Hospital Database

Introduction

In this project, we develop a database management system for a private hospital. Hospital databases are a critical component for the digital management of a hospital. They contain sensitive information about patients, as well as organizational information for the staff or hospital assets.

A well-designed database system can be helpful in building a management software that can be used by the receptionists, doctors and nurses of the hospital. It can help them keep track of appointments, tasks, stationary patients, medical records and payments. The database can also be used to build a future online presence for the hospital, so that patients book appointments with doctors.

We identify:

- 22 entities, including:
    - 6 weak entities
    - 1 associative entity
- 28 relationships

Moreover, we identify 4 main entities in our database management system, namely: assets, departments, doctors and staff. These entities are central aspects of our database, as they contain many relationships with other entities.

Developing our project is a case of greenfield engineering, meaning that no prior system exists and we will develop from scratch. In our case, requirements are extracted from the client. We take as a reference for the client the American Hospital in Tirana.

Finally, we identify the following quality requirements:

- Robustness: The system must be able to prevent unwanted delete queries that might violate integrity of the data. For that reason, we use triggers.
- Safety: Protection against unwanted incidents, using triggers and reversible transactions.
- Maintainability: The system should be well designed, so that it always produces the correct behavior and feature changes must be simple to do.
- Portability: The system must be easy to deploy in different operating systems or database systems. For that reason, we use as little dialect specific queries as possible.

Analysis with Entity-Relationship diagrams

Aspect: Staff

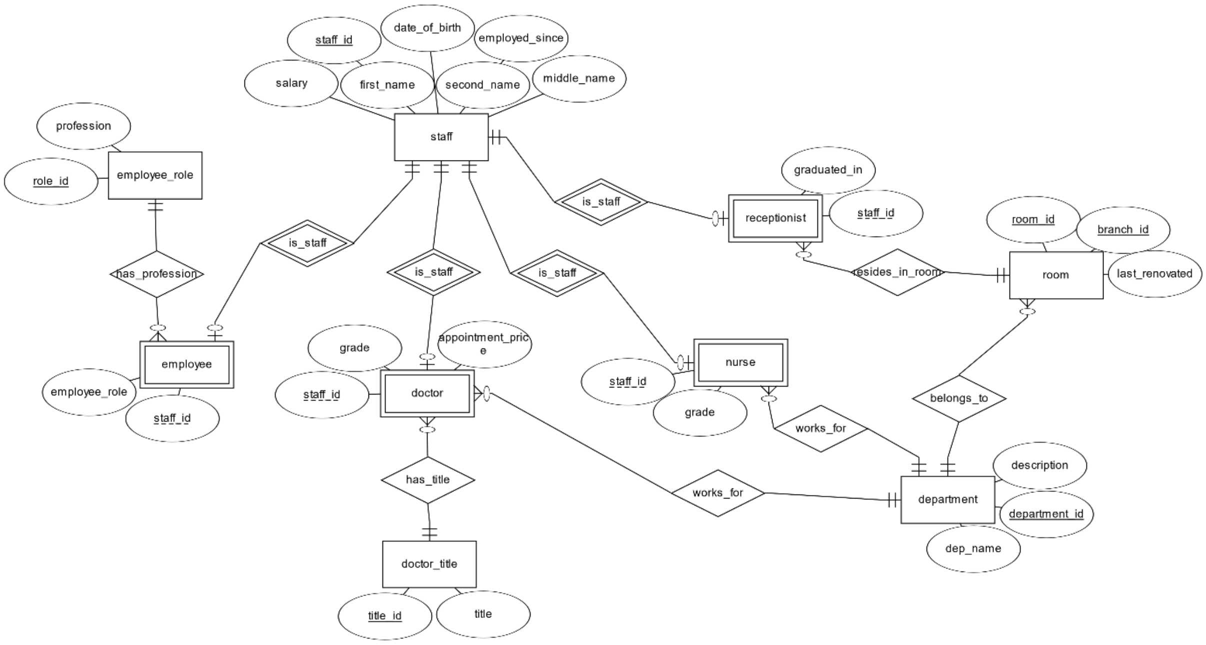


Figure 1. ER Diagram aspect around staff main entity.

