**DATABASE MANAGEMENT FOR A DENTIST’S POLYCLINIC**

**ABSTRACT**

A Dentists’ Polyclinic has various dental departments such as General Dentistry, Periodontist and Endodontist. The patients can go to all these Departments. The Doctor’s work from Monday to Saturday, 6 hours each day. The patients can be of two types, a regular patient and a new patient. A patient who has been coming for 10 years or more gets a 10% discount (regular patient) . A new patient gets the first check-up free. The patients’ medical history such as heart problems, allergies, other illnesses are stored within the database so that the doctor can give a treatment accordingly. A patient can come to the polyclinic with a dependent.

1. Database Management for a Dentist’s Polyclinic

1)The Polyclinic has the following details stored:

a) Name of the Polyclinic(Primary key)

b) Address

c) Timings (Hours)

d) Phone Number

2) For the Doctors in this Polyclinic, the following details are saved:

a) Name

b) Department Number

c) Department Id

d) Doctor Id (Primary key)

e) Salary Slip Number

f) Salary

g) Phone Number

3) The polyclinic has many types of doctors depending on the dental departments.

1. General dentistry: Cavities, Missing Teeth, Mobile Teeth
2. Endodontist: Rootcanals
3. Periodontist: Gums

4) On Mon-Sat the doctors work.

5) A patient coming can be of two types, a new patient and a regular patient. A regular patient’s history is accessible.

6) The doctors work for 6 hours on any given day.

7) The appointments are given according to the above timings.

8) A patient has the following details saved

a) Name

b) Patient ID (Primary Key)

c) Last date of visit

d) Date of Birth

e) Previous Treatment

9) The Medical history is also saved:

a) Allergies

b) Heart Problems

c) Pain in any tooth/teeth

d) Past Treatments (Discriminator)

e) Other Illnesses

10) Every new patient is assigned a new patient ID , his name and date is saved. The first check-up for the patient is free.

11) Patients that have been coming for the last 10 years, get a 10% discount.

12) Each patient can give the details of one dependents and the following details of the dependents are saved:

a) Name (Discriminator)

b) Phone number

13) Each patient can go to multiple departments in this polyclinic.

**2. Components of Database Design**

**2.1) Entities**

An entity is any object in the system that we want to model and store information about. Every entity has at least one attribute: its primary key. In the Database for a Dentists’ Polyclinic, the entities are Polyclinic, Doctors, Patients. Endodontist, Periodontist and General Dentist are entities of the form of non-disjoint specialization. Regular Patient and New Patient also are entities of the form of non-disjoint specialization.

**2.2) Weak Entities**

A weak entity is one which doesn’t have a primary key of its own. It depends on a strong entity. The discriminator of the weak entity and the primary key of the strong entity on which it depends makes the superkey of the Weak Entity. In the Database for a Dentists’ Polyclinic, Medical History and Dependents are weak entities.

**2.3) Relationships**

A relationship is an association among several entities. In the Database for a Dentists’ Polyclinic, Polyclinic ‘handles’ Patients is an example of a relationship in which ‘handles’ the relationship between the two entities.

**2.4) Attributes**

Attributes constitute a character and explains the characteristics of an entity. Attributes describe the entity or relationship modeled by a table. In the Database for a Dentists’ Polyclinic, an example of attribute is ‘Name’ of Polyclinic. In this, ‘Name’ is used to describe the entity which is Polyclinic. An attribute can be multivalued, derived or composite.

**2.5) Primary Key**

The primary key is the attribute which helps us uniquely identify each record in the database. In the Database for a Dentists’ Polyclinic, Patient\_id is the primary key of Patient because it helps us uniquely identify each Patient record.

