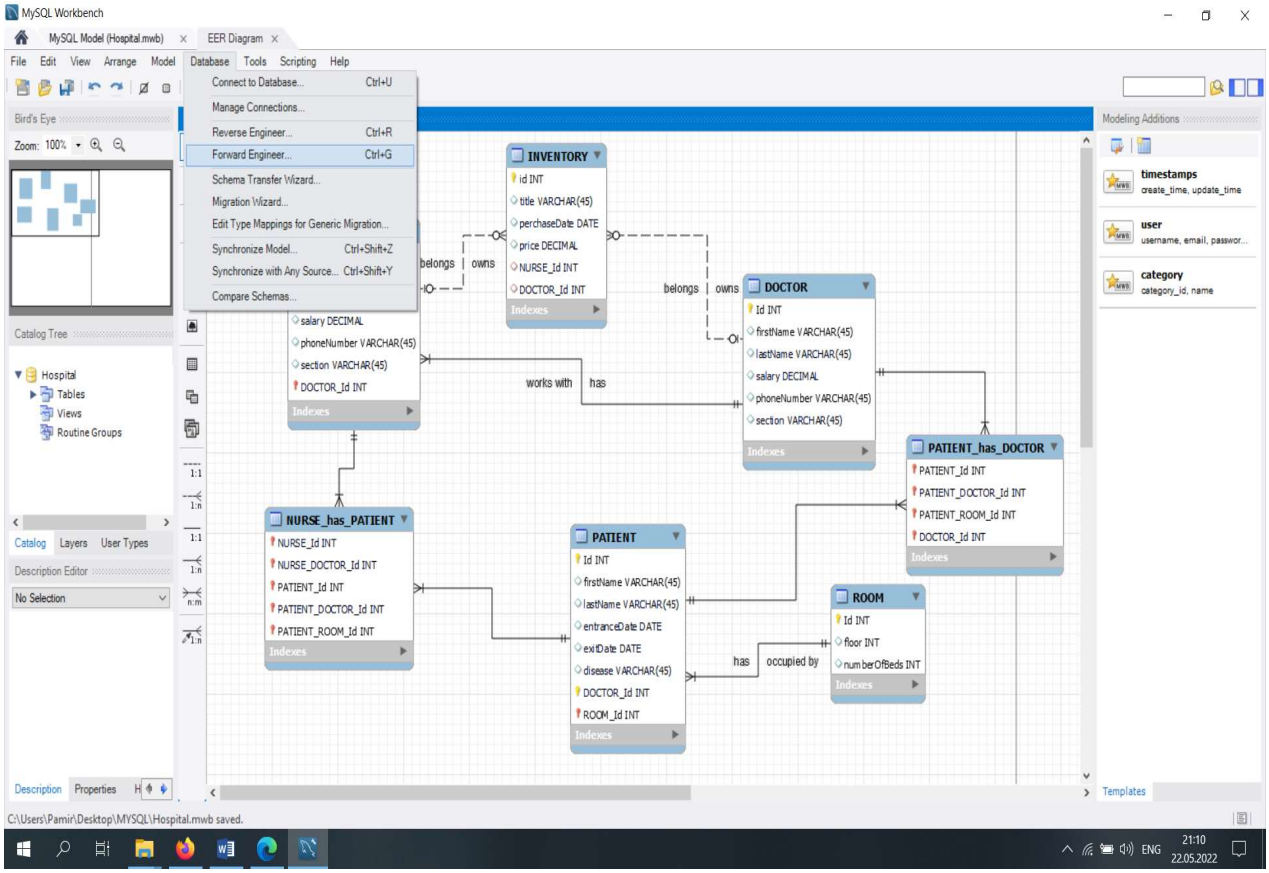


16 Figure room table 's relationships

2.1 Transformation to Physical Model

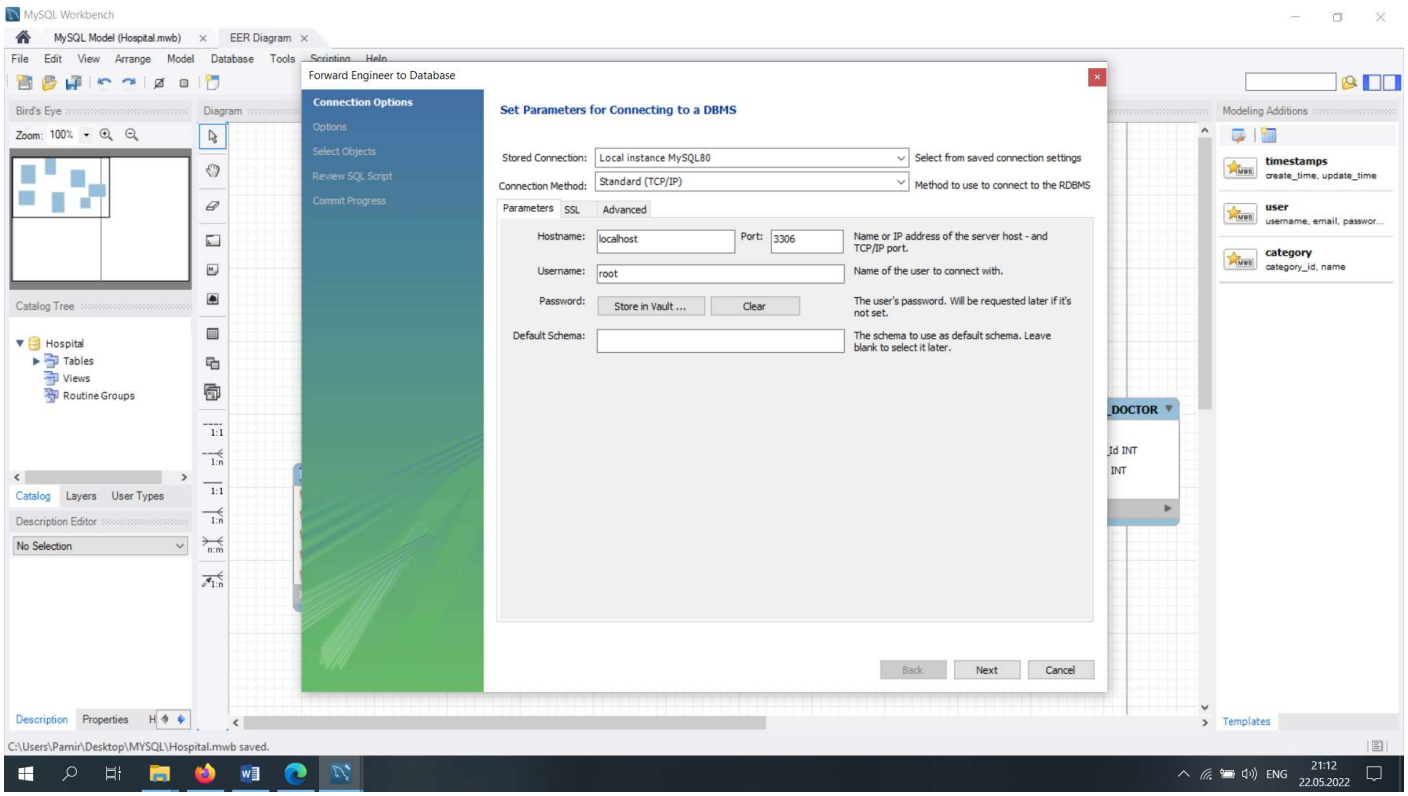
This hospital database logical model was built in MySQL Workbench. Using a DBSM software has a lot of benefits and Workbench give us ability to use its forward engineering tool to convert our logical model to physical model. we will also use forward engineering tool for conversion. Every step will be shown with pictures and a short explanation.