Entities:

staff, receptionist, nurse, doctor, doctor_title, employee, employee_role, department

Relationships:

employee 0..N – has_profession – 1..1 employee_role

employee 0..1 – is_staff – 1..1 staff

doctor 0..1 – is_staff – 1..1 staff

doctor 0..N – has_title – 1..1 doctor_title

doctor 0..N – works_for – 1..1 department

nurse 0..1 – is_staff – 1..1 staff

nurse 0..N – works_for – 1..1 department

receptionist 0..1 – is_staff – 1..1 staff

receptionist 0..N – works_for – 1..1 department

Aspect: Doctor

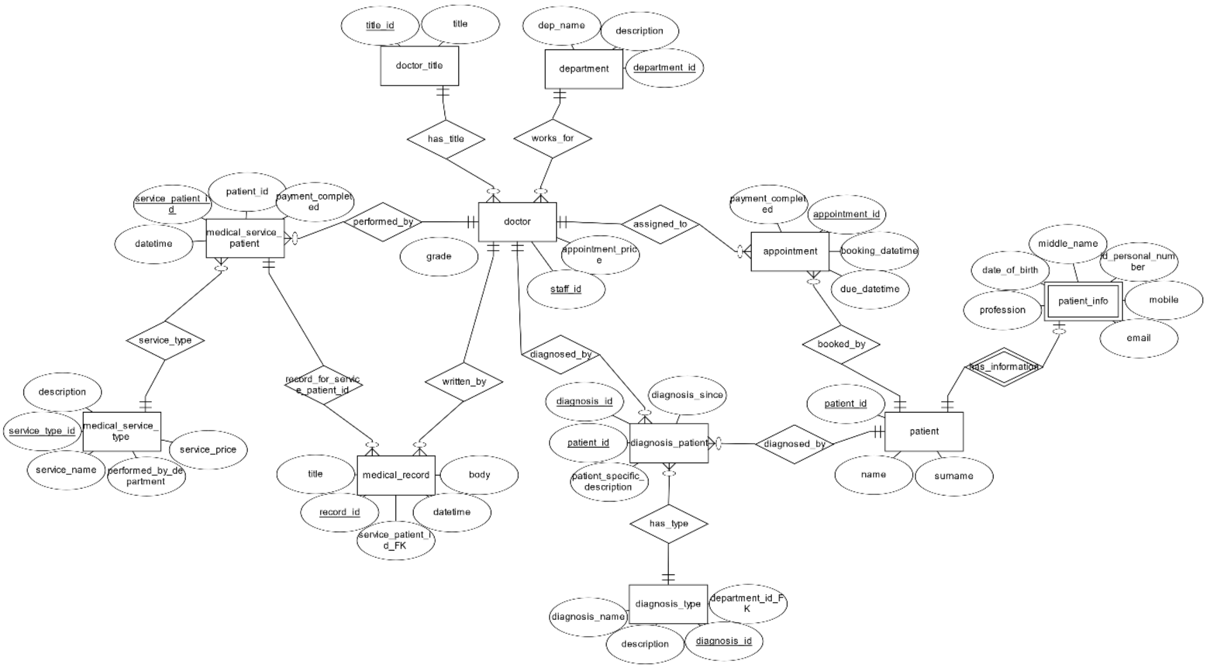


Figure 2. ER Diagram aspect around doctor main entity.

