# Class Project

Group members:
Danay Fernandez Alfonso
Nick Zelada
Maurice Ngouen

Florida International University

COP6727: Advanced Database Systems

November 25th, 2020

## 1. Design

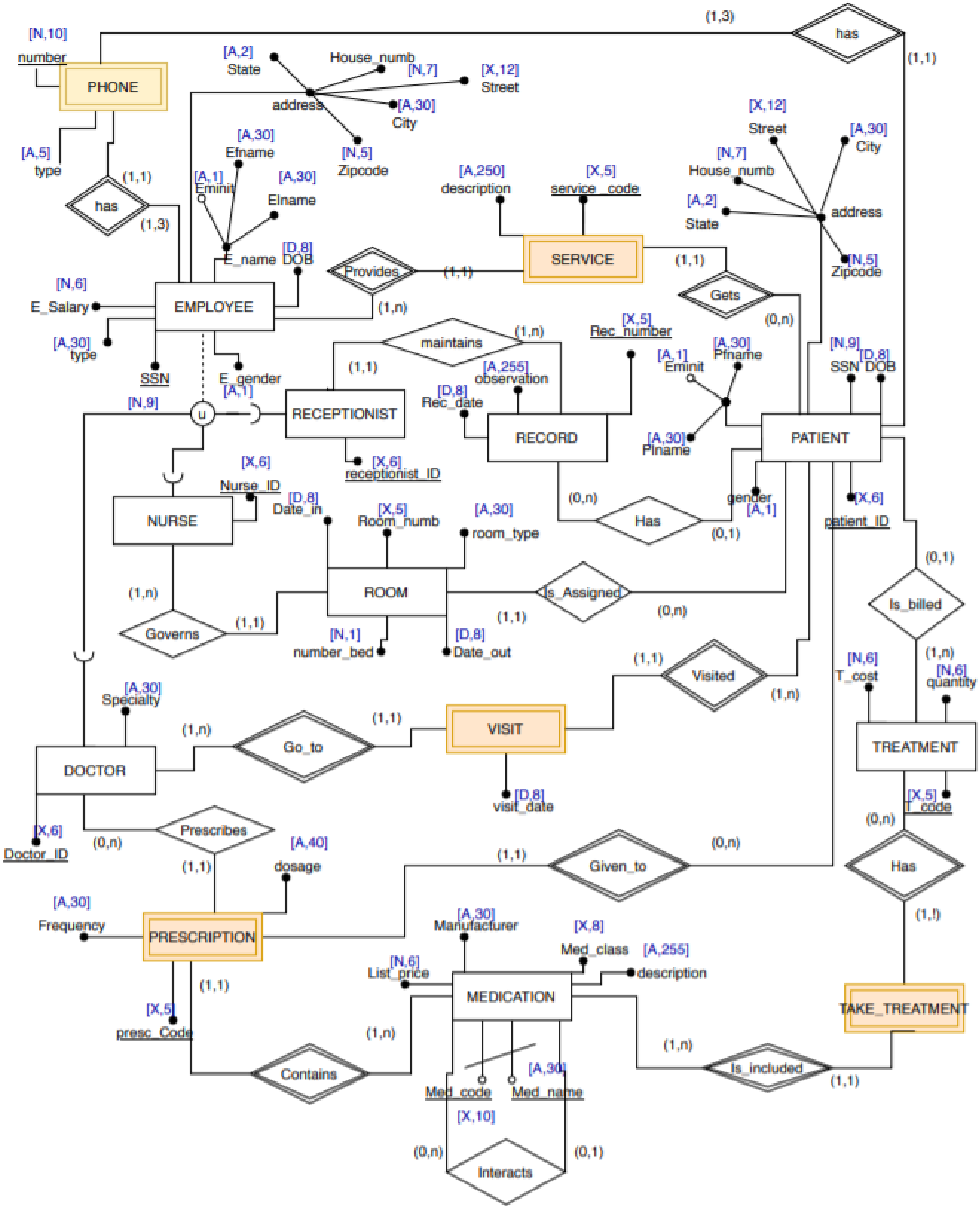
*a) Explain what type of database architecture you recommend this type of database. For example, a centralized database, a distributed database, etc and why.*

For this type of database we recommend a relational distributed database. A distributed database would be good because it allows for fast access in databases spreaded across different physical locations which would be the case for a hospital system. Additionally, a distributed database system will make sure that the system continues working even if a piece of it is not.

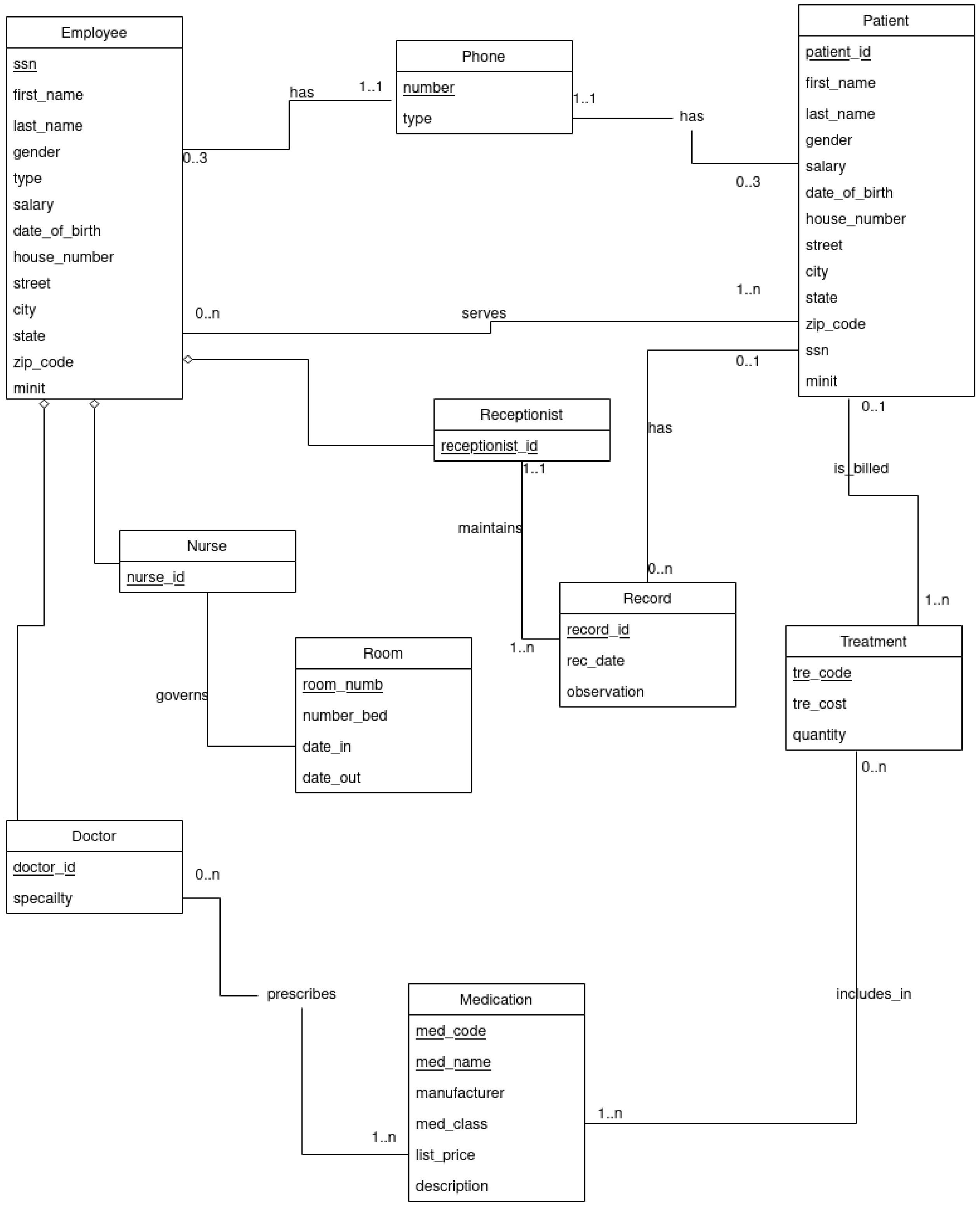
*b) Create a conceptual design for the hospital system database by using an ER diagram (10 points). Your conceptual design of the database should include the followings but not limit to:*
*• Entities*
*• Relationships*
*• Keys*
*• Structural constraints (Cardinality ratio)*

Additional requirements:

- A patient's age can be obtained as a derived attribute from date of birth (DBO)
- This design keeps track of a patient's date admitted (Date_in) and date discharged (Date_out) in the Room table since each current patient is assigned to a room
- In the Room table, the attribute Room_type is being used to keep track of whether or not a patient was admitted via ER. Therefore, Number_bed is being used to indicate if the room has a single bed (1) or two beds (2)
- The design also includes manufacturer and class in the Medication table
- The design includes a Service table which keeps track of when a doctor visits a patient and the procedure they do
- Phone table which keeps track of employees' and patients' phone numbers
- Take_treatment keeps track of the treatment each patient is taking at a given time
- Prescription table keeps track of the prescription given to a patient by a doctor


(1,3)
has
(1,1)
[N,10]
number
PHONE
[A,2]
State
House_numb
[N,7]
[X,12]
Street
address
[A,30]
City
[N,5]
Zipcode
[A,5]
type
has
(1,1)
(1,3)
[A,30]
Efname
[A,1]
Eminit
[A,30]
Elname
E_name
[D,8]
DOB
Provides
(1,1)
SERVICE
[A,250]
description
[X,5]
service _code
(1,1)
Gets
(0,n)
[X,12]
Street
[A,30]
City
[N,7]
House_numb
[A,2]
State
address
[N,5]
Zipcode
[N,6]
E_Salary
EMPLOYEE
(1,n)
[A,30]
type
SSN
E_gender
[A,1]
maintains
(1,n)
(1,1)
[X,5]
Rec_number
[A,255]
observation
[D,8]
Rec_date
RECORD
[A,1]
Eminit
[A,30]
Pfname
[A,30]
Plname
[N,9]
SSN
[D,8]
DOB
PATIENT
[N,9]
u
RECEPTIONIST
[X,6]
receptionist_ID
[X,6]
Nurse_ID
NURSE
[D,8]
Date_in
[X,5]
Room_numb
[A,30]
room_type
(0,n)
Has
(0,1)
gender
[A,1]
[X,6]
patient_ID
(0,1)
Is_billed
(1,n)
Governs
(1,n)
(1,1)
ROOM
Is_Assigned
(1,1)
(0,n)
[N,1]
number_bed
[D,8]
Date_out
(1,1)
Visited
(1,n)
[N,6]
T_cost
[N,6]
quantity
TREATMENT
[X,5]
T_code
[A,30]
Specialty
(1,n)
Go_to
(1,1)
VISIT
DOCTOR
[D,8]
visit_date
[X,6]
Doctor_ID
(0,n)
Prescribes
(1,1)
[A,40]
dosage
(1,1)
Given_to
(0,n)
(0,n)
Has
(1,!)
[A,30]
Frequency
PRESCRIPTION
[A,30]
Manufacturer
[X,8]
Med_class
[A,255]
description
[N,6]
List_price
MEDICATION
(1,1)
[X,5]
presc_Code
(1,n)
(1,n)
TAKE_TREATMENT
Contains
Is_included
(1,1)
[A,30]
Med_code
Med_name
[X,10]
(0,n)
(0,1)
Interacts

*c) Create a conceptual design for the hospital system database using UML class diagram. (5 points)*



**2. Transform the ER schema of the database you get from step 1 into the corresponding relational database schema.** (10 points)
*a. Specify all the key attributes of relations and any referential integrity constraints.*
*b. Specify the data item format for each attribute in each relation schema.*
*c. Specify all the functional dependencies you could infer from the requirements*