1. Whenever the logical model is ready, go to the menu bar click on Database menu than click on Forward Engineering.



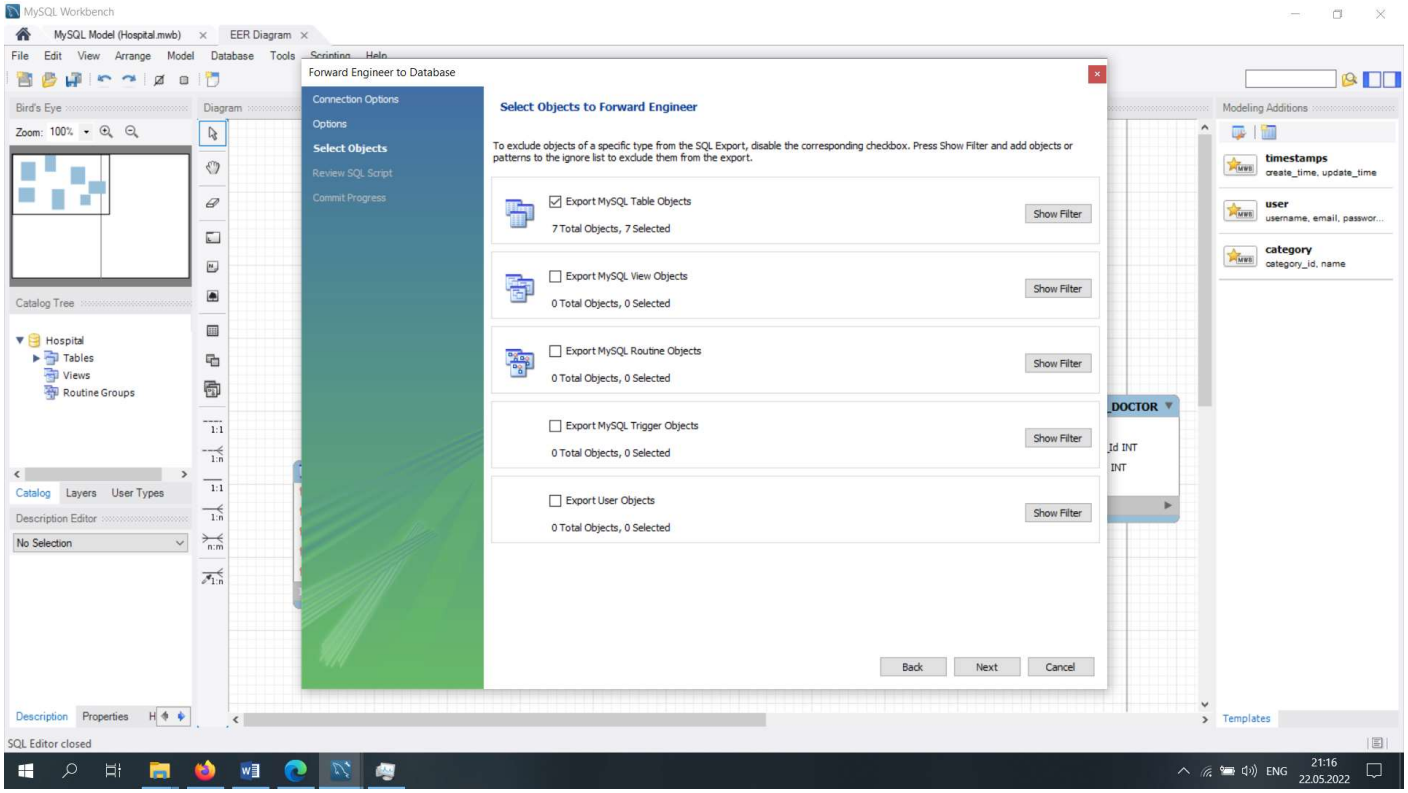
17 Figure. transformation to physical model 1st step

2. A window will be opened showing your host name and port information. You don't have to change anything here just click next.



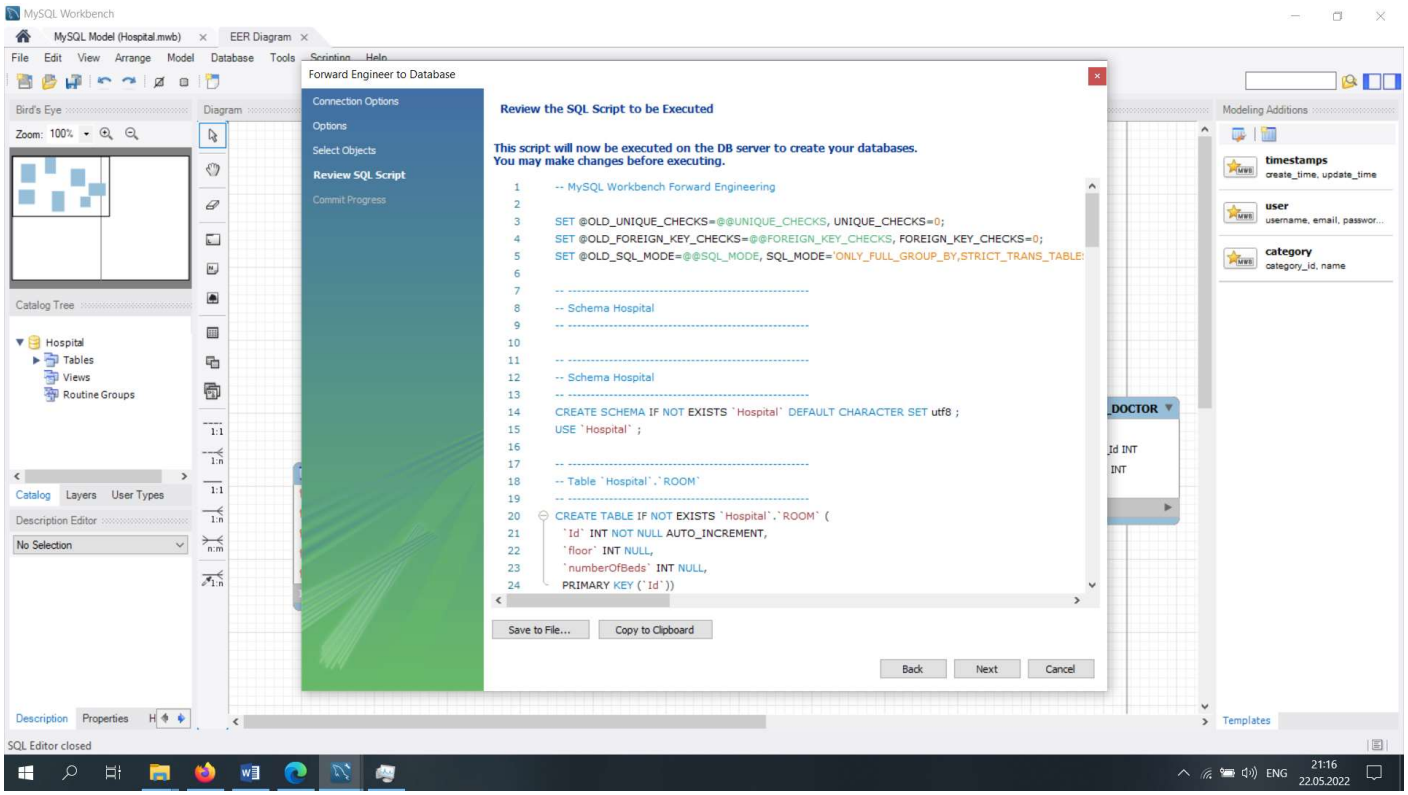
18 Figure. transformation to physical model 2nd step