<u>Entities:</u>

doctor, doctor_title, department, medical_service_patient, appointment, patient, patient_info, diagnosis_patient, diagnosis_type, medical_record, medical_service_type

<u>Relationships:</u>

doctor 0..N – has_title – 1..1 doctor_title

doctor 0..N – works_for – 1..1 department

doctor 1..1 – performed_by – 0..N medical_service_patient

doctor 1..1 – assigned_to – 0..N appointment

doctor 1..1 – written_by – 0..N medical_record

medical_service_patient 1..1 – record_for_service_patient_id – 0..N medical_record

doctor 1..1 – diagnosed_by – 0..N diagnosis_patient

doctor 1..1 – assigned_to – 0..N appointment

diagnosis_patient 0..N – has_type – 1..1 diagnosis_type

diagnosis_patient 0..N – diagnosed_by – 1..1 patient

appointment 0..N – booked_by – 1..1 patient

patient 1..1 – has_information – 0..1 patient_info

Aspect: Department

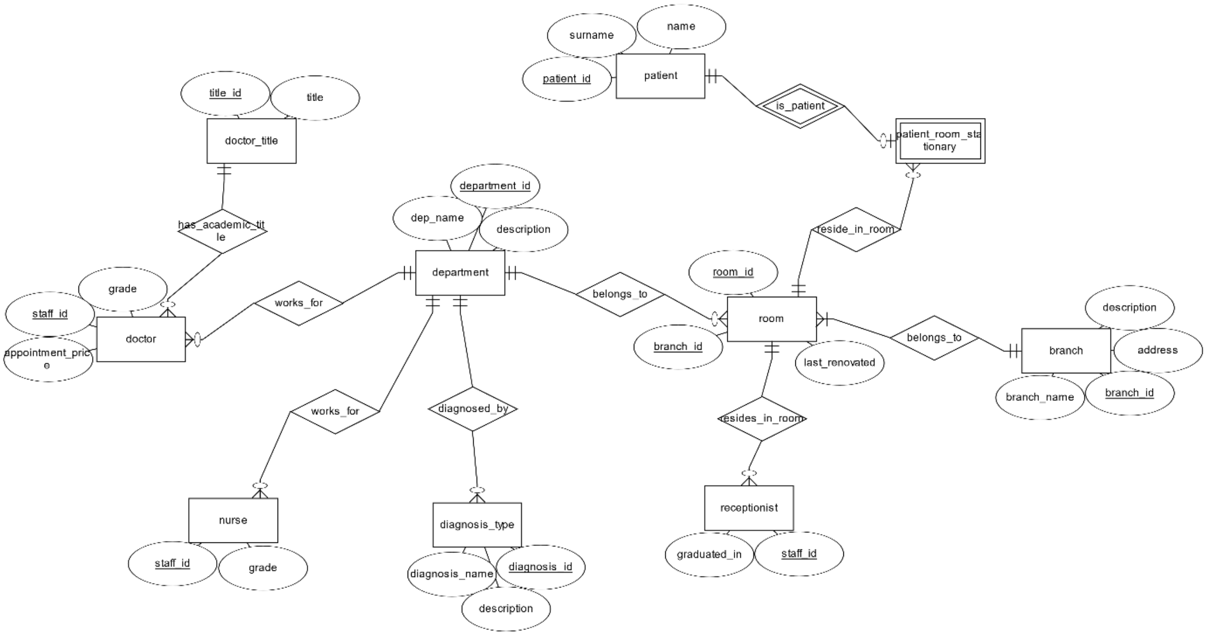


Figure 3. ER Diagram aspect around department main entity.

Entities:

doctor, doctor_title, department, nurse, receptionist, diagnosis_type, room, branch, patient, patient_room_stationary