R1: EMPLOYEE (SSN, E_Salary, DOB, E_Gender, E_First_name, E_Last_name, E_mint
House_numb, Street, City,  State, Zip_code, Type)

R2: PATIENT (Patient_ID, SSN, DOB, P_Gender, P_First_name, P_Last_name, p_mint,
house_numb, Street, City,  State, Zip_code)

R3: SERVICE ( Service _code, Description,  **Ser_patient_ID**, **Ser_SSN**,)
# SERVICE.{Ser_patient_ID} ⊆ PATIENT.Patient_ID}
AND
# SERVICE.{Ser_SSN} ⊆ EMPLOYEE.{SSN}

R4: RECEPTIONIST ( Receptionist_ID,  **Rec_SSN**)
#  RECEPTIONIST.{Rec_SSN} ⊆ EMPLOYEE.{SSN}

R5: RECORD ( Recp_number, Rec_date, Observation,  **Rec_receptionist_ID**,
**Rec_patient_ID**)
#  RECORD.{Rec_receptionist_ID} ⊆  RECEPTIONIST.{Receptionist_ID}
AND
#  RECORD.{Rec_patient_ID} ⊆ PATIENT.{Patient_ID}

R6: NURSE ( Nurse_ID, **Nur_SSN**)
#  NURSE.{Nur_SSN} ⊆ EMPLOYEE.{SSN}

R7: ROOM Room_numb, Numb_Bed, Date_in, Date_out, Room_type, **Roo_nurse_ID**,
**Roo_patient_ID**)
#  ROOM.{Roo_nurse_ID} ⊆ NURSE.{Nurse_ID}
AND
# ROOM.{Roo_patient_ID} ⊆ PATIENT.{Patient_ID}

R8: DOCTOR (Doctor_ID, **Doc_SSN**, Specialty)
#   DOCTOR.{Doc_SSN} ⊆ EMPLOYEE.{SSN}

R9: VISIT ( **Vis_patient_ID**, **Vis_doctor_ID**, Visit_date )
#   VISIT.{Vis_patient_ID} ⊆ PATIENT.{Patient_ID}
AND
#  VISIT.{Vis_doctor_ID} ⊆ DOCTOR.{Doctor_ID}

R10: MEDICATION (Med_code, Med_name, List_price, Manufacturer, Class,   Description)

R11: PRESCRIPTION ( Presc_Code, Frequency, Dosage, **Pre_med_code**, **Pre_med_name**, **Pre_patient_ID**, **Pre_doctor_ID**)

# PRESCRIPTION.{Pre_med_code} ⊆ MEDICATION.{Med_code, Med_name,}

AND

# PRESCRIPTION.{Pre_patient_ID} ⊆ PATIENT.{Patient_ID}

AND

# PRESCRIPTION.{Pre_doctor_ID} ⊆ DOCTOR.{Doctor_ID}

OR

# PRESCRIPTION.{Pre_med_name} ⊆ MEDICATION.{Med_name,}

AND

# PRESCRIPTION.{Pre_patient_ID} ⊆ PATIENT.{Patient_ID}

AND

# PRESCRIPTION.{Pre_doctor_ID} ⊆ DOCTOR.{Doctor_ID}

R12: TREATMENT ( Tre_code, Tre_cost, Quantity, **Tre_patient_ID**)

# TREATMENT.{Tre_patient_ID} ⊆ PATIENT.{Patient_ID}

R13: TAKE_TREATMENT (**Tak_med_code**, **Tak_med_name**, **Tak_trea_code**)

# TAKE_TREATMENT.{Tak_med_code, Tak_med_name} ⊆ MEDICATION.{Med_code, Med_name}

AND

# TAKE_TREATMENT.{Tak_trea_code)} ⊆ TREATMENT.{Treat_code)}

R14: PHONE ( Number, Type, **Pho_patient_ID**, **Pho_ SSN)**

# PHONE.{Pho_patient_ID} ⊆ PATIENT.{Pho_patient_ID}

AND

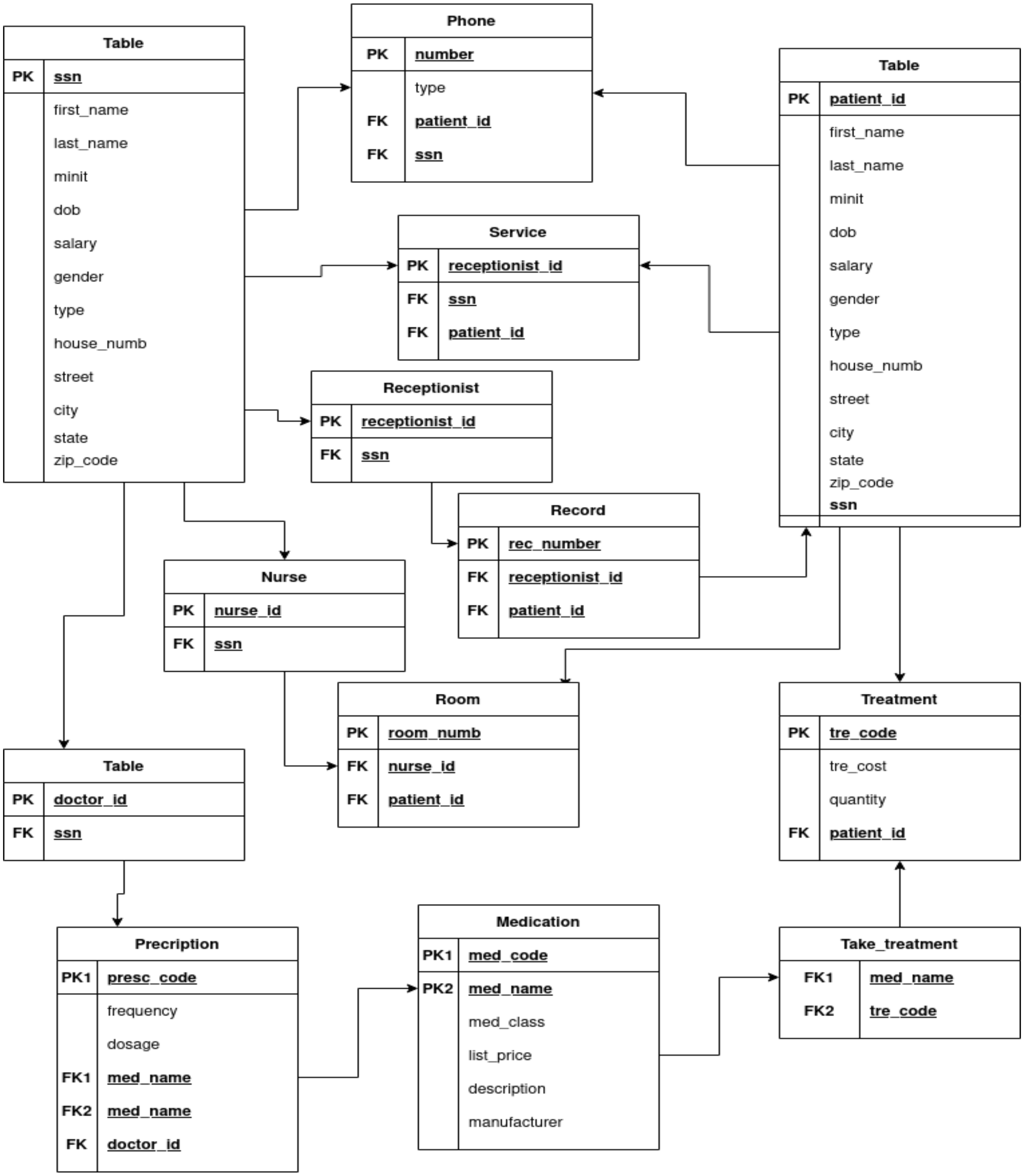
# PHONE.{Pho_ SSN} ⊆ EMPLOYEE.{SSN}

| Attribute | Description | Data type |
|---|---|---|
| SSN | Employee’s and patient’s SSN | Numeric |
| E_Salary | Employee salary | Numeric |
| DOB | Employee’s and patient’s date of birth | Date |
| E_Gender | Employee’s gender | Alphabetic |
| E_First_name | Employee’s first name | Alphabetic |
| E_Last_name | Employee’s last name | Alphabetic |
| E_mint | Employee’s middle initial | Alphabetic |
| Type | Employee’s type | Alphabetic |
| House_numb | Employee’s and patient’s house/apartment number | Alphanumeric |
| Street | Employee’s and patient’s street name | Alphanumeric |
| City | Employee’s and patient’s city | Alphabetic |
| State | Employee’s and patient’s state | Alphabetic |
| Zip_code | Employee’s and patient’s zip code | Numeric |
| Patient_ID | Patient’s unique identifier | Numeric |
| P_Gender | Patient’s gender | Alphabetic |
| P_First_name | Patient’s first name | Alphabetic |
| P_Last_name | Patient’s last name | Alphabetic |
| P_mint | Patient’s middle name | Alphabetic |
| Service _code | Services’ unique identifier | Alphanumeric |
| Description | Service description | Alphabetic |
| Ser_patient_ID | Patient’s ID (FK) | Alphanumeric |
| Ser_SSN | Employee's SSN (FK) | Numeric |
| Receptionist_ID | Receptionist’s ID | Alphanumeric |
| Rec_SSN | Receptionist’s SSN (FK) | Numeric |

| | | |
|---|---|---|
| Recp_number | Record’s unique identifier | Numeric |
| Rec_date | Record’s creation date | Date |
| Observation | Patient’s observations in this record | Alphabetic |
| Rec_recepcionist_ID | Receptionist’s ID (FK) | Alphanumeric |
| Nurse_ID | Nurse’s ID | Alphanumeric |
| Nur_SSN | Nurse’s SSN (FK) | Numeric |
| Room_numb | Room’s unique identifier | Alphanumeric |
| Numb_bed | Number of beds in a room | Numeric |
| Date_in | Patient’s date admitted | Date |
| Date_out | Patient’s date discharged | Date |
| Room_type | Whether or not patient was admitted via ER | Alphabetic |
| Roo_nurse_ID | Nurse’s ID (FK) | Alphanumeric |
| Roo_patient_ID | Patient’s ID (FK) | Numeric |
| Doctor_ID | Doctor’s ID | Alphanumeric |
| Doc_SSN | Doctor’s SSN (FK) | Alphanumeric |
| Specialty | Doctor’s specialty. One of the seven given | Alphabetic |
| Vis_patient_ID | Patient’s ID (FK) | Numeric |
| Vis_doctor_ID | Doctor’s ID (FK) | Numeric |
| Visit_date | Date of the visit | Date |
| Med_code | Medicine’s unique identifier | Numeric |
| Med_name | Medicine’s name | Alphabetic |
| List_price | Medicine’s price | Numeric |
| Manufacturer | Medicine’s manufacturer | Alphabetic |
| Class | Medicine’s drug class | Alphabetic |
| Description | Medicine’s description | Alphabetic |

|  |  |  |
|---|---|---|
| Presc_code | Prescription’s unique identifier | Numeric |
| Frequency | Frequency of the prescribed medicine | Alphabetic |
| Dosage | Dosage of the prescribed medicine | Alphabetic |
| Pre_med_code | Medicine’s unique identifier (FK) | Numeric |
| Pre_med_name | Medicine’s name (FK) | Alphabetic |
| Pre_patient_ID | Patient’s ID (FK) | Numeric |
| Pre_doctor_ID | Doctor’s ID (FK) | Numeric |
| Tre_code | Treatment’s unique identifier | Numeric |
| Tre_cost | Treatment’s cost | Numeric |
| Take_med_name | Medicine’s name (FK) | Alphabetic |
| Tak_trea_code | Treatment’s unique identifier (FK) | Numeric |
| Number | Phone number | Numeric |
| Type | Type of phone | Alphabetic |
| Pho_patient_ID | Patient’s ID (FK) | Alphanumeric |
| Pho_SSN | Employee’s SSN (FK) | Numeric |