3. Click next once again on the window
4. Click next once again on the page, shown below.



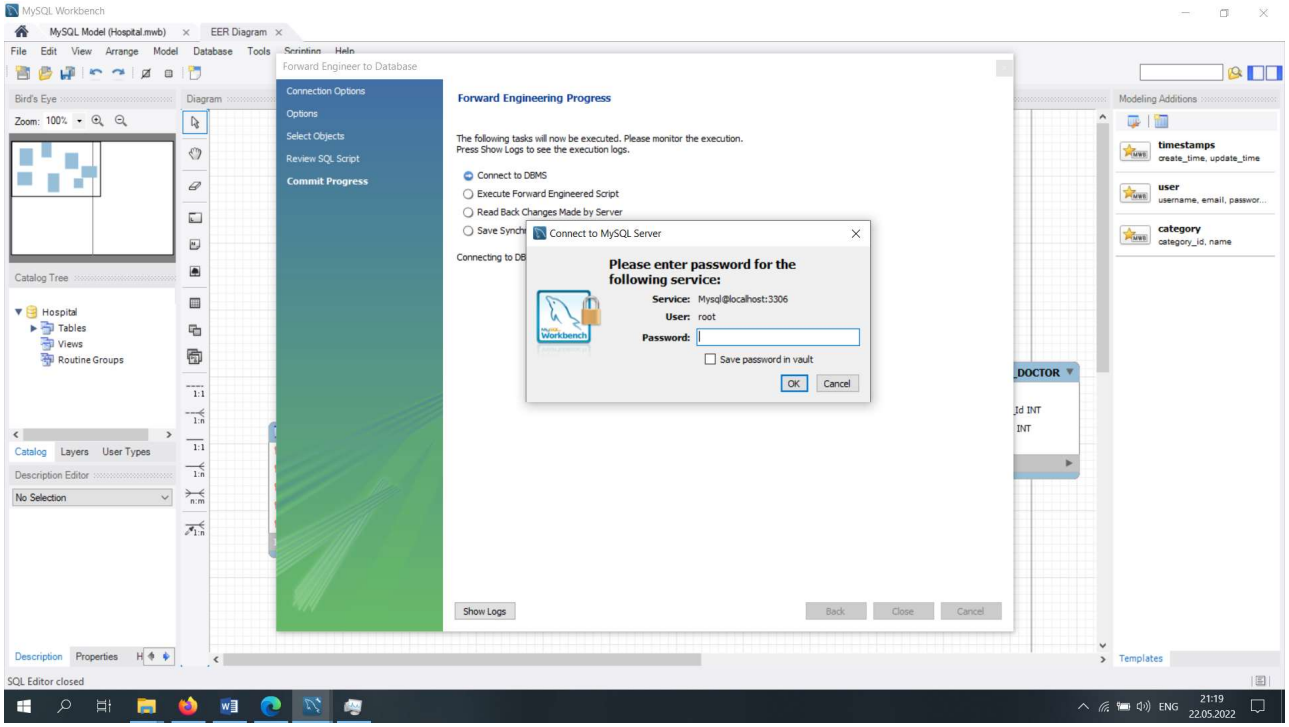
19 Figure. transformation to physical model 4th step

5. On the next page SQL script of the project will be shown. You can check the script to make sure everything is correct once you are satisfied with the script click next.



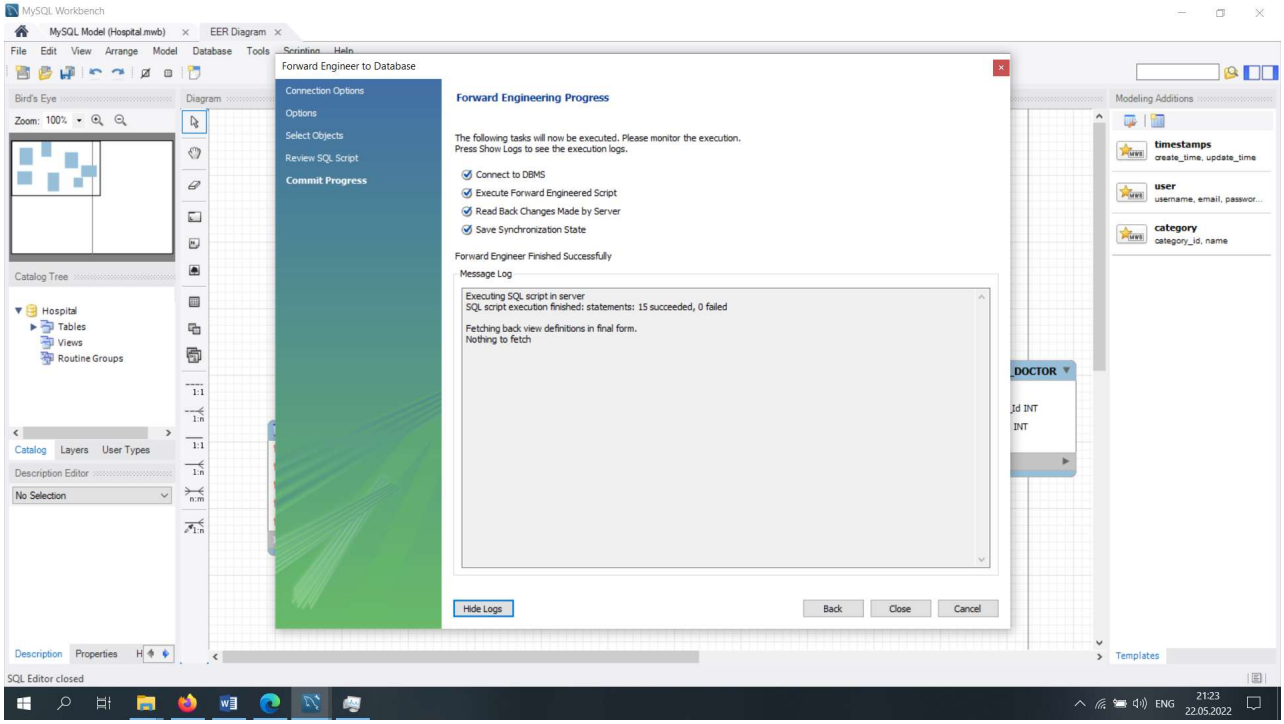
20 Figure. transformation to physical model 5th step

6. After click next on the window shown below you will be asked to enter password. Enter your password for connection to local host and click ok.



21 Figure. transformation to physical model 6th step

7. This is the last step the software will convert your model to physical database. If there will be any problem the software will give you an error. You can click the 'Show logs' button to see more about the error. If there are no error just click close. The database is now ready go and check your database in scheme's list.