**2.6) Foreign Key**

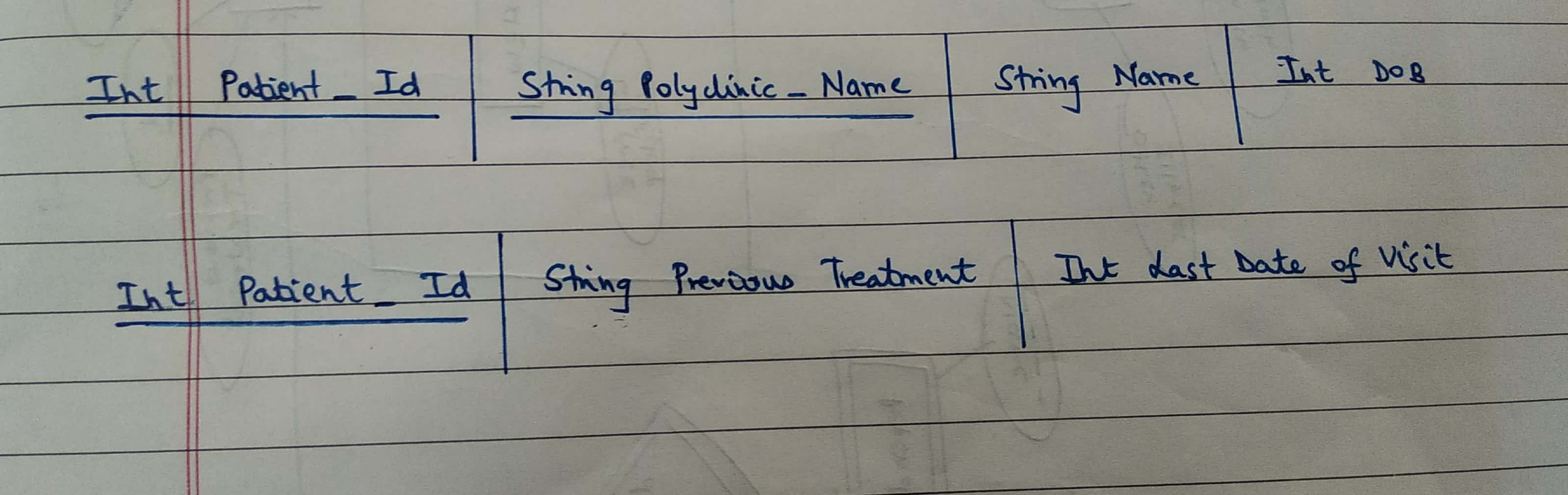
A foreign key is a set of attributes in a referencing relation, such that for each tuple in the referencing relation, the values of the foreign key attributes are guaranteed to occur as the primary key value of a tuple in the referenced relation. In the Database for a Dentists’ Polyclinic, an example of foreign key is Polyclinic\_name is a foreign key in Patient entity.