Relationships:

doctor 0..N – has_academic_title – 1..1 doctor_title

doctor 0..N – works_for – 1..1 department

nurse 0..N – works_for – 1..1 department

receptionist 1..N – works_for – 1..1 department

diagnosis_type 0..N – diagnosed_by – 1..1 department

room 0..N – belongs_to – 1..1 department

room 1..1 – reside_in_room – 0..N patient_room_stationary

room 1..N – belongs_to – 1..1 branch

patient_room_stationary 0..1 – is_patient – 1..1 patient

Aspect: Assets

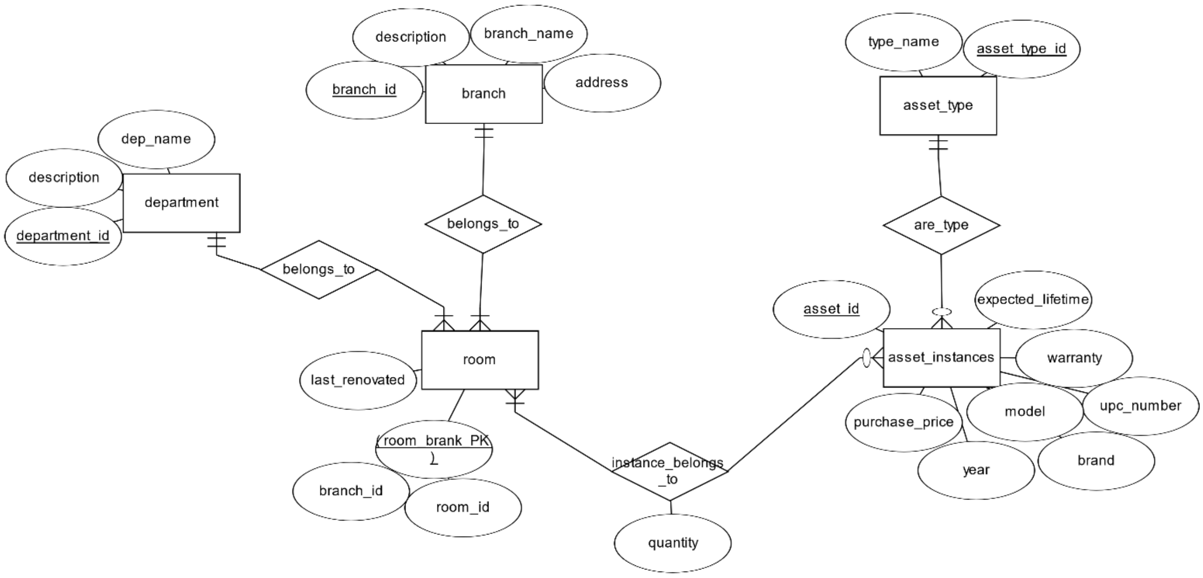


Figure 4. ER Diagram aspect around assets main entity.

