Appropriate design decisions:

Firstly, we decided to construct our PostgreSQL database using inheritance for employers, but found out that inheritance and foreign keys can not work together effectively
So we decided to get rid of inheritance and construct weak entities using foreign keys and imitating inheritance
We achieved this by creating entity person which has SERIAL as primary key and all attributes normal person has(i.e. ssn, address, small dick etc.), all entities reference this key and has it as primary key, so it is very quick to find out who fucks who.
This decision simplified message system.

Secondly, as doctor and nurse (are different entities that have nothing in common) (usually have different qualifications), we decided to divide table medic_qualifications into two tables: nurse_qualification, doctor_qualifications

Also, we decided that appointment is just meeting with certain doctor and nurse assigned to this doctor. So we deleted opportunity to have several doctors and nurses because, while before we had opportunity to have more doctors and nurses.

In our understanding nurse is a helper of doctor, and usually nurses help several doctors, and doctors have several nurses. We called this relation as assignment.

Orders can contain several medicines and services. Each invoice is based on one order. Order price is calculated from its components price. Prices of services and medicines, as well as their names and descriptions, are in database in tables called service and medicine correspondingly.

Each patient has medical chart. Medical chart is changed dynamically by adding new reports by doctors and nurses.