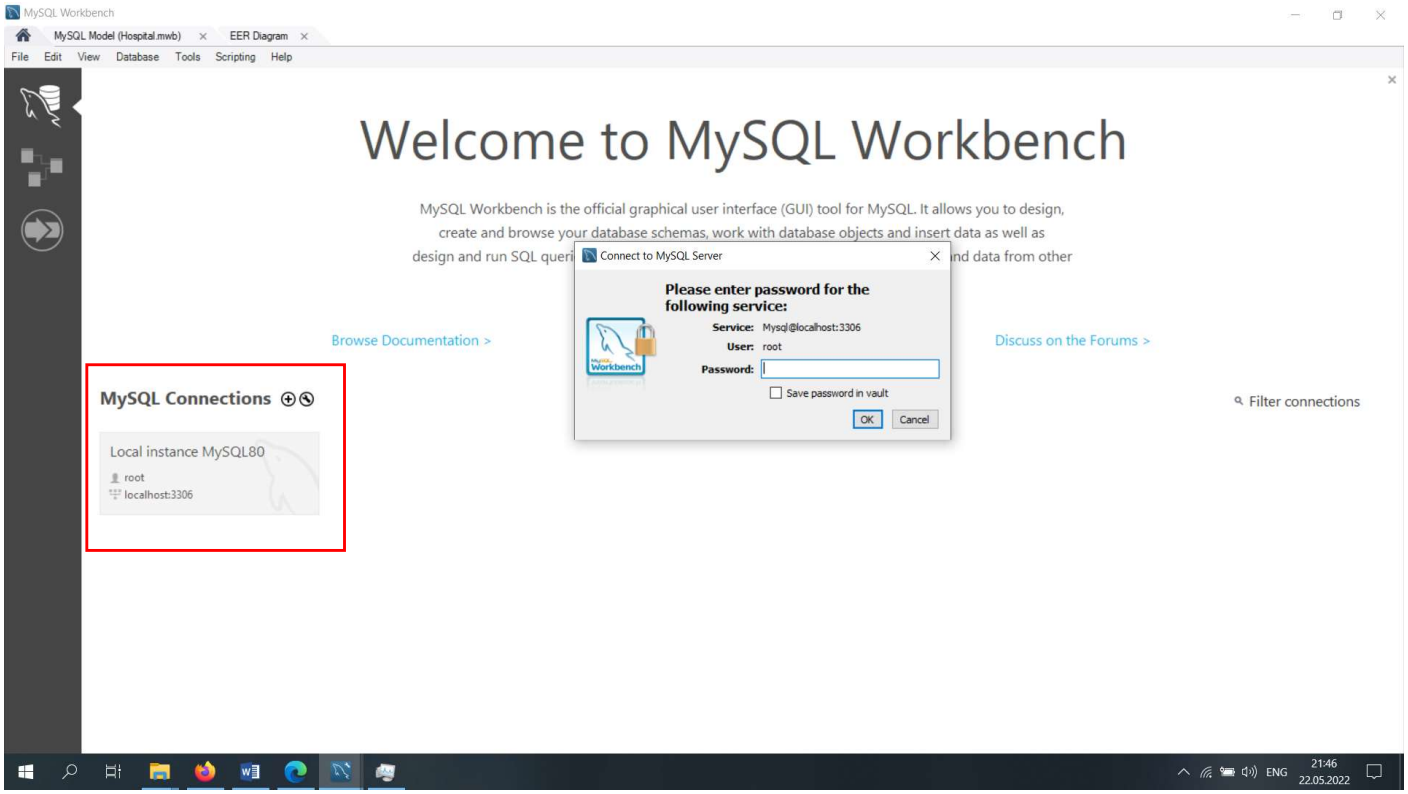


22 Figure. transformation to physical model 7th step

3 Join to Server

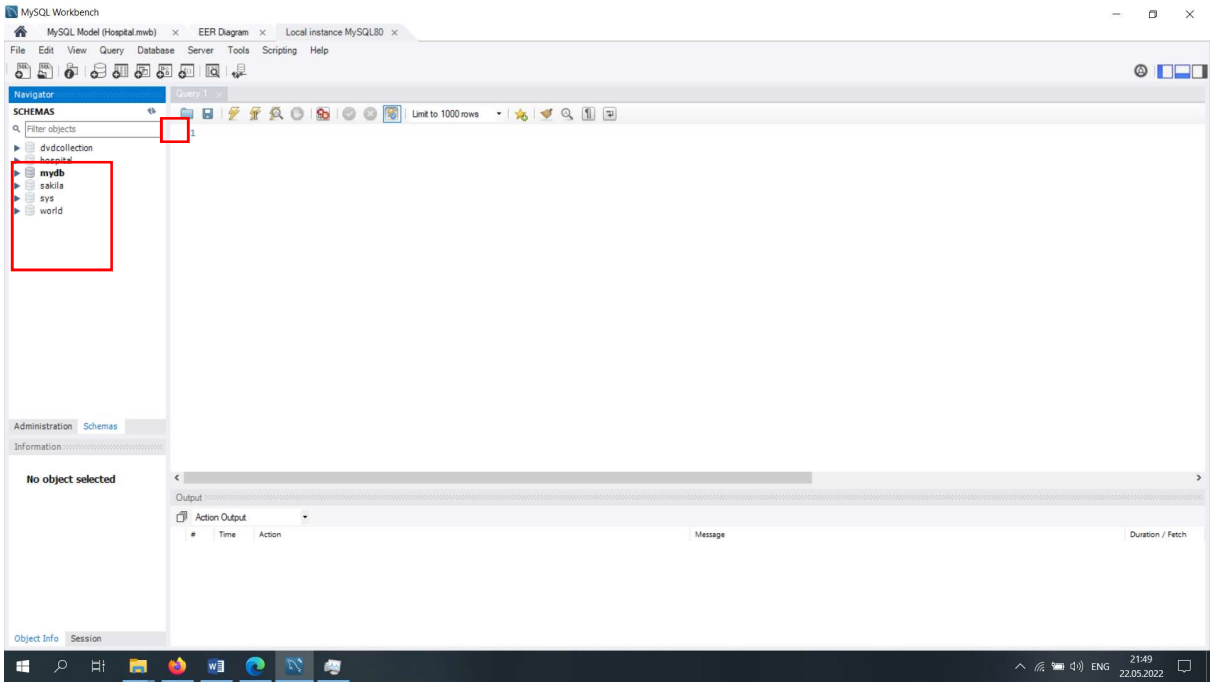
In last section we have created our database from the logical model in MySQL Workbench software. In this section we will connect to server check our database if everything was created correctly and at end some dummy data will be added to the tables to check the functionality of the database.

In the first step MySQL Workbench program should be opened then click on the “Local instance Mysql80” at the bottom of the window. After clicking a small window will pop up and ask for password. Enter your password and click OK.