<u>Entities:</u>

department, room, branch, asset_instances, asset_type

<u>Relationships:</u>

department 1..1 – belongs_to – 1..N room

branch 1..1 – belongs_to – 1..N room

asset_instances 0..N – instance_belongs_to – 1..N room

asset_type 1..1 – are_type – 0..N asset_instances

Full UML diagram

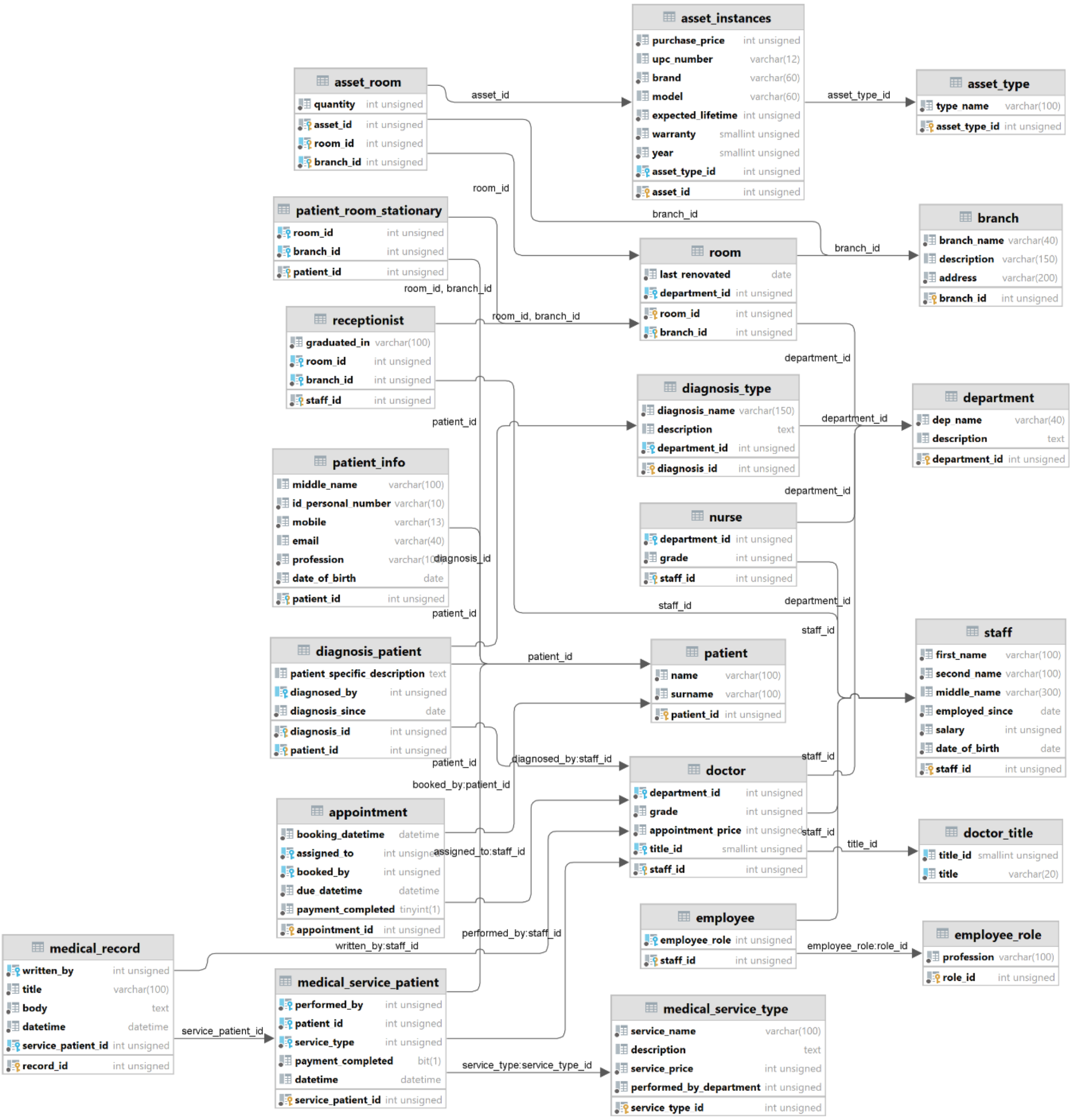


Figure 5. Full UML diagram for the hospital database.

Relational schema

asset_type {[asset_type_id, type_name]}

asset_instances {[asset_id, purchase_price, upc_number, brand, model, expected_lifetime, warranty, year, *asset_type_id*]}

branch {[branch_id, branch_name, description, address]}

department {[department_id, dep_name, description]}

diagnosis_type {[diagnosis_id, diagnosis_name, *department_id*]}

doctor_title {[title_id, title(unique)]}

employee_role {[role_id, profession]}

medical_service_type {[service_type_id, service_name, description, service_price, performed_by_department]}

patient {[patient_id, name, surname]}

patient_info {[*patient_id*, middle_name, id_personal_number, mobile, email, profession, date_of_birth]}

room {[room_id, branch_id, last_renovated, *department_id*]}

asset_room {[*asset_id, room_id, branch_id*, quantity]}

patient_room_stationary {[patient_id, *room_id*, *branch_id*]}

staff {[staff_id, first_name, second_name, middle_name, employed_since, salary, date_of_birth]}

doctor {[*staff_id*, *department_id*, grade, appointment_price, *title_id*]}

appointment {[appointment_id, booking_datetime, *assigned_to*, *booked_by*, due_datetime, payment_completed]}

diagnosis_patient {[*diagnosis_id*, *patient_id*, patient_specific_description, *diagnosed_by*, diagnosis_since]}

employee {[*staff_id*, *employee_role*]}

medical_service_patient {[service_patient_id, *performed_by*, *patient_id*, *service_type*, payment_completed]}