**3. Normalize relation schema in the database design that you get from step 4 into either 3NF or BCNF if it is necessary.** (10 points)



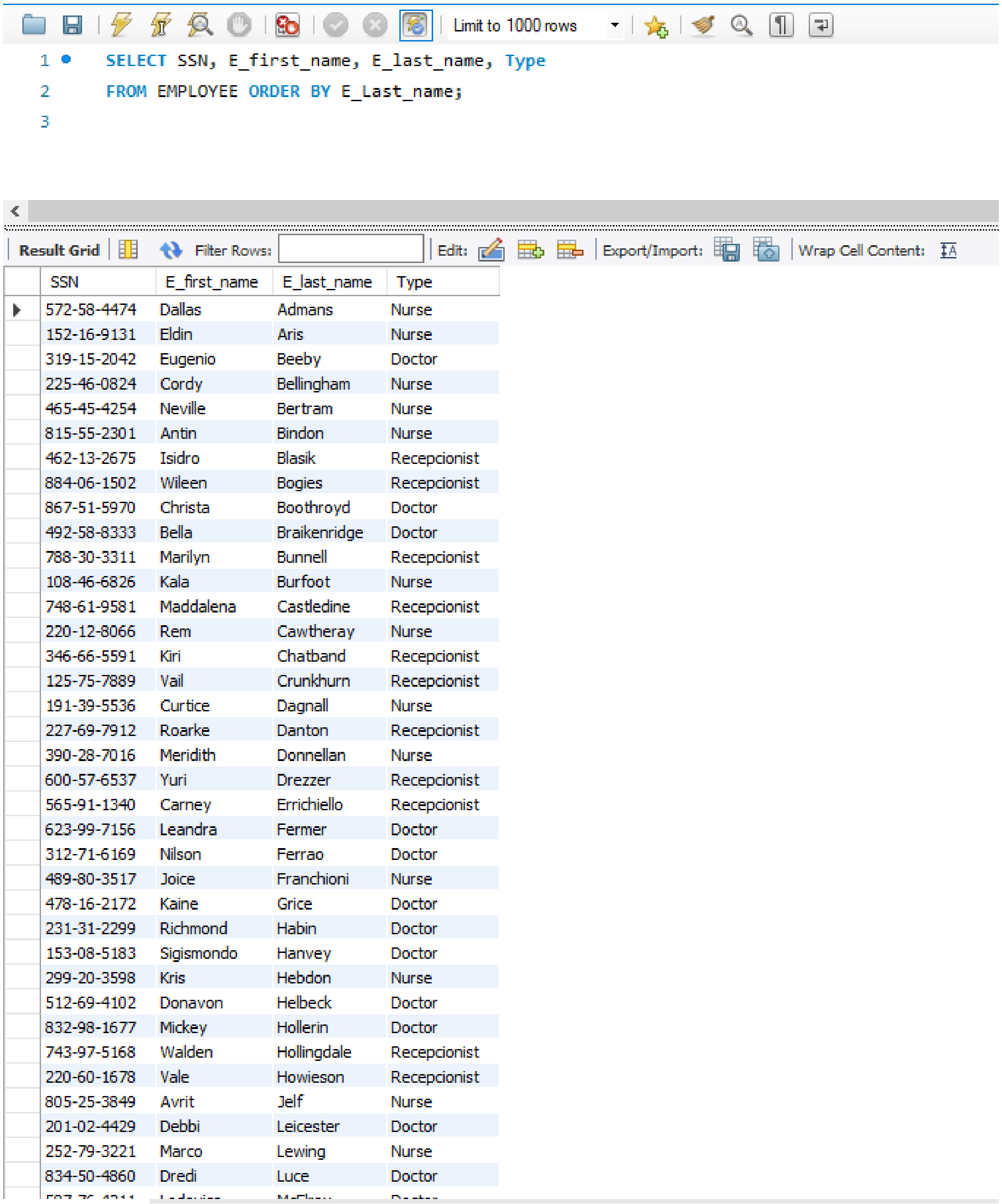
**4. Implement the relational database you get in step 5, via PostgreSQL, this includes creating the database, creating the corresponding relation schemas, data preparation and loading data into the database.** (30 points)

The implementation of the relational database can be found in the HospitalDB.sql file.

**5. Implement the given queries using PostgreSQL. Provide the SQL script for each query** (30 points)

*1. List the last name, name, employee number, type of employee of all employees ordered by last name.*

SELECT SSN, E_first_name, E_last_name, Type
FROM EMPLOYEE ORDER BY E_Last_name;

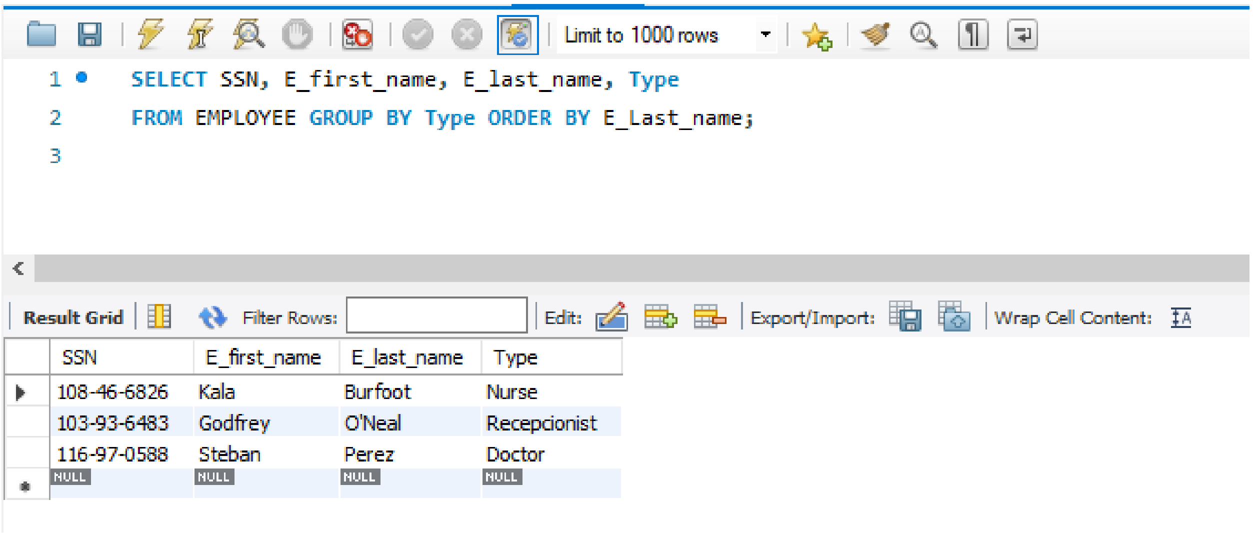


```
SELECT SSN, E_first_name, E_last_name, Type
FROM EMPLOYEE ORDER BY E_Last_name;
```

| SSN | E_first_name | E_last_name | Type |
|---|---|---|---|
| 572-58-4474 | Dallas | Admans | Nurse |
| 152-16-9131 | Eldin | Aris | Nurse |
| 319-15-2042 | Eugenio | Beeby | Doctor |
| 225-46-0824 | Cordy | Bellingham | Nurse |
| 465-45-4254 | Neville | Bertram | Nurse |
| 815-55-2301 | Antin | Bindon | Nurse |
| 462-13-2675 | Isidro | Blasik | Recepcionist |
| 884-06-1502 | Wileen | Bogies | Recepcionist |
| 867-51-5970 | Christa | Boothroyd | Doctor |
| 492-58-8333 | Bella | Braikenridge | Doctor |
| 788-30-3311 | Marilyn | Bunnell | Recepcionist |
| 108-46-6826 | Kala | Burfoot | Nurse |
| 748-61-9581 | Maddalena | Castledine | Recepcionist |
| 220-12-8066 | Rem | Cawtheray | Nurse |
| 346-66-5591 | Kiri | Chatband | Recepcionist |
| 125-75-7889 | Vail | Crunkhurn | Recepcionist |
| 191-39-5536 | Curtice | Dagnall | Nurse |
| 227-69-7912 | Roarke | Danton | Recepcionist |
| 390-28-7016 | Meridith | Donnellan | Nurse |
| 600-57-6537 | Yuri | Drezzer | Recepcionist |
| 565-91-1340 | Carney | Errichiello | Recepcionist |
| 623-99-7156 | Leandra | Fermer | Doctor |
| 312-71-6169 | Nilson | Ferrao | Doctor |
| 489-80-3517 | Joice | Franchioni | Nurse |
| 478-16-2172 | Kaine | Grice | Doctor |
| 231-31-2299 | Richmond | Habin | Doctor |
| 153-08-5183 | Sigismondo | Hanvey | Doctor |
| 299-20-3598 | Kris | Hebdon | Nurse |
| 512-69-4102 | Donavon | Helbeck | Doctor |
| 832-98-1677 | Mickey | Hollerin | Doctor |
| 743-97-5168 | Walden | Hollingdale | Recepcionist |
| 220-60-1678 | Vale | Howieson | Recepcionist |
| 805-25-3849 | Avrit | Jelf | Nurse |
| 201-02-4429 | Debbi | Leicester | Doctor |
| 252-79-3221 | Marco | Lewing | Nurse |
| 834-50-4860 | Dredi | Luce | Doctor |
| 587-76-4311 | Ludovico | McElroy | Doctor |

*2. List the last name, name, employee number, type of employee of all employees ordered by last name grouped by employee type.*

SELECT SSN, E_first_name, E_last_name, Type
FROM EMPLOYEE GROUP BY Type ORDER BY E_Last_name;