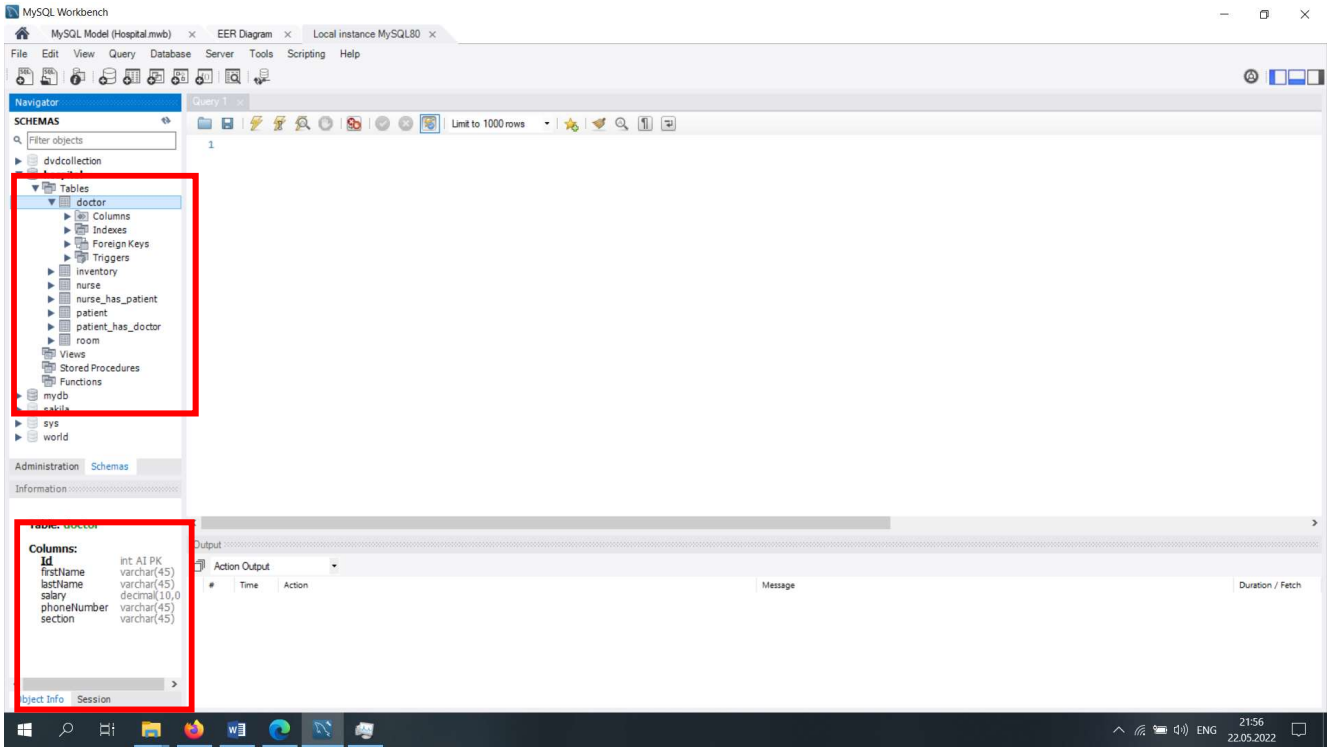
23 Figure. Join to server 1st step

A new window will be opened showing a page where you can write queries and at the left side of the page you can find all of your database. If your database is not shown yet you have to click the “refresh” button at the top right of the database list. You can find visual info in the picture bellow. The button and list will be marked with red rectangle.



24 Figure. Join to server 2nd step

Now double click on the hospital database to select it as default database. Once you have double clicked on the data base all list will be opened bellow the name of the database. You can double click on tables to show the tables of the selected database. Now you should be able to see all tables belongs to this particular database, make sure that every table you have created in logical model is here. You can also click on tables to check which attributes the have. List of the attribute for clicked table will be shown in the left bottom part of the window.



25 Figure. Join to server 3rd step

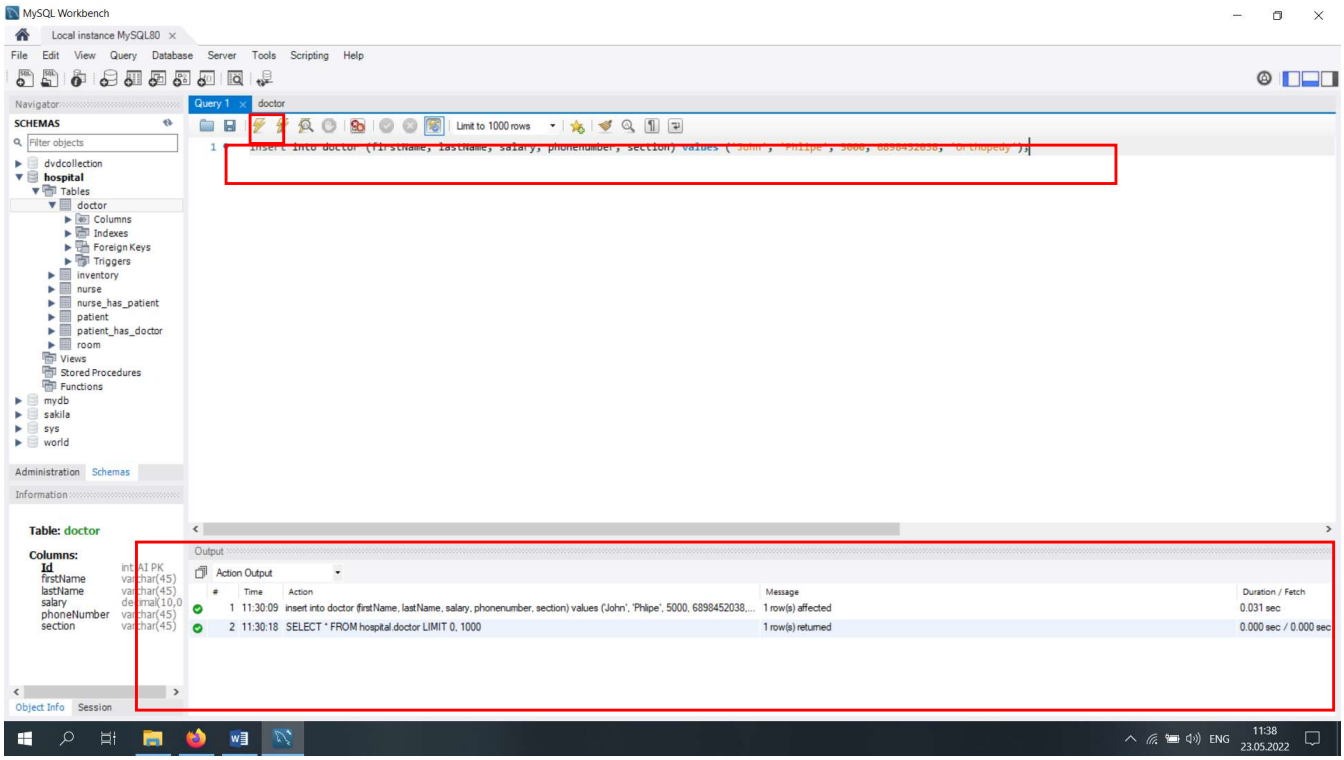
3.1 Filling Tables with Data

While we are using MySQL Workbench there are two different ways to insert data into your table. You can choose one either way and the result will absolutely the same. However before starting to fill the table we have to pay attention in which order we need to fill our table because they have relationships between each other and some records needs to exist before showing it as a foreign key. In this particular database we have to start filling tables in given order: doctor, nurse, inventory, room, patient, nurse_has_patient and finally the table doctor_has_patient.