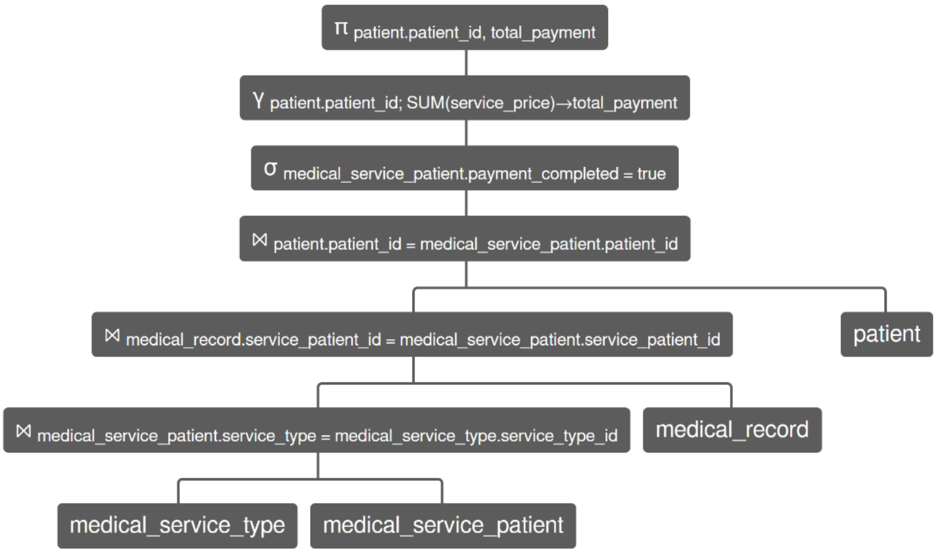
medical_record {[record_id, *written_by*, title, body, datetime, *service_patient_id*]}

nurse {[*staff_id*, *department_id*, grade]}

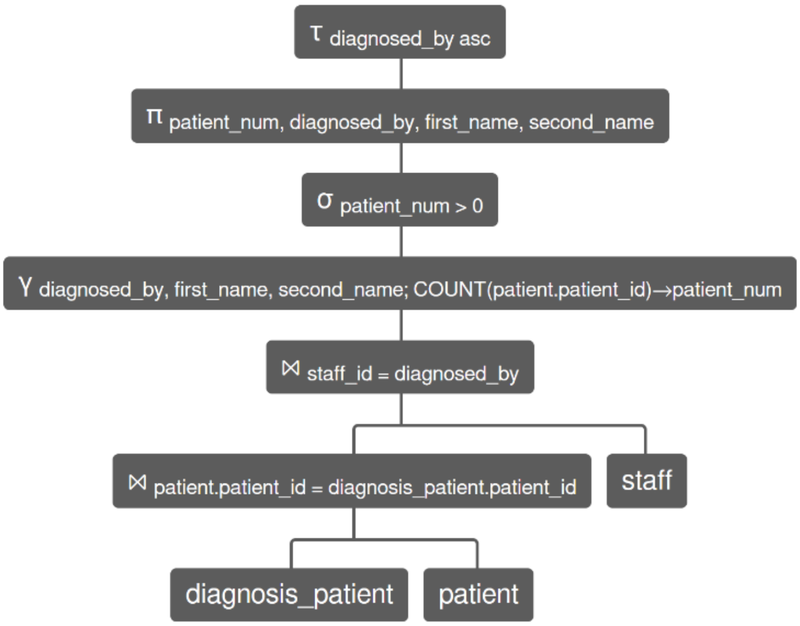
receptionist {[*staff_id*, *room_id*, *branch_id*, graduated_in]}