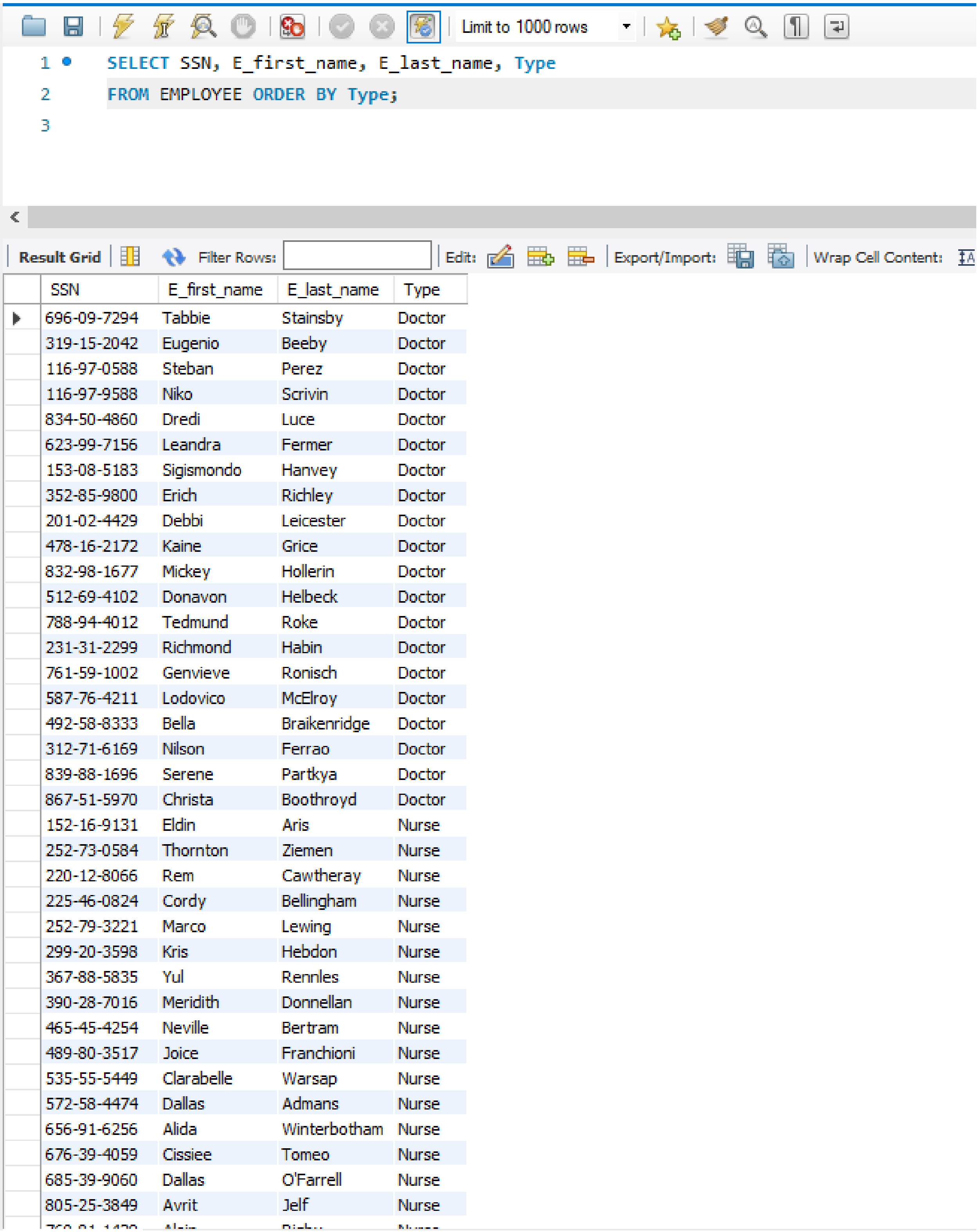
```
SELECT SSN, E_first_name, E_last_name, Type
FROM EMPLOYEE GROUP BY Type ORDER BY E_Last_name;
```

| SSN | E_first_name | E_last_name | Type |
|---|---|---|---|
| 108-46-6826 | Kala | Burfoot | Nurse |
| 103-93-6483 | Godfrey | O'Neal | Recepcionist |
| 116-97-0588 | Steban | Perez | Doctor |
| NULL | NULL | NULL | NULL |

Note: The query above does not show many results because of the GROUP BY statement.

The query below is an example of what kind of output we would get if we do ORDER BY Type instead of GROUP BY Type.

SELECT SSN, E_first_name, E_last_name, Type
FROM EMPLOYEE ORDER BY Type;

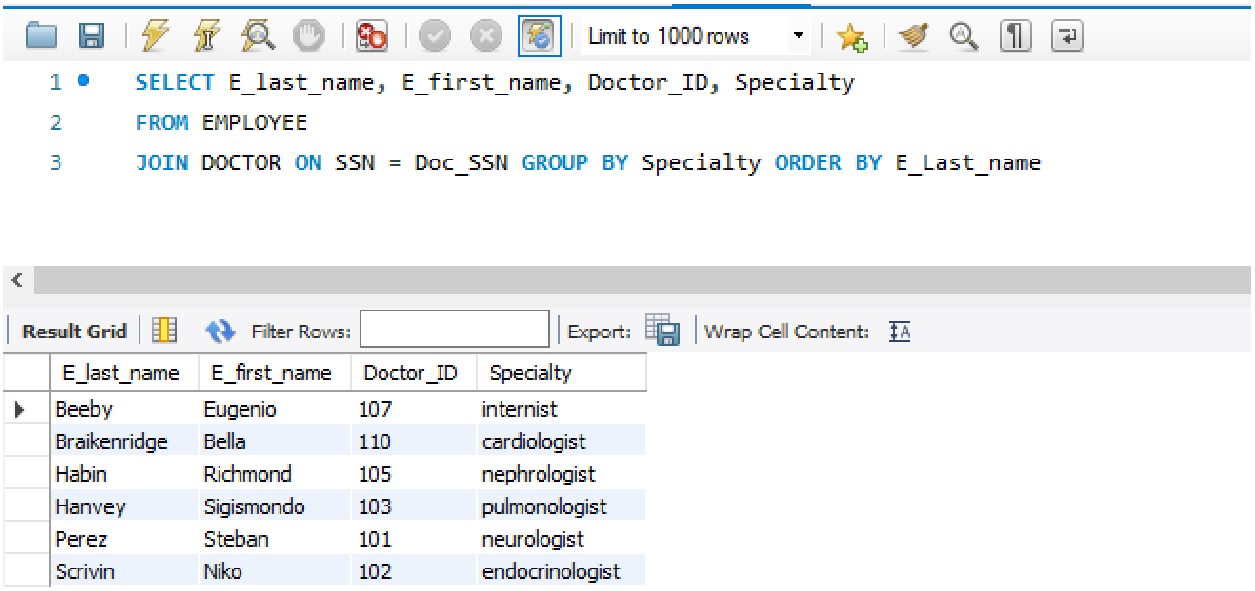


```
SELECT SSN, E_first_name, E_last_name, Type
FROM EMPLOYEE ORDER BY Type;
```

| SSN | E_first_name | E_last_name | Type |
|---|---|---|---|
| 696-09-7294 | Tabbie | Stainsby | Doctor |
| 319-15-2042 | Eugenio | Beeby | Doctor |
| 116-97-0588 | Steban | Perez | Doctor |
| 116-97-9588 | Niko | Scrivin | Doctor |
| 834-50-4860 | Dredi | Luce | Doctor |
| 623-99-7156 | Leandra | Fermer | Doctor |
| 153-08-5183 | Sigismondo | Hanvey | Doctor |
| 352-85-9800 | Erich | Richley | Doctor |
| 201-02-4429 | Debbi | Leicester | Doctor |
| 478-16-2172 | Kaine | Grice | Doctor |
| 832-98-1677 | Mickey | Hollerin | Doctor |
| 512-69-4102 | Donavon | Helbeck | Doctor |
| 788-94-4012 | Tedmund | Roke | Doctor |
| 231-31-2299 | Richmond | Habin | Doctor |
| 761-59-1002 | Genvieve | Ronisch | Doctor |
| 587-76-4211 | Lodovico | McElroy | Doctor |
| 492-58-8333 | Bella | Braikenridge | Doctor |
| 312-71-6169 | Nilson | Ferrao | Doctor |
| 839-88-1696 | Serene | Partkya | Doctor |
| 867-51-5970 | Christa | Boothroyd | Doctor |
| 152-16-9131 | Eldin | Aris | Nurse |
| 252-73-0584 | Thornton | Ziemen | Nurse |
| 220-12-8066 | Rem | Cawtheray | Nurse |
| 225-46-0824 | Cordy | Bellingham | Nurse |
| 252-79-3221 | Marco | Lewing | Nurse |
| 299-20-3598 | Kris | Hebdon | Nurse |
| 367-88-5835 | Yul | Rennles | Nurse |
| 390-28-7016 | Meridith | Donnellan | Nurse |
| 465-45-4254 | Neville | Bertram | Nurse |
| 489-80-3517 | Joice | Franchioni | Nurse |
| 535-55-5449 | Clarabelle | Warsap | Nurse |
| 572-58-4474 | Dallas | Admans | Nurse |
| 656-91-6256 | Alida | Winterbotham | Nurse |
| 676-39-4059 | Cissiee | Tomeo | Nurse |
| 685-39-9060 | Dallas | O'Farrell | Nurse |
| 805-25-3849 | Avrit | Jelf | Nurse |
| 760-01-1429 | Alain | Bigby | Nurse |

*3. List the name, last name, employee number, Specialty of doctors. Group by specialty and order by last name.*

SELECT E_last_name, E_first_name, Doctor_ID, Specialty
FROM EMPLOYEE
JOIN DOCTOR ON SSN = Doc_SSN GROUP BY Specialty ORDER BY E_Last_name



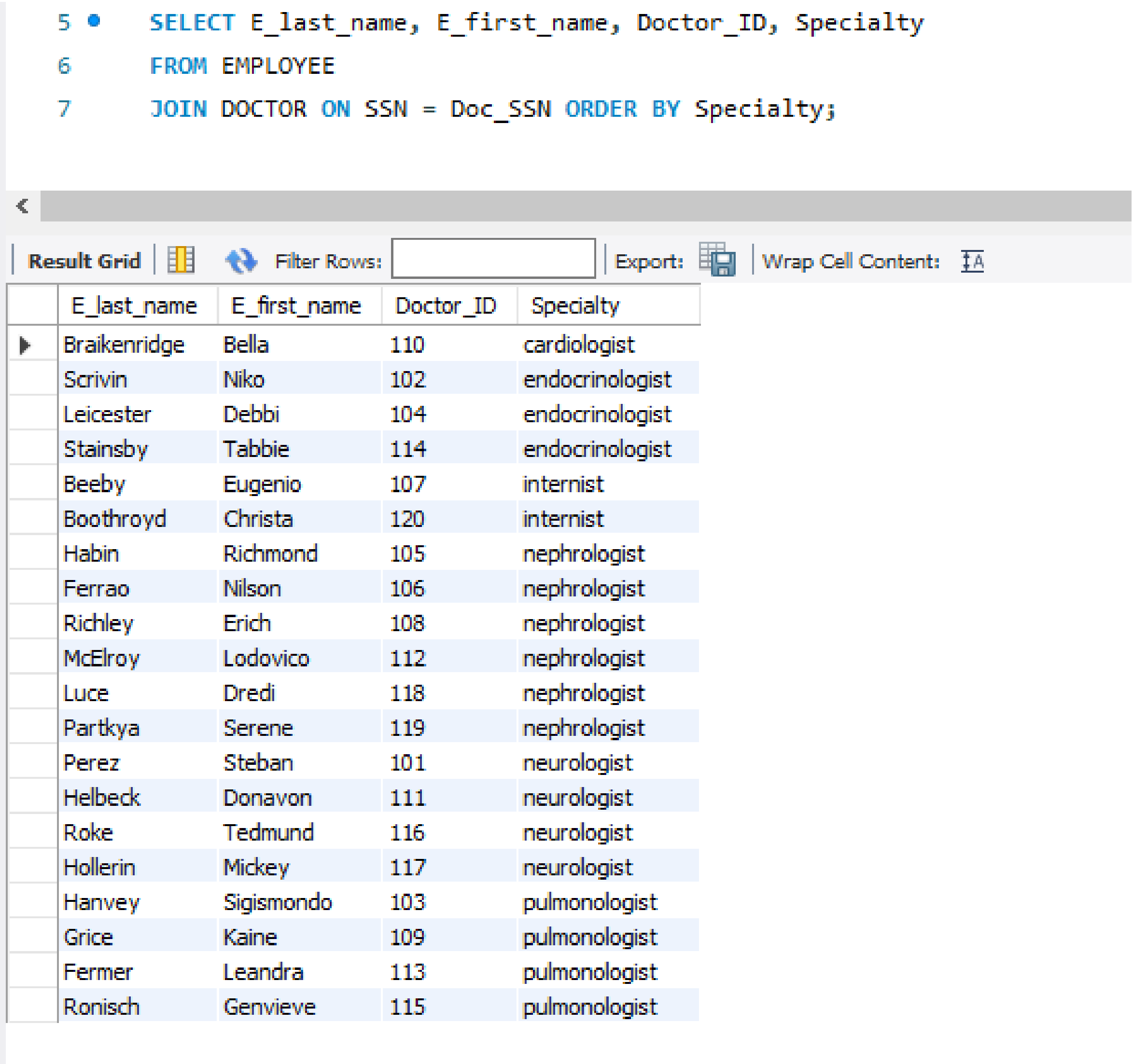
```
SELECT E_last_name, E_first_name, Doctor_ID, Specialty
FROM EMPLOYEE
JOIN DOCTOR ON SSN = Doc_SSN GROUP BY Specialty ORDER BY E_Last_name
```

| E_last_name | E_first_name | Doctor_ID | Specialty |
|---|---|---|---|
| Beeby | Eugenio | 107 | internist |
| Braikenridge | Bella | 110 | cardiologist |
| Habin | Richmond | 105 | nephrologist |
| Hanvey | Sigismondo | 103 | pulmonologist |
| Perez | Steban | 101 | neurologist |
| Scrivin | Niko | 102 | endocrinologist |

Note: The query above does not show many results because of the GROUP BY statement.