- The first way is to write an insert query for a specific table and enter the values you want to add to the table. The query will be given below with a picture showing where and how to write the query.

```
insert into doctor (firstName, lastName, salary, phonenumber, section) values ('John', 'Phlipe', 5000, 6898452038, 'Orthopedy');
```

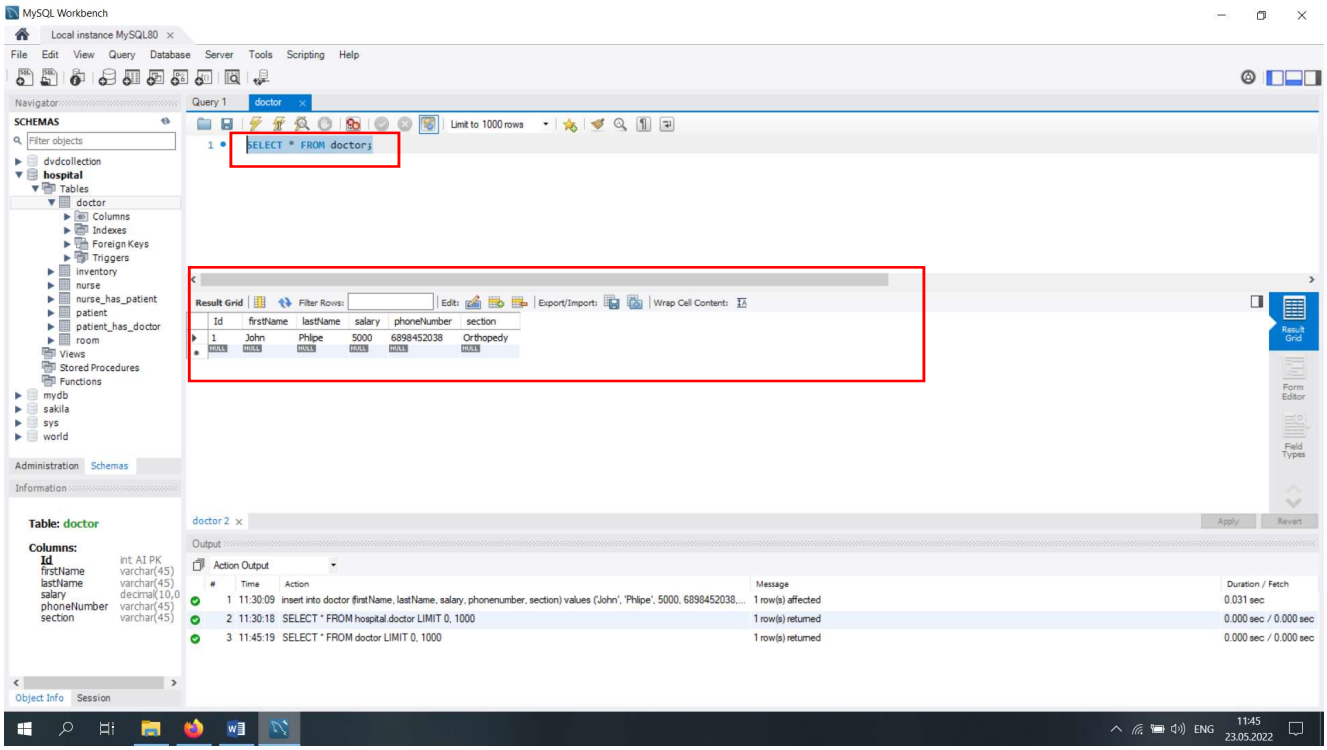
once you have written the query you can press the “Execute” button which has a thunder sign to run the query. It is good idea the check the logs bar after executing the query to see if it is executed successfully. The log bar or output section will be in the bottom of the window.



26 Figure. Filling data 1st step

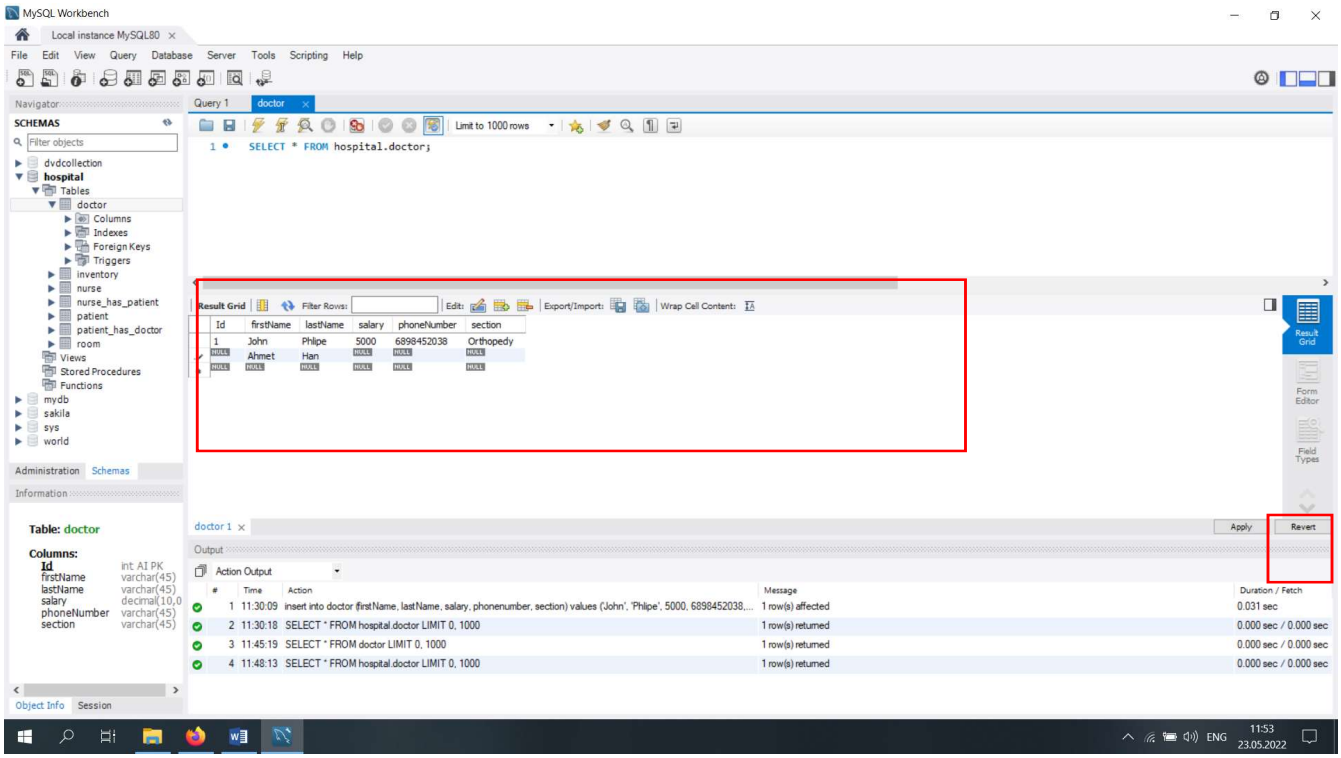
Now you can write a select query to inspect the data which was just added to the table. Query is given below. The result will be also shown in the picture below. The query and the result will be surrounded by red rectangles.