Relational Algebra (optimized)

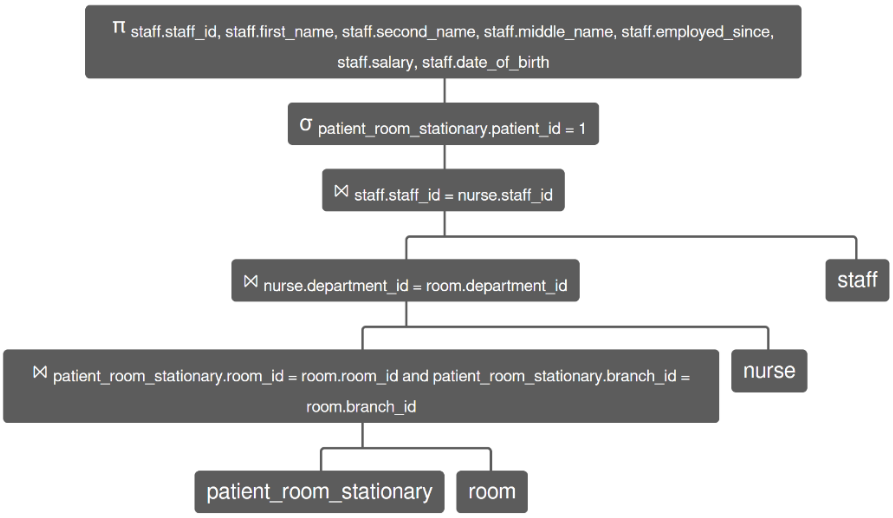
How much money did each patient already pay for services?


π patient.patient_id, total_payment
γ patient.patient_id; SUM(service_price)→total_payment
σ medical_service_patient.payment_completed = true
⋈ patient.patient_id = medical_service_patient.patient_id
⋈ medical_record.service_patient_id = medical_service_patient.service_patient_id
patient
⋈ medical_service_patient.service_type = medical_service_type.service_type_id
medical_record
medical_service_type
medical_service_patient

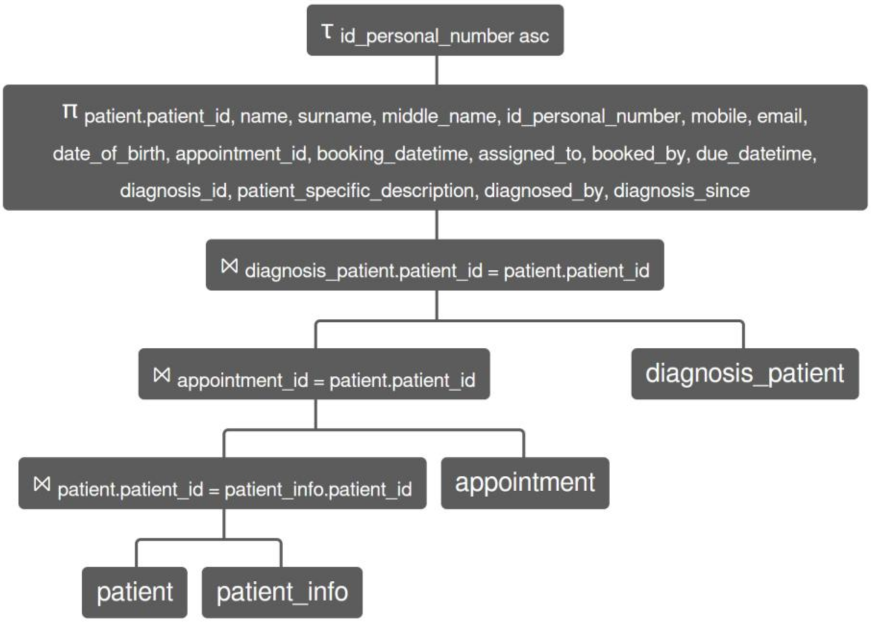
How many patients were diagnosed by each of the doctors?