The query below is an example of what kind of output we would get if we do ORDER BY Specialty instead of GROUP BY Specialty.

SELECT E_last_name, E_first_name, Doctor_ID, Specialty
FROM EMPLOYEE
JOIN DOCTOR ON SSN = Doc_SSN ORDER BY Specialty;

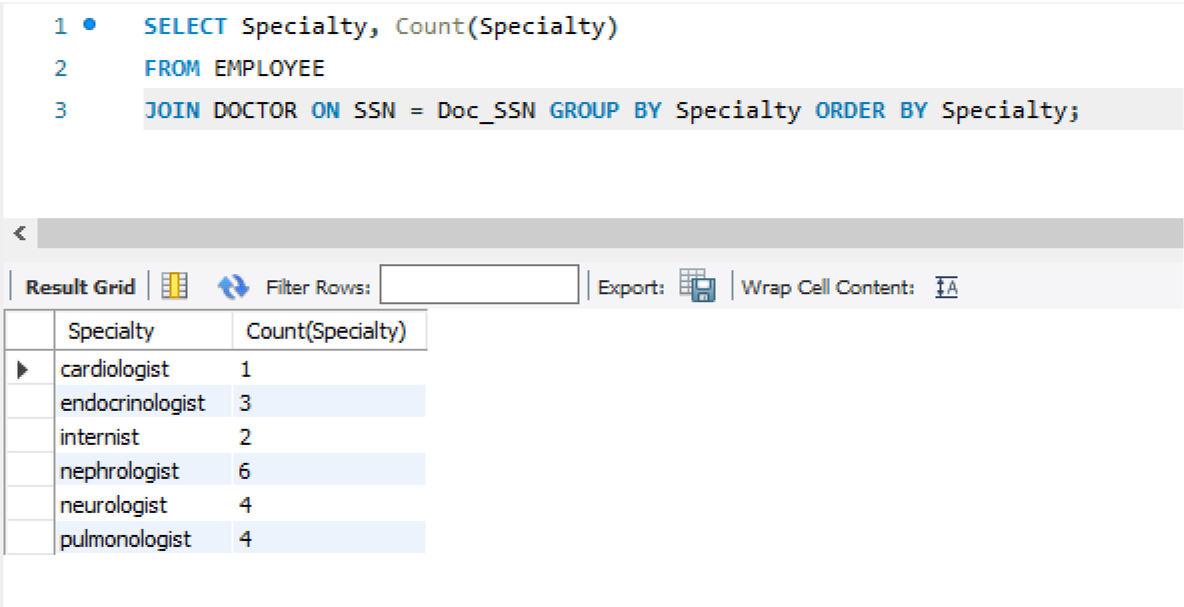


```
SELECT E_last_name, E_first_name, Doctor_ID, Specialty
FROM EMPLOYEE
JOIN DOCTOR ON SSN = Doc_SSN ORDER BY Specialty;
```

| E_last_name | E_first_name | Doctor_ID | Specialty |
|---|---|---|---|
| Braikenridge | Bella | 110 | cardiologist |
| Scrivin | Niko | 102 | endocrinologist |
| Leicester | Debbi | 104 | endocrinologist |
| Stainsby | Tabbie | 114 | endocrinologist |
| Beeby | Eugenio | 107 | internist |
| Boothroyd | Christa | 120 | internist |
| Habin | Richmond | 105 | nephrologist |
| Ferrao | Nilson | 106 | nephrologist |
| Richley | Erich | 108 | nephrologist |
| McElroy | Lodovico | 112 | nephrologist |
| Luce | Dredi | 118 | nephrologist |
| Partkya | Serene | 119 | nephrologist |
| Perez | Steban | 101 | neurologist |
| Helbeck | Donavon | 111 | neurologist |
| Roke | Tedmund | 116 | neurologist |
| Hollerin | Mickey | 117 | neurologist |
| Hanvey | Sigismondo | 103 | pulmonologist |
| Grice | Kaine | 109 | pulmonologist |
| Fermer | Leandra | 113 | pulmonologist |
| Ronisch | Genvieve | 115 | pulmonologist |

*4. List the count of doctors per specialty order the list by specialty name.*

SELECT Specialty, Count(Specialty)
FROM EMPLOYEE
JOIN DOCTOR ON SSN = Doc_SSN GROUP BY Specialty ORDER BY Specialty

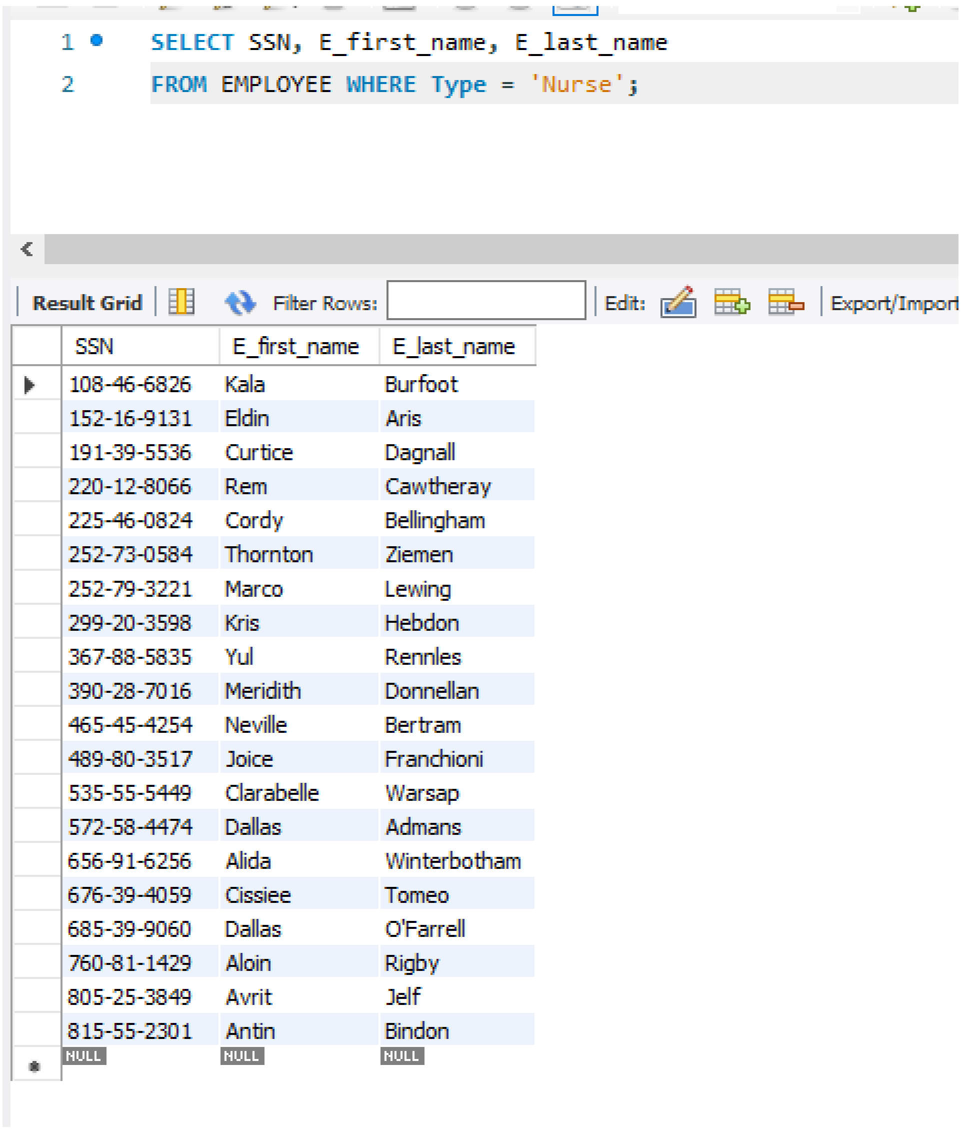


```sql
SELECT Specialty, Count(Specialty)
FROM EMPLOYEE
JOIN DOCTOR ON SSN = Doc_SSN GROUP BY Specialty ORDER BY Specialty;
```

| Specialty | Count(Specialty) |
|---|---|
| cardiologist | 1 |
| endocrinologist | 3 |
| internist | 2 |
| nephrologist | 6 |
| neurologist | 4 |
| pulmonologist | 4 |

*5. List the name, last name, employee number of all the nurses.*

SELECT SSN, E_first_name, E_last_name
FROM EMPLOYEE WHERE Type = 'Nurse';

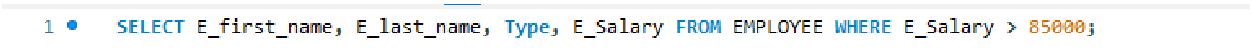


```
SELECT SSN, E_first_name, E_last_name
FROM EMPLOYEE WHERE Type = 'Nurse';
```

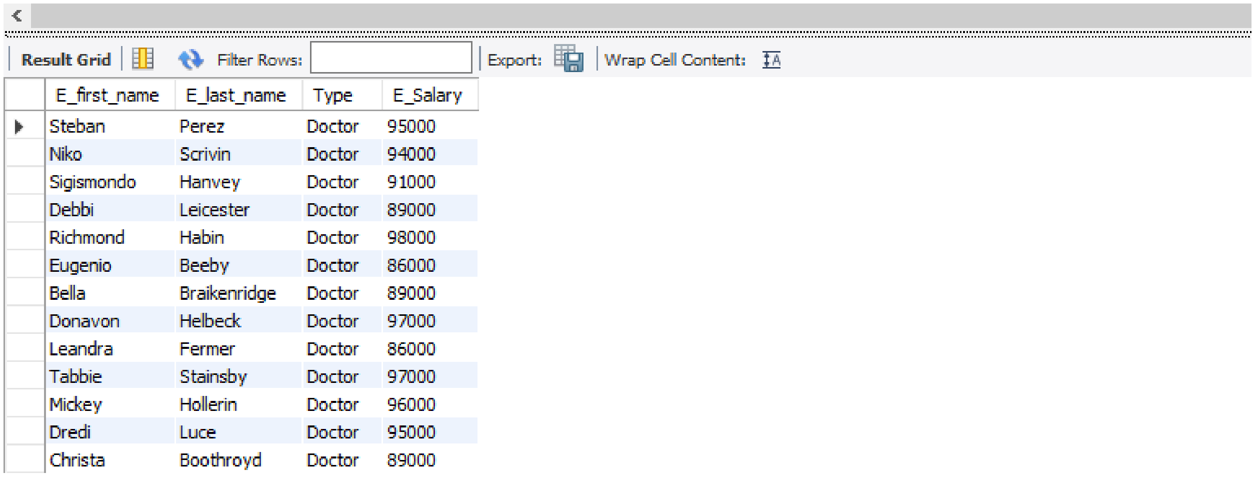
| SSN | E_first_name | E_last_name |
|---|---|---|
| 108-46-6826 | Kala | Burfoot |
| 152-16-9131 | Eldin | Aris |
| 191-39-5536 | Curtice | Dagnall |
| 220-12-8066 | Rem | Cawtheray |
| 225-46-0824 | Cordy | Bellingham |
| 252-73-0584 | Thornton | Ziemen |
| 252-79-3221 | Marco | Lewing |
| 299-20-3598 | Kris | Hebdon |
| 367-88-5835 | Yul | Rennles |
| 390-28-7016 | Meridith | Donnellan |
| 465-45-4254 | Neville | Bertram |
| 489-80-3517 | Joice | Franchioni |
| 535-55-5449 | Clarabelle | Warsap |
| 572-58-4474 | Dallas | Admans |
| 656-91-6256 | Alida | Winterbotham |
| 676-39-4059 | Cissiee | Tomeo |
| 685-39-9060 | Dallas | O'Farrell |
| 760-81-1429 | Aloin | Rigby |
| 805-25-3849 | Avrit | Jelf |
| 815-55-2301 | Antin | Bindon |
| NULL | NULL | NULL |