SELECT * FROM doctor;



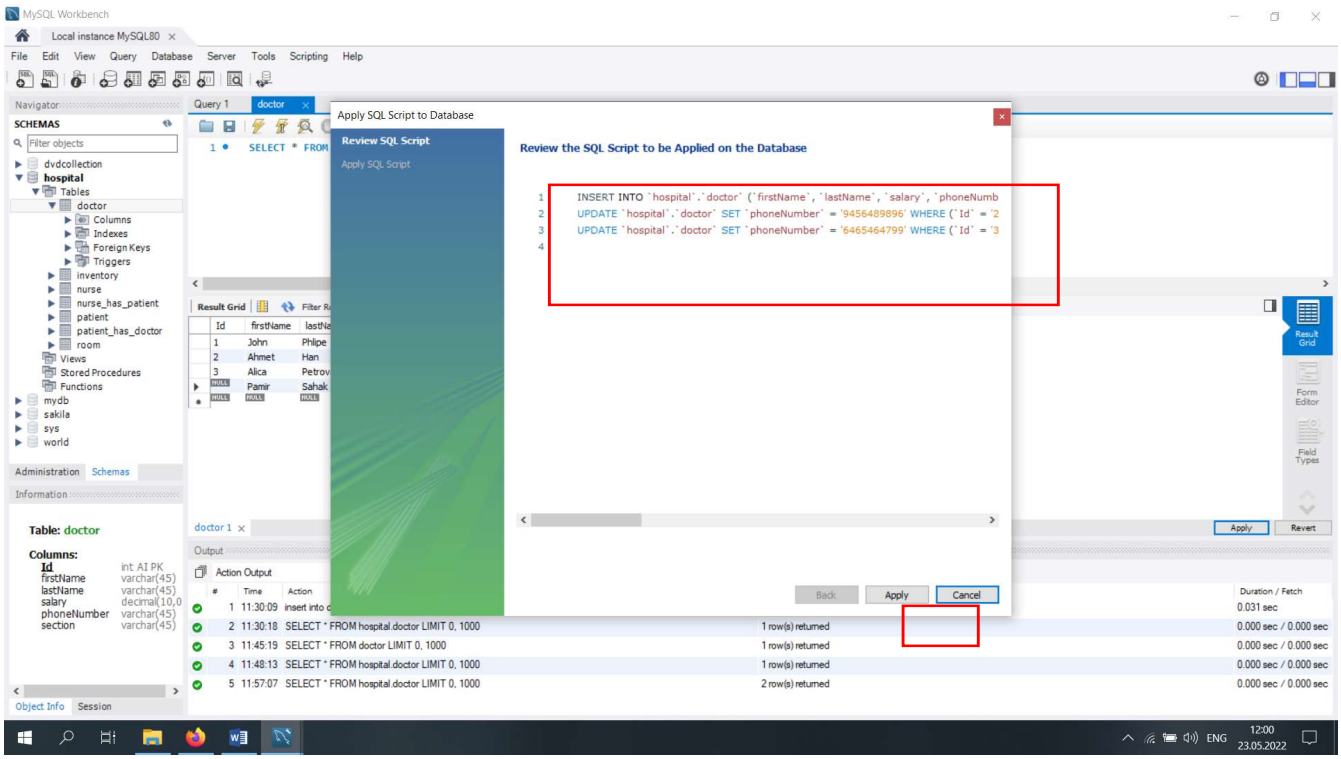
27 Figure. Filling data 2nd step

- The second way of inserting data to a table is simpler and more based on visual tools. First you have to write a simple select query to see all the data in specific table than go to the bottom of the table where the data of the last record will all “NULL”. Double click on each column you want to insert data and enter your data.



28 Figure. Filling data 3rd step

When you entered all the data you needed just simply click on “Apply” button in the right bottom of the window. Clicking apply is just same like clicking execute button. After clicking the apply button a window will pop up with some SQL query. You can see now that in the background the program actually written a query. Check if everything is correct than click apply to execute the query. Finally click the finish button.



29 Figure. Filling data, 4th step

4 Demonstration

This is the last but one of important step in designing database. The designed database is all ready to be used but before implementing the database to the system it is necessary to write some real-life scenario queries and check for correctness of the result. In this section we will start with writing some basic queries and continue towards advanced queries using MySQL functions and joins.

- This is the first query the aim is to see if the data exist in the table

```
9 •  SELECT * FROM hospital.patient;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Cont

Id	firstName	lastName	entranceDate	exitDate	disease	ROOM_Id
1	Pauline	Sellers	2005-08-05	2005-09-01	Kidney Sergury	1
2	Shaunna	Potter	2007-08-12	2007-09-25	Allergy	2
3	Alexandru	Munro	2015-03-15	2016-08-15	Colds and Flu	3
4	Danial	Fuentes	2016-08-21	2016-09-19	Pink eye	6
5	Ivie	Battle	2019-09-20	NULL	Diarrhea	1
6	Nazim	Chamberlain	2022-05-23	NULL	Headaches	2
7	Marianne	Snderson	2022-04-30	NULL	Stocmach aches	4
8	Dean	Chadwick	2022-02-15	2022-03-21	AIDS	6
9	Emyr	Flynn	2021-10-13	2021-12-25	Alzheimer	3
10	Socha	Hart	2019-11-19	2019-12-12	Obesity	13
11	Luca	Kaye	2019-12-30	NULL	Oral Cancer	21
12	Rachel	Francis	2022-05-12	2022-05-22	Malaria	6
13	Alayah	Hampton	2022-05-23	NULL	Fever	7
14	Dion	Quinn	2022-05-15	NULL	Bird Flu	8
15	Katharine	Deacon	2021-09-20	2021-09-25	Black Lung	9
16	Abdullah	Jacobs	2019-07-25	2019-09-10	Hemopjila	7
17	Mindly	Vinson	2022-01-31	NULL	Hendra Virus	14
18	Sneha	Craig	2009-05-09	2009-06-20	Hepatitis A	17
19	Max	Monroe	2006-12-23	2007-01-25	Influenza	20
20	Jobe	Steele	2011-09-05	2012-09-02	Bird Flu	16

30 Figure. Select query 's result

- We know that we have 23 patients registered in our patient table. Now we will use count fuction to see if it is correct.

```
9 •  SELECT COUNT(*) FROM hospital.patient;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

COUNT(*)
23

31 Figure. Using count function

- In this example we will try to find how many days each patient stays in the hospital. Of course, we have to only select the patient how has exit date value, beside numbers of the stay days, id, name and last name of the patient will be also shown.

```
8 •  SELECT id, firstName, lastName, datediff(exitDate, entranceDate) as stay_days
9    FROM hospital.patient
10   where exitdate is not null;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	firstName	lastName	stay_days
1	Pauline	Sellers	27
2	Shaunna	Potter	44
3	Alexandru	Munro	519
4	Danial	Fuentes	29
8	Dean	Chadwick	34
9	Emyr	Flynn	73
10	Socha	Hart	23
12	Rachel	Francis	10
15	Katharine	Deacon	5
16	Abdullah	Jacobs	47
18	Sneha	Craig	42
19	Max	Monroe	33
20	Jobe	Steele	363
22	Darsh	Deacon	389
23	Aman	Battle	12

32 Figure. Using datediff function

- Here group by statement will be used with count function to find the number of patients staying in each room. We will just show the room id and number of patient's column.