τ diagnosed_by asc
π patient_num, diagnosed_by, first_name, second_name
σ patient_num > 0
γ diagnosed_by, first_name, second_name; COUNT(patient.patient_id)→patient_num
⋈ staff_id = diagnosed_by
⋈ patient.patient_id = diagnosis_patient.patient_id
staff
diagnosis_patient
patient

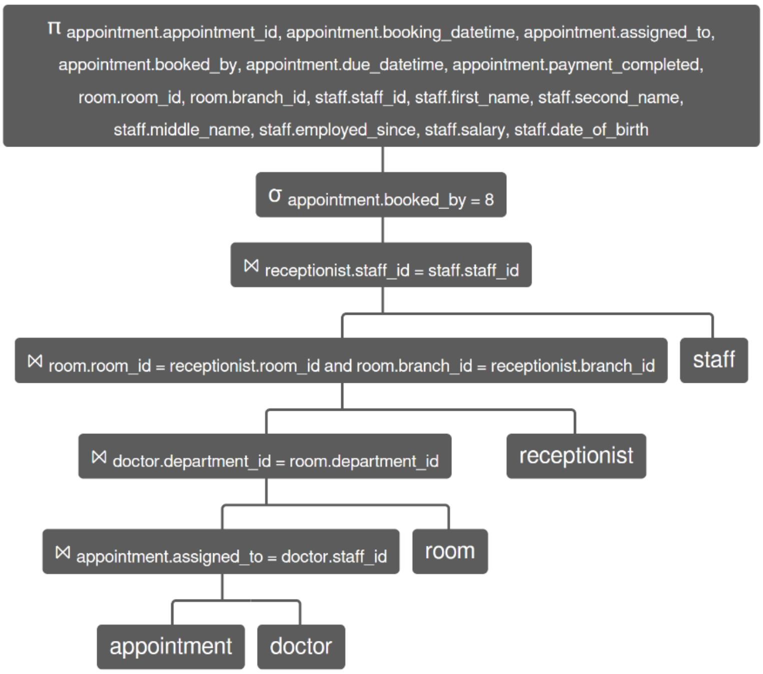
Show nurses at disposition of patient with patient_id:


π staff.staff_id, staff.first_name, staff.second_name, staff.middle_name, staff.employed_since, staff.salary, staff.date_of_birth
σ patient_room_stationary.patient_id = 1
⋈ staff.staff_id = nurse.staff_id
⋈ nurse.department_id = room.department_id
staff
⋈ patient_room_stationary.room_id = room.room_id and patient_room_stationary.branch_id = room.branch_id
nurse
patient_room_stationary
room

Show most important information on patient.


τ id_personal_number asc
π patient.patient_id, name, surname, middle_name, id_personal_number, mobile, email, date_of_birth, appointment_id, booking_datetime, assigned_to, booked_by, due_datetime, diagnosis_id, patient_specific_description, diagnosed_by, diagnosis_since
⋈ diagnosis_patient.patient_id = patient.patient_id
⋈ appointment_id = patient.patient_id
diagnosis_patient
⋈ patient.patient_id = patient_info.patient_id
appointment
patient
patient_info

Show which receptionist(s) to go to for which appointment:


π appointment.appointment_id, appointment.booking_datetime, appointment.assigned_to, appointment.booked_by, appointment.due_datetime, appointment.payment_completed, room.room_id, room.branch_id, staff.staff_id, staff.first_name, staff.second_name, staff.middle_name, staff.employed_since, staff.salary, staff.date_of_birth
σ appointment.booked_by = 8
⋈ receptionist.staff_id = staff.staff_id
⋈ room.room_id = receptionist.room_id and room.branch_id = receptionist.branch_id
staff
⋈ doctor.department_id = room.department_id
receptionist
⋈ appointment.assigned_to = doctor.staff_id
room
appointment
doctor