*6. List the employees name, last name, employee type, salary with salaries greater than 85K.*

SELECT E_first_name, E_last_name, Type, E_Salary FROM EMPLOYEE WHERE E_Salary > 85000;



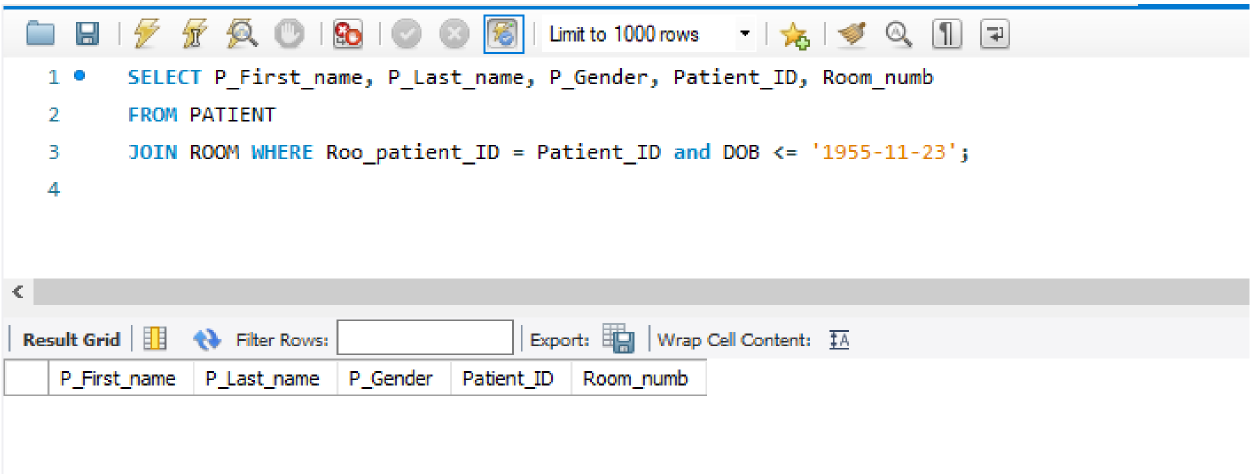
```sql
SELECT E_first_name, E_last_name, Type, E_Salary FROM EMPLOYEE WHERE E_Salary > 85000;
```



| E_first_name | E_last_name | Type | E_Salary |
|---|---|---|---|
| Steban | Perez | Doctor | 95000 |
| Niko | Scrivin | Doctor | 94000 |
| Sigismondo | Hanvey | Doctor | 91000 |
| Debbi | Leicester | Doctor | 89000 |
| Richmond | Habin | Doctor | 98000 |
| Eugenio | Beeby | Doctor | 86000 |
| Bella | Braikenridge | Doctor | 89000 |
| Donavon | Helbeck | Doctor | 97000 |
| Leandra | Fermer | Doctor | 86000 |
| Tabbie | Stainsby | Doctor | 97000 |
| Mickey | Hollerin | Doctor | 96000 |
| Dredi | Luce | Doctor | 95000 |
| Christa | Boothroyd | Doctor | 89000 |

*7. List the name, last name, sex, patient id and room number of all the patients not discharged yet and who are older than 65 years old.*

SELECT P_First_name, P_Last_name, P_Gender, Patient_ID, Room_numb
FROM PATIENT
JOIN ROOM WHERE Roo_patient_ID = Patient_ID and DOB <= '1955-11-23';

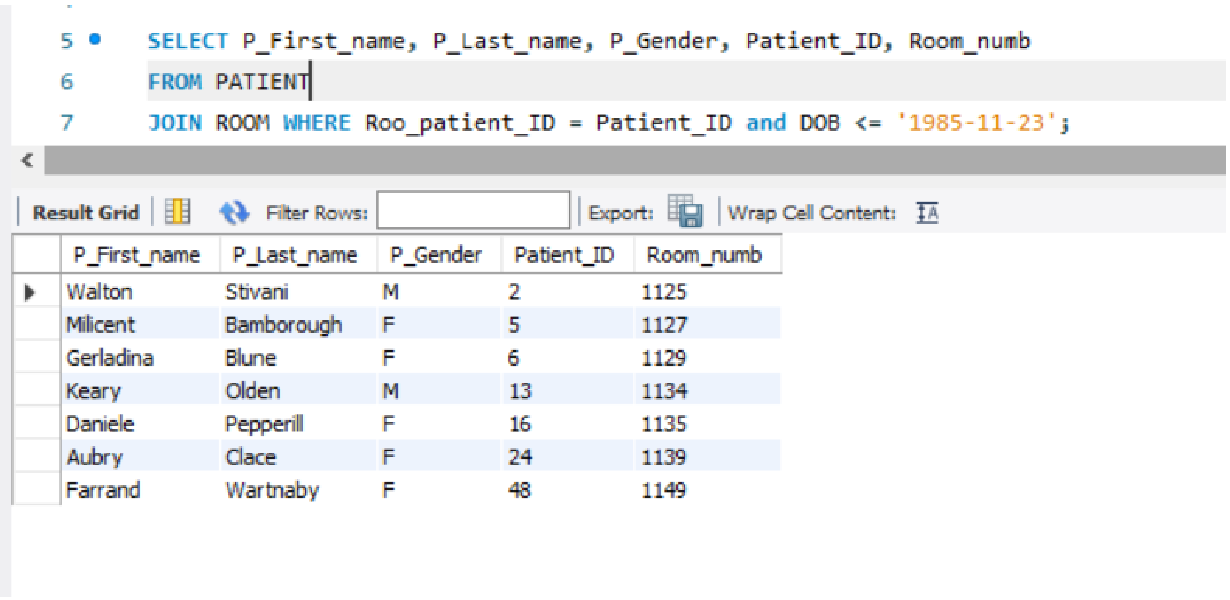


```
SELECT P_First_name, P_Last_name, P_Gender, Patient_ID, Room_numb
FROM PATIENT
JOIN ROOM WHERE Roo_patient_ID = Patient_ID and DOB <= '1955-11-23';
```

| P_First_name | P_Last_name | P_Gender | Patient_ID | Room_numb |
|---|---|---|---|---|

Note: There are no admitted patients in the data who are older than 65 years old.

The query below is an example of what the output will look like for patients who are older than 35 years old.

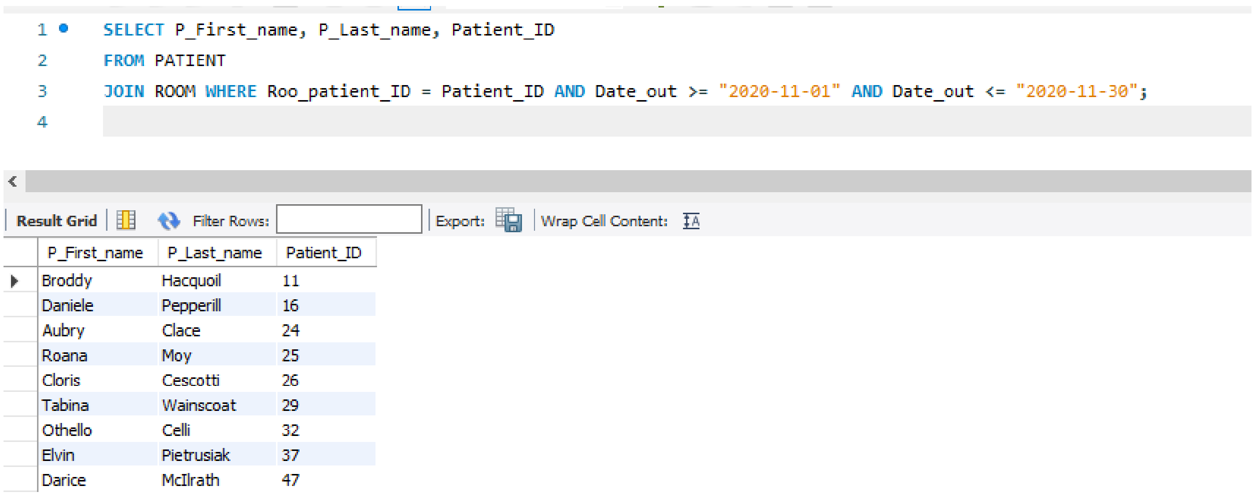


```
SELECT P_First_name, P_Last_name, P_Gender, Patient_ID, Room_numb
FROM PATIENT
JOIN ROOM WHERE Roo_patient_ID = Patient_ID and DOB <= '1985-11-23';
```

| P_First_name | P_Last_name | P_Gender | Patient_ID | Room_numb |
|---|---|---|---|---|
| Walton | Stivani | M | 2 | 1125 |
| Milicent | Bamborough | F | 5 | 1127 |
| Gerladina | Blune | F | 6 | 1129 |
| Keary | Olden | M | 13 | 1134 |
| Daniele | Pepperill | F | 16 | 1135 |
| Aubry | Clace | F | 24 | 1139 |
| Farrand | Wartnaby | F | 48 | 1149 |

*8. List the patient name, last name, patient id of patients discharged in one specific month (specified the month based on the data you used to populate the database).*

SELECT P_First_name, P_Last_name, Patient_ID
FROM PATIENT
JOIN ROOM WHERE Roo_patient_ID = Patient_ID AND Date_out >= "2020-11-01" AND Date_out <= "2020-11-30";



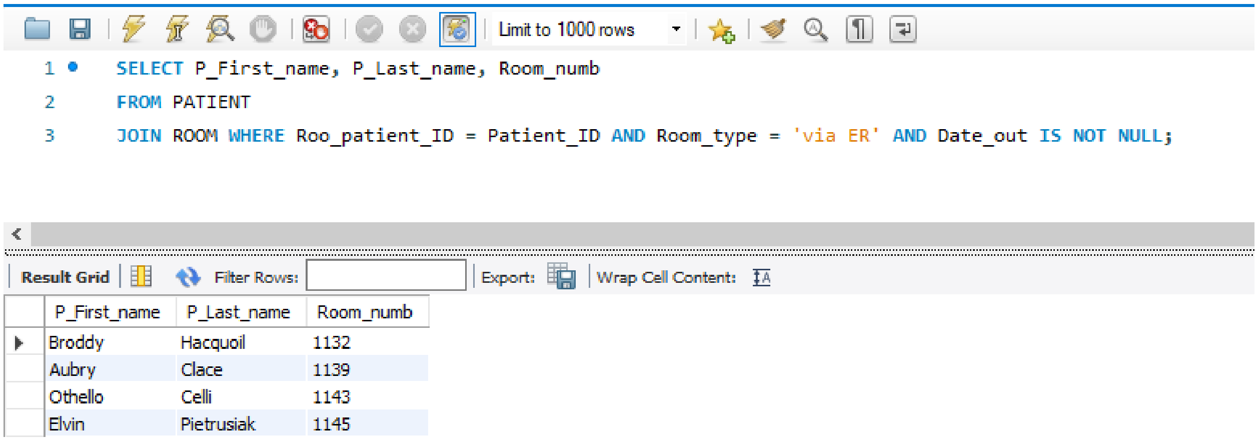
```
SELECT P_First_name, P_Last_name, Patient_ID
FROM PATIENT
JOIN ROOM WHERE Roo_patient_ID = Patient_ID AND Date_out >= "2020-11-01" AND Date_out <= "2020-11-30";
```

| P_First_name | P_Last_name | Patient_ID |
|---|---|---|
| Broddy | Hacquoil | 11 |
| Daniele | Pepperill | 16 |
| Aubry | Clace | 24 |
| Roana | Moy | 25 |
| Cloris | Cescotti | 26 |
| Tabina | Wainscoat | 29 |
| Othello | Celli | 32 |
| Elvin | Pietrusiak | 37 |
| Darice | McIlrath | 47 |

Note: Based on November.