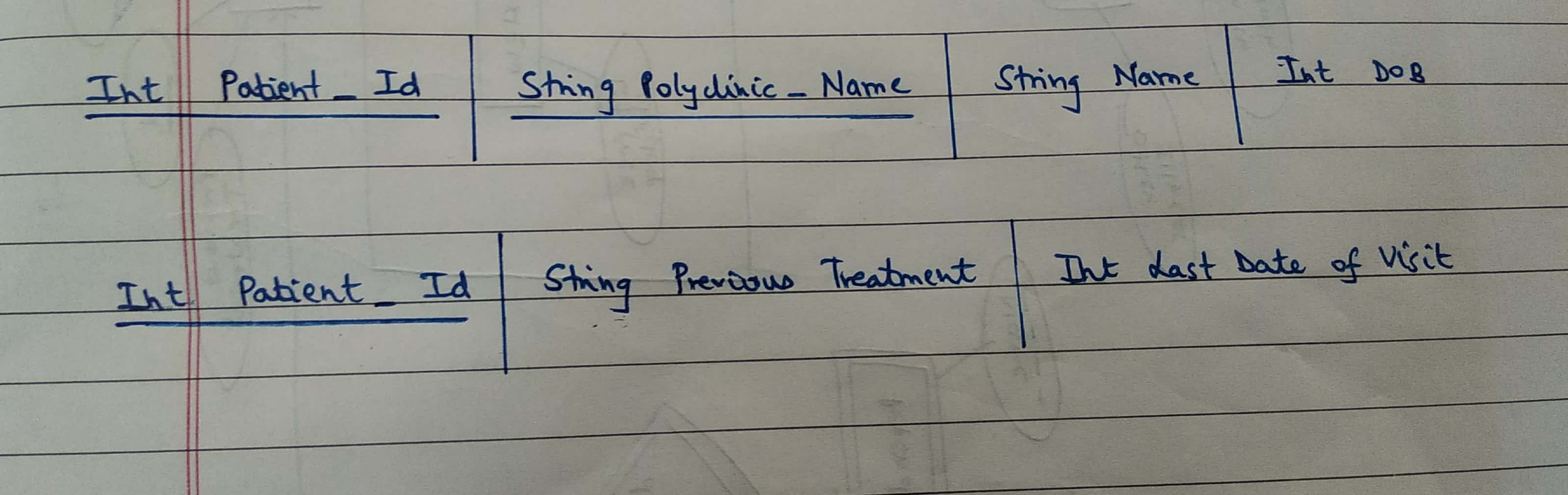
**2.7) Cardinalities**

Cardinalities help us understand the relationship between two entities, whether it is one-one, many-one, one-many, many-many. In the Database for a Dentists’ Polyclinic, an example of cardinality is that ‘many’ Patients go to ‘one’ Polyclinic.