```
8 •  select ROOM_Id, count(*) as number_of_patients
9    from patient group by ROOM_Id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ROOM_Id	number_of_patients
1	2
2	3
3	3
4	2
6	3
7	2
8	1
9	1
13	1
14	1
16	1
17	1
20	1
21	1

33 Figure. Using group by function

- This is one of the advanced examples using MySQL joins. In this query we have to join three table which are doctor, patient and patient_has_doctor the goal is to find name, surname and room number of each patient belongs to a particular doctor. Doctor ID, name and surname will be also shown.

```
1  select doctor.id as doctor_id, doctor.firstName as doctor_name, doctor.lastname as doctror_surname,
2  patient.firstName as patient_name, patient.lastName as patient_surname, patient.ROOM_Id as room
3  from ((doctor_has_patient join doctor on doctor.id = doctor_has_patient.DOCTOR_Id)
4  join patient on patient.Id = doctor_has_patient.PATIENT_Id);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

doctor_id	doctor_name	doctror_surname	patient_name	patient_surname	room
1	John	Phlipe	Danial	Fuentes	6
1	John	Phlipe	Tolga	Sanderson	2
2	Ahmet	Han	Emyr	Flynn	3
2	Ahmet	Han	Luca	Kaye	21
3	Alica	Petrova	Katharine	Deacon	9
3	Alica	Petrova	Max	Monroe	20
4	Pamir	Sahak	Abdullah	Jacobs	7
4	Pamir	Sahak	Mindly	Vinson	14
5	Declan	Cline	Darsh	Deacon	3
6	Yusuf	Proctor	Shaunna	Potter	2
7	Lucy	Santiago	Socha	Hart	13
8	Elise	Werner	Ivie	Battle	1
8	Elise	Werner	Marianne	Snderson	4
9	Jennifer	Johnson	Alexandru	Munro	3
9	Jennifer	Johnson	Sneha	Craig	17
10	Paula	Reyes	Alayah	Hampton	7
10	Paula	Reyes	Jobe	Steele	16
11	Elijah	Fisher	Aman	Battle	4
12	Patrica	Himilton	Nazim	Chamberlain	2
12	Patrica	Himilton	Rachel	Francis	6
13	Samuel	Patterson	Pauline	Sellers	1
13	Samuel	Patterson	Dean	Chadwick	6

34 Figure. Using 3 joins in tables

- This is another example of using joins, here we try to find all inventories which belongs to nurses of the hospital with the name, surname, id and section information from the nurse table and title, purchase date information form inventory table.

```
4 • select nurse.id, nurse.firstName, nurse.lastName, nurse.section,
5   inventory.title, inventory.perchaseDate
6   from nurse join inventory on nurse.id = inventory.NURSE_Id
7   where inventory.NURSE_Id is not null;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	firstName	lastName	section	title	perchaseDate
3	Irvin	Obrien	Radiology	computer	2010-05-21
9	Francis	Chambers	Ophthalmology	chair	2022-02-22
10	Thelma	Nash	Pediatrics	Stethoscope	2021-07-25

Result 5 ×

35 Figure. Using inner join

- Now we will try to find which doctor each nurse belongs to. We will use again the MySQL joins to retrieve data from both nurse and doctor table. We will show information such as id, first name, last name for nurse and id, first name, last name and section information for the doctor.

```
 9 •  select nurse.id as nurse_id, nurse.firstName, nurse.lastName,
10    doctor.id as doctor_id, doctor.firstName as doctor_name,
11    doctor.lastName as doctor_surname, doctor.section
12    from nurse join doctor on nurse.DOCTOR_Id = doctor.id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nurse_id	firstName	lastName	doctor_id	doctor_name	doctor_surname	section
5	Sonja	Carroll	1	John	Phlipe	Orthopedy
3	Irvin	Obrien	2	Ahmet	Han	Neurology
7	Morris	Edwards	3	Alica	Petrova	Urology
8	Marguerite	Massey	4	Pamir	Sahak	Pediatrics
1	Margie	Bass	5	Declan	Cline	Genetics
4	Tina	Oslon	7	Lucy	Santiago	Internal
2	Olga	Harper	8	Elise	Werner	Dermatology
9	Francis	Chambers	9	Jennifer	Johnson	Ophthalmology
10	Thelma	Nash	10	Paula	Reyes	Pathology
6	Tiffany	Zimmerman	11	Elijah	Fisher	Surgery

36 Figure. Using join 2nd example

- In this last example we will join four join four table and we will get data from patient, doctor, nurse and nurse_has_patient tables. Information such as patient id, name, surname, disease, room id, nurse name and last name and doctor name and last name will be shown.

```
 9 •  select patient.id as patient_id, patient.firstName as pateint_name, patient.lastName as patient_surname,
10    patient.disease, patient.ROOM_Id, nurse.firstName as nurse_name, nurse.lastName as nurse_surname,
11    doctor.firstName as doctor_name, doctor.lastName as doctor_surname, doctor.section
12    from (((patient join nurse_has_patient on patient.id = nurse_has_patient.PATIENT_Id)
13    join nurse on nurse.id = nurse_has_patient.nurse_id)
14    join doctor on doctor.id = nurse_has_patient.NURSE_DOCTOR_Id);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

patient_id	pateint_name	patient_surname	disease	ROOM_Id	nurse_name	nurse_surname	doctor_name	doctor_surname	section
1	Pauline	Sellers	Kidney Sergury	1	Irvin	Obrien	Ahmet	Han	Neurology
2	Shaunna	Potter	Allergy	2	Marguerite	Massey	Pamir	Sahak	Pediatrics
3	Alexandru	Munro	Colds and Flu	3	Francis	Chambers	Jennifer	Johnson	Ophthalmology
4	Danial	Fuentes	Pink eye	6	Thelma	Nash	Paula	Reyes	Pathology
5	Ivie	Battle	Diarrhea	1	Sonja	Carroll	John	Phlipe	Orthopedy
6	Nazim	Chamberlain	Headaches	2	Tina	Oslon	Lucy	Santiago	Internal
7	Marianne	Snderson	Stocmach aches	4	Sonja	Carroll	John	Phlipe	Orthopedy
8	Dean	Chadwick	AIDS	6	Tiffany	Zimmerman	Elijah	Fisher	Surgery
9	Emyr	Flynn	Alzheimer	3	Margie	Bass	Declan	Cline	Genetics
10	Socha	Hart	Obesity	13	Irvin	Obrien	Ahmet	Han	Neurology
11	Luca	Kaye	Oral Cancer	21	Tina	Oslon	Lucy	Santiago	Internal
12	Rachel	Francis	Malaria	6	Morris	Edwards	Alica	Petrova	Urology
13	Alayah	Hampton	Fever	7	Marguerite	Massey	Pamir	Sahak	Pediatrics

36 Figure. Using several joins

Conclusion and recommendations

1- The main idea in this project was to build a database that the management of the hospital will be able to store information about their staff, patient and inventory, by inspecting the designed database we are able to see that the main goal is achieved. Because the necessary tables are designed and linked logically with each other. We must not forget that the database has ability to be updated or expanded later if it is necessary.

2- In the physical model section of this project, all the created tables are shown. They are doctor, nurse, patient, inventory and room, you will also notice two extra tables doctor_has_patient and nurse_has_patient which are created because of the relationship between nurse, doctor and patient table.

3- We started this database project from the scratch and build it step by step to the end. First of all, we wrote the business requirements for the domain and the next step was designing our logical model. when the logical model was ready it was time to transform logical model to physical model. After that we entered to MySQL Workbench to check our designed database and fill its tables with data. The required tools to build this database is not a lot. You will need an operating system which has a Rational Database System Management software however advance knowledge about the MySQL language is a must.

4- The designed data is filled with dummy (random) data so we can check the performance and functionality of our database. Data were inserted to tables by specific order.

5- In the conclusion every specified task was completed and its completion was supported with evidences such as pictures, demo or explanation. In the last section we used MySQL functions and joins to query for demonstration purpose. The Hospital database is all ready to be implemented to the system and start using.