*9. List the name and last name and room number of patients admitted through the ER and already discharged*

SELECT P_First_name, P_Last_name, Room_numb
FROM PATIENT
JOIN ROOM WHERE Roo_patient_ID = Patient_ID AND Room_type = 'via ER' AND Date_out IS NOT NULL;

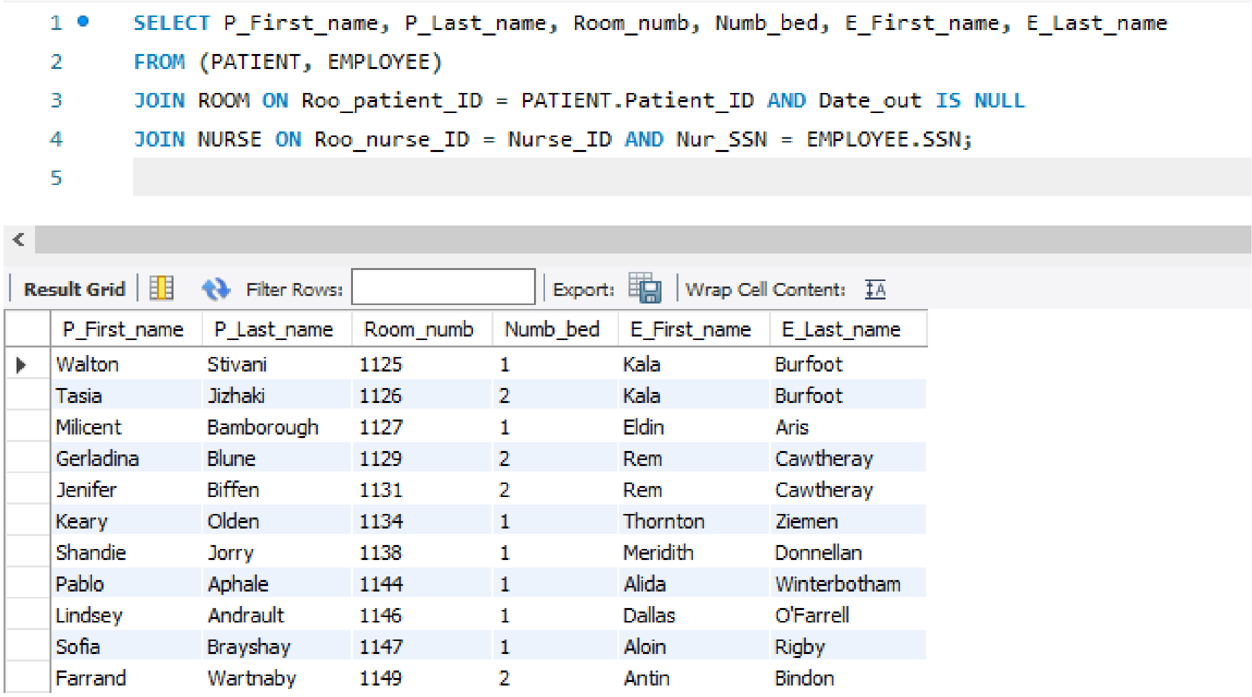


```
SELECT P_First_name, P_Last_name, Room_numb
FROM PATIENT
JOIN ROOM WHERE Roo_patient_ID = Patient_ID AND Room_type = 'via ER' AND Date_out IS NOT NULL;
```

| P_First_name | P_Last_name | Room_numb |
|---|---|---|
| Broddy | Hacquoil | 1132 |
| Aubry | Clace | 1139 |
| Othello | Celli | 1143 |
| Elvin | Pietrusiak | 1145 |

*10. List the name and last name, assigned room, room type and assigned nurse name and last name of patients not discharged yet.*

SELECT P_First_name, P_Last_name, Room_numb, Numb_bed, E_First_name, E_Last_name
FROM (PATIENT, EMPLOYEE)
JOIN ROOM ON Roo_patient_ID = PATIENT.Patient_ID AND Date_out IS NULL
JOIN NURSE ON Roo_nurse_ID = Nurse_ID AND Nur_SSN = EMPLOYEE.SSN;

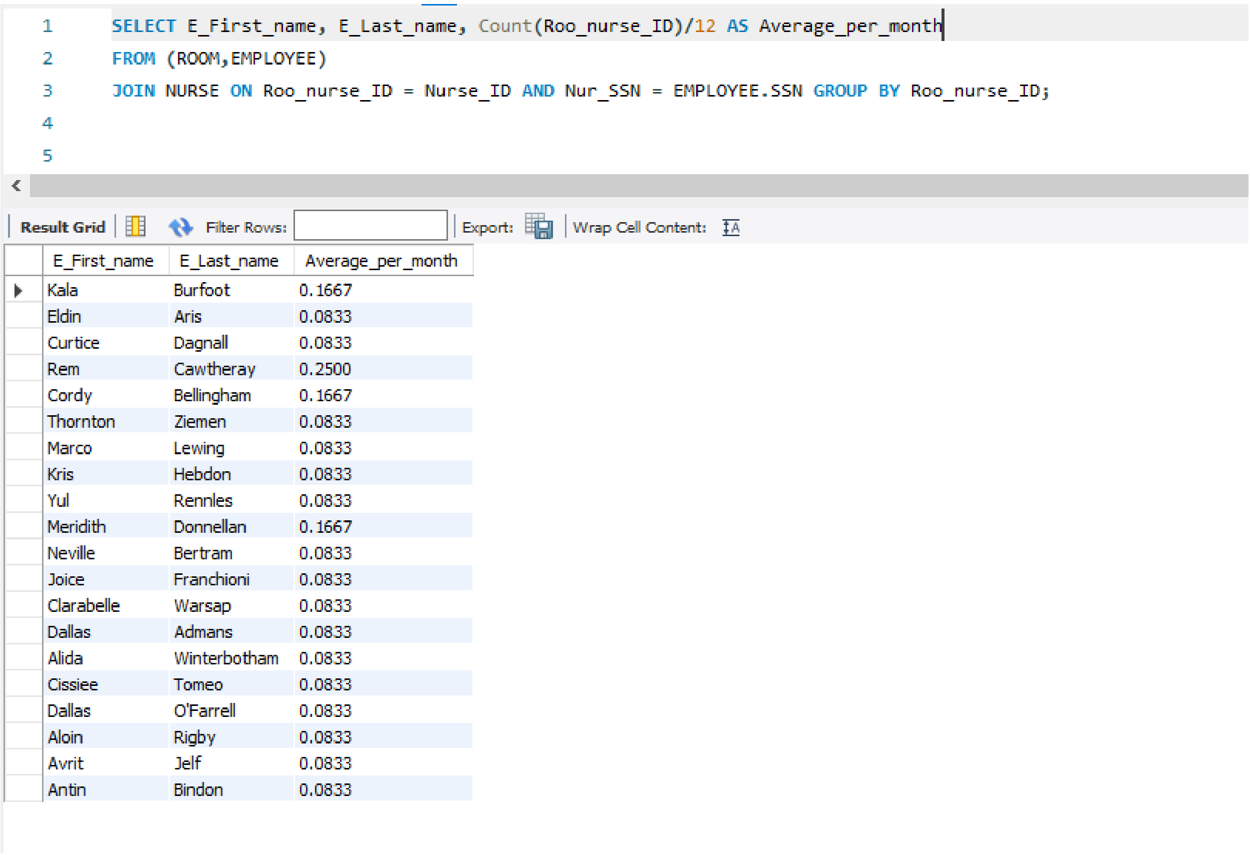


```
SELECT P_First_name, P_Last_name, Room_numb, Numb_bed, E_First_name, E_Last_name
FROM (PATIENT, EMPLOYEE)
JOIN ROOM ON Roo_patient_ID = PATIENT.Patient_ID AND Date_out IS NULL
JOIN NURSE ON Roo_nurse_ID = Nurse_ID AND Nur_SSN = EMPLOYEE.SSN;
```

| P_First_name | P_Last_name | Room_numb | Numb_bed | E_First_name | E_Last_name |
|---|---|---|---|---|---|
| Walton | Stivani | 1125 | 1 | Kala | Burfoot |
| Tasia | Jizhaki | 1126 | 2 | Kala | Burfoot |
| Milicent | Bamborough | 1127 | 1 | Eldin | Aris |
| Gerladina | Blune | 1129 | 2 | Rem | Cawtheray |
| Jenifer | Biffen | 1131 | 2 | Rem | Cawtheray |
| Keary | Olden | 1134 | 1 | Thornton | Ziemen |
| Shandie | Jorry | 1138 | 1 | Meridith | Donnellan |
| Pablo | Aphale | 1144 | 1 | Alida | Winterbotham |
| Lindsey | Andrault | 1146 | 1 | Dallas | O'Farrell |
| Sofia | Brayshay | 1147 | 1 | Aloin | Rigby |
| Farrand | Wartnaby | 1149 | 2 | Antin | Bindon |

*11. List the nurse name, nurse last name and average patient they took care per month.*

SELECT E_First_name, E_Last_name, Count(Roo_nurse_ID)/12 AS Average_per_month
FROM (ROOM,EMPLOYEE)
JOIN NURSE ON Roo_nurse_ID = Nurse_ID AND Nur_SSN = EMPLOYEE.SSN GROUP BY Roo_nurse_ID;