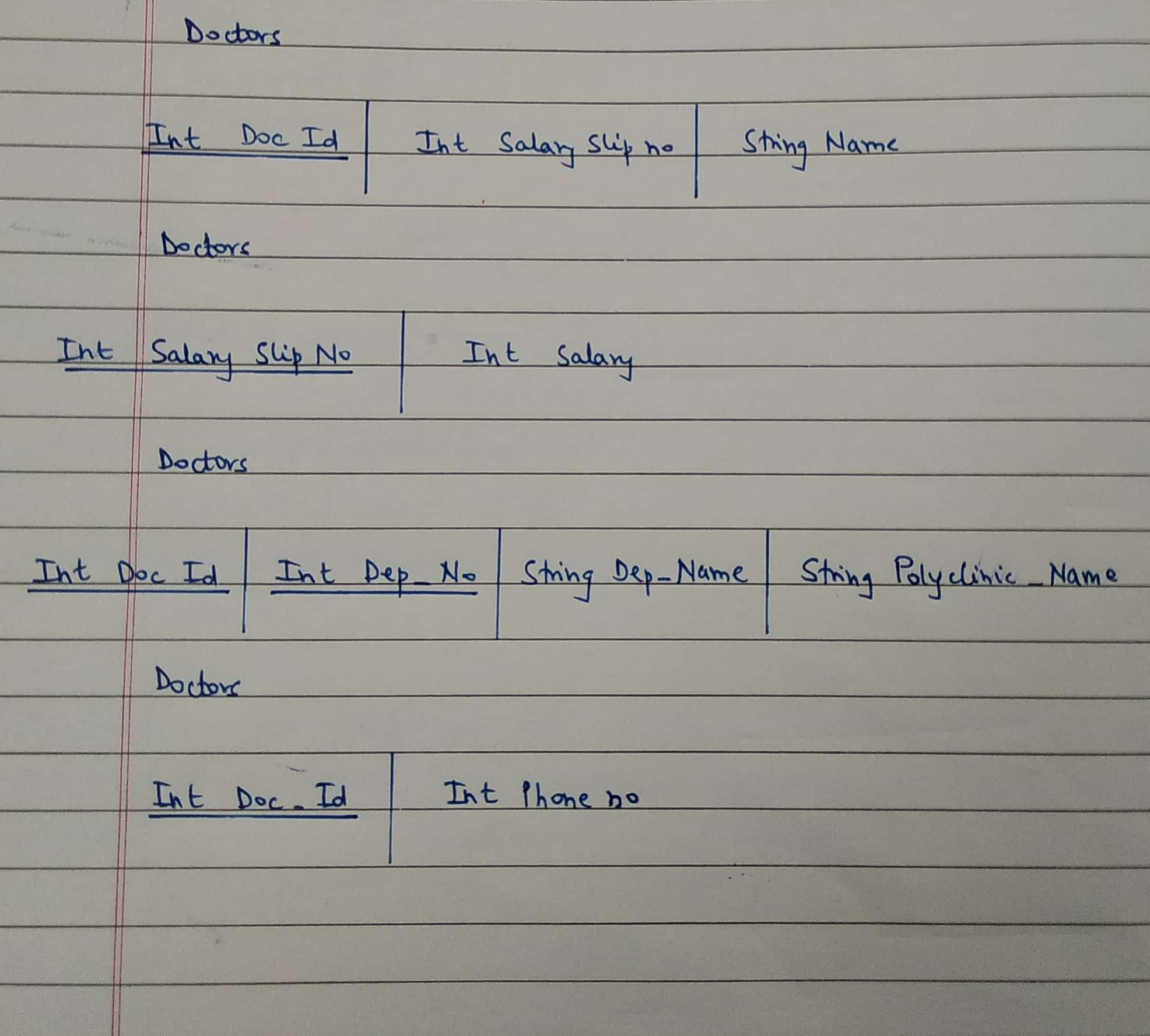
**Normalization**

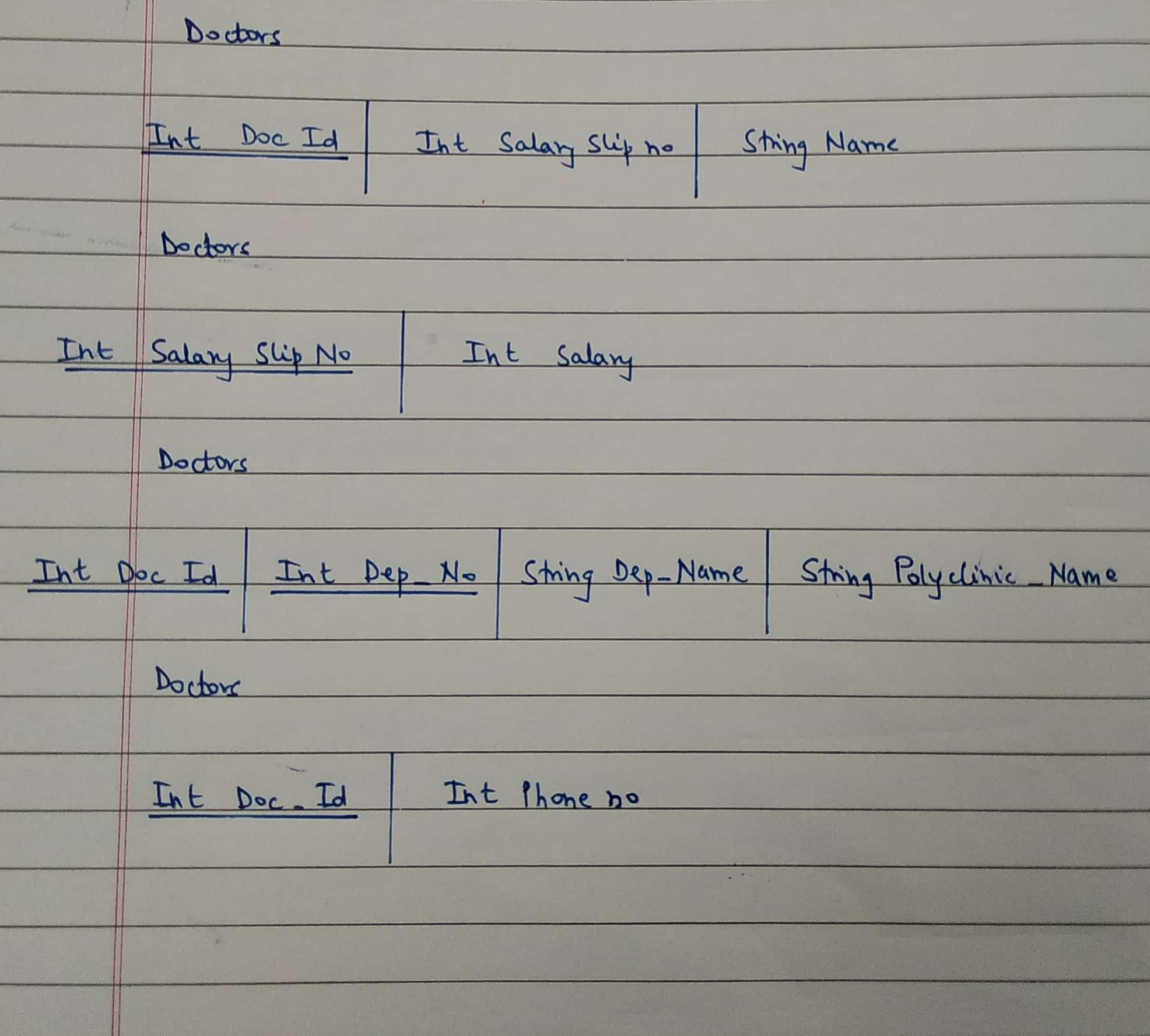
5.1)Patient

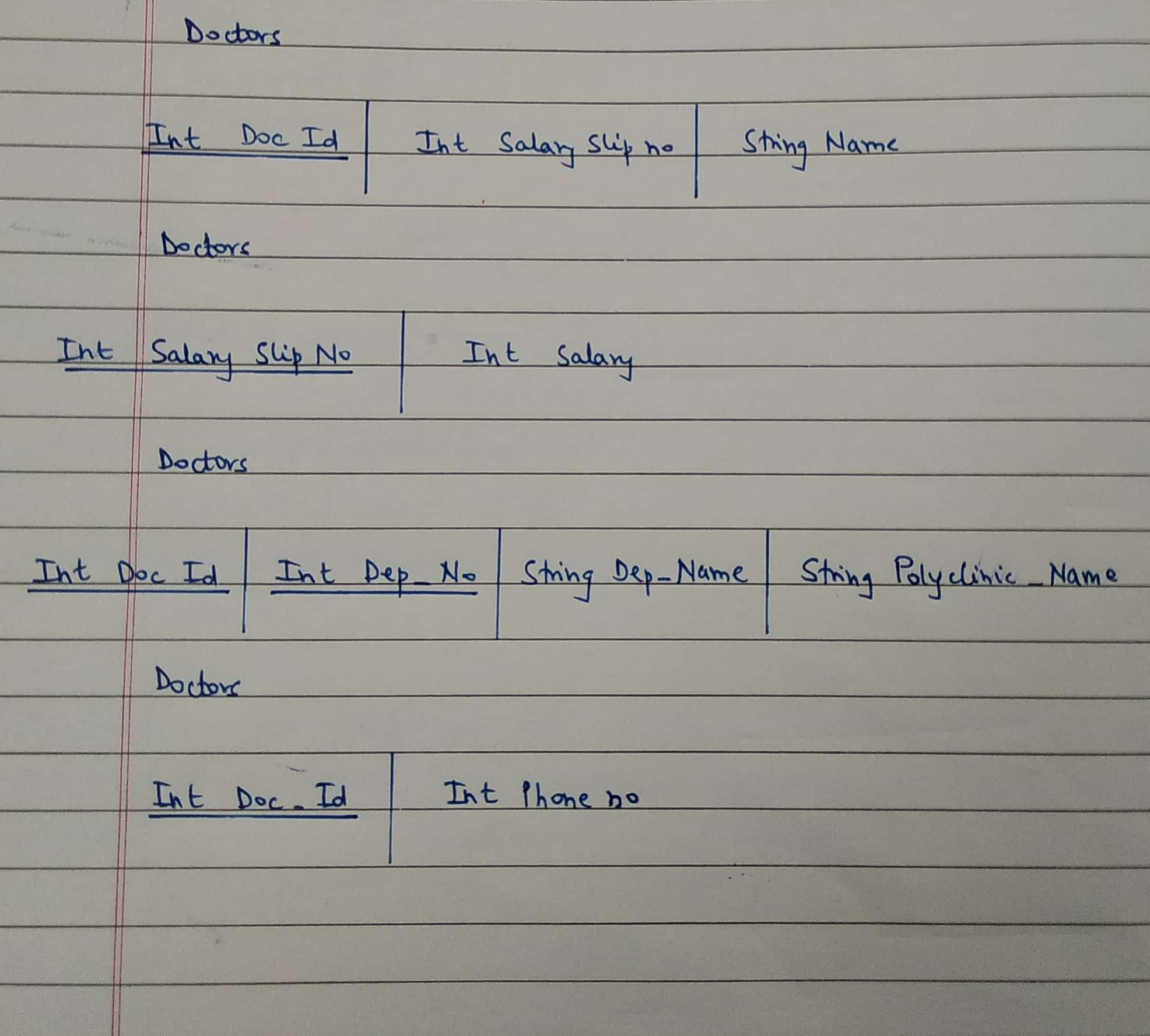