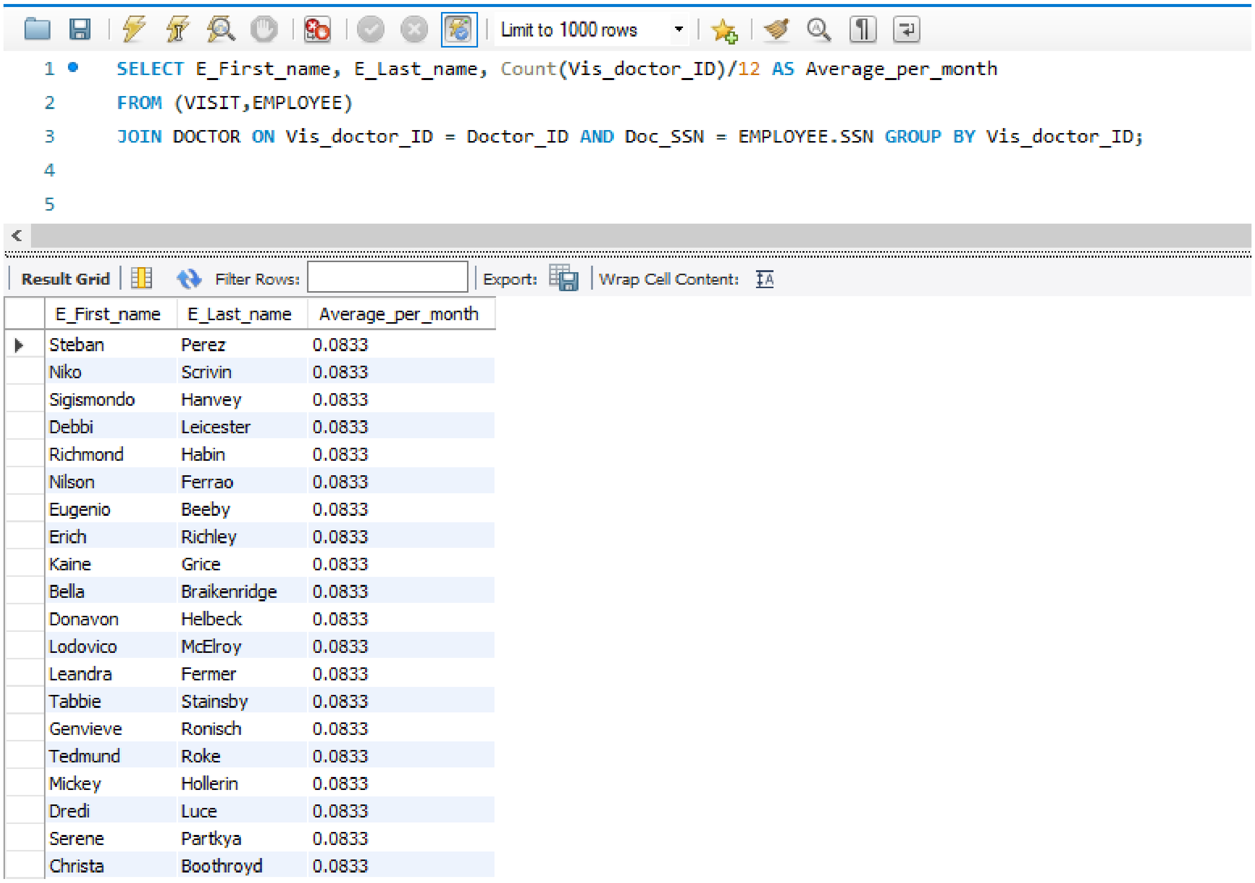


```
SELECT E_First_name, E_Last_name, Count(Roo_nurse_ID)/12 AS Average_per_month
FROM (ROOM,EMPLOYEE)
JOIN NURSE ON Roo_nurse_ID = Nurse_ID AND Nur_SSN = EMPLOYEE.SSN GROUP BY Roo_nurse_ID;
```

| E_First_name | E_Last_name | Average_per_month |
|---|---|---|
| Kala | Burfoot | 0.1667 |
| Eldin | Aris | 0.0833 |
| Curtice | Dagnall | 0.0833 |
| Rem | Cawtheray | 0.2500 |
| Cordy | Bellingham | 0.1667 |
| Thornton | Ziemen | 0.0833 |
| Marco | Lewing | 0.0833 |
| Kris | Hebdon | 0.0833 |
| Yul | Rennles | 0.0833 |
| Meridith | Donnellan | 0.1667 |
| Neville | Bertram | 0.0833 |
| Joice | Franchioni | 0.0833 |
| Clarabelle | Warsap | 0.0833 |
| Dallas | Admans | 0.0833 |
| Alida | Winterbotham | 0.0833 |
| Cissiee | Tomeo | 0.0833 |
| Dallas | O'Farrell | 0.0833 |
| Aloin | Rigby | 0.0833 |
| Avrit | Jelf | 0.0833 |
| Antin | Bindon | 0.0833 |

*12. List the doctor name and last name and average patient attended per month for all the doctors.*

SELECT E_First_name, E_Last_name, Count(Vis_doctor_ID)/12 AS Average_per_month FROM (VISIT,EMPLOYEE) JOIN DOCTOR ON Vis_doctor_ID = Doctor_ID AND Doc_SSN = EMPLOYEE.SSN GROUP BY Vis_doctor_ID;

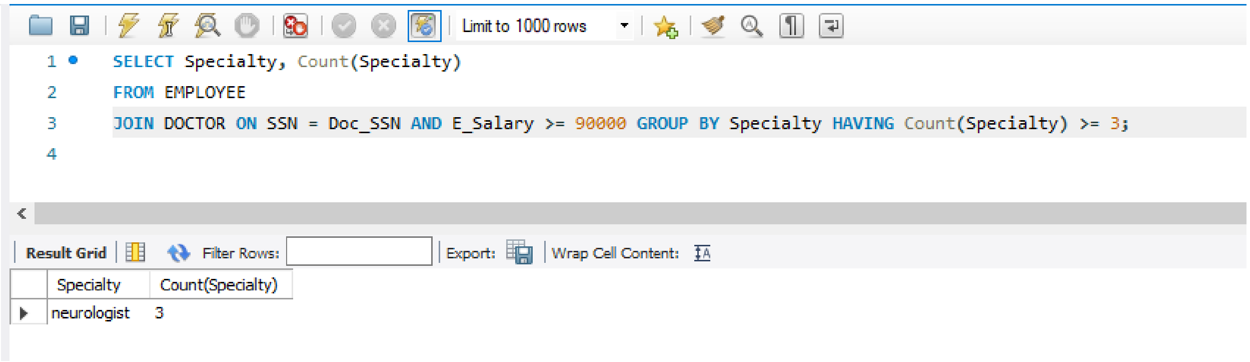


```
SELECT E_First_name, E_Last_name, Count(Vis_doctor_ID)/12 AS Average_per_month
FROM (VISIT,EMPLOYEE)
JOIN DOCTOR ON Vis_doctor_ID = Doctor_ID AND Doc_SSN = EMPLOYEE.SSN GROUP BY Vis_doctor_ID;
```

| E_First_name | E_Last_name | Average_per_month |
|---|---|---|
| Steban | Perez | 0.0833 |
| Niko | Scrivin | 0.0833 |
| Sigismondo | Hanvey | 0.0833 |
| Debbi | Leicester | 0.0833 |
| Richmond | Habin | 0.0833 |
| Nilson | Ferrao | 0.0833 |
| Eugenio | Beeby | 0.0833 |
| Erich | Richley | 0.0833 |
| Kaine | Grice | 0.0833 |
| Bella | Braikenridge | 0.0833 |
| Donavon | Helbeck | 0.0833 |
| Lodovico | McElroy | 0.0833 |
| Leandra | Fermer | 0.0833 |
| Tabbie | Stainsby | 0.0833 |
| Genvieve | Ronisch | 0.0833 |
| Tedmund | Roke | 0.0833 |
| Mickey | Hollerin | 0.0833 |
| Dredi | Luce | 0.0833 |
| Serene | Partkya | 0.0833 |
| Christa | Boothroyd | 0.0833 |

*13. List the specialty and number of doctors for the specialty that has more than 3 doctors that make more than 100K a year.*

SELECT Specialty, Count(Specialty)
FROM EMPLOYEE
JOIN DOCTOR ON SSN = Doc_SSN AND E_Salary >= 90000 GROUP BY Specialty
HAVING Count(Specialty) >= 3;



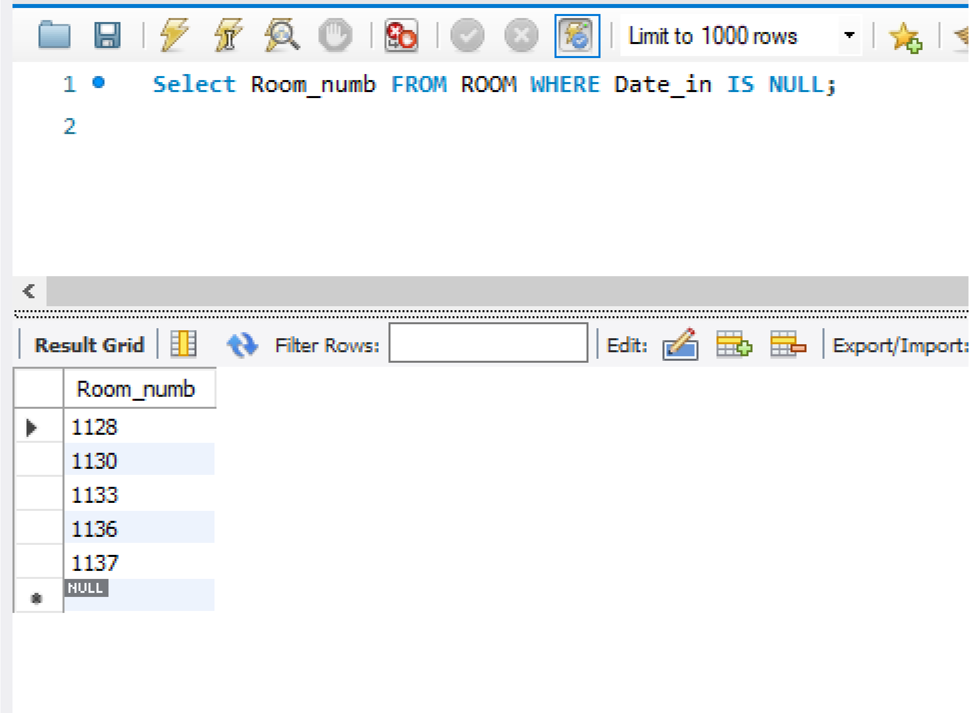
```
SELECT Specialty, Count(Specialty)
FROM EMPLOYEE
JOIN DOCTOR ON SSN = Doc_SSN AND E_Salary >= 90000 GROUP BY Specialty HAVING Count(Specialty) >= 3;
```

| Specialty | Count(Specialty) |
|---|---|
| neurologist | 3 |

Note: We do not have any employees whose salary is over 100K a year, we decided to use 90K instead.

*14. List the room number of any empty room.*

Select Room_numb FROM ROOM WHERE Date_in IS NULL;

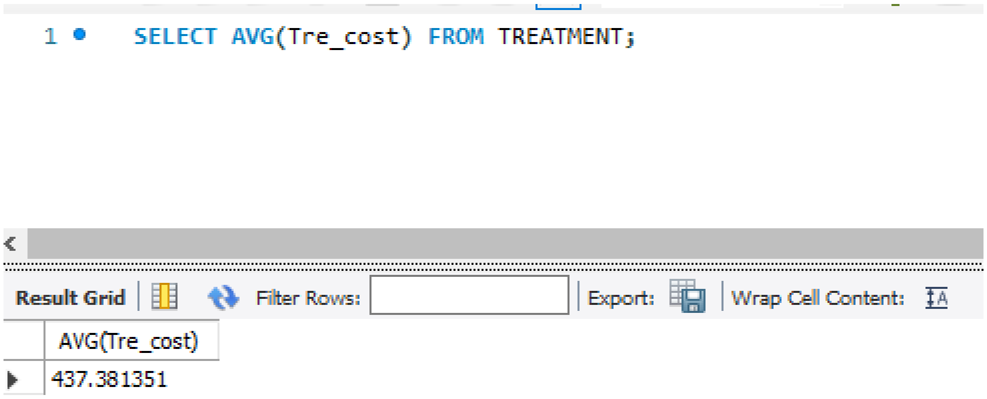


```
Select Room_numb FROM ROOM WHERE Date_in IS NULL;
```

| Room_numb |
|---|
| 1128 |
| 1130 |
| 1133 |
| 1136 |
| 1137 |
| NULL |

*15. List the average cost of a treatment.*

SELECT AVG(Tre_cost) FROM TREATMENT;



```
SELECT AVG(Tre_cost) FROM TREATMENT;
```

| AVG(Tre_cost) |
|---|
| 437.381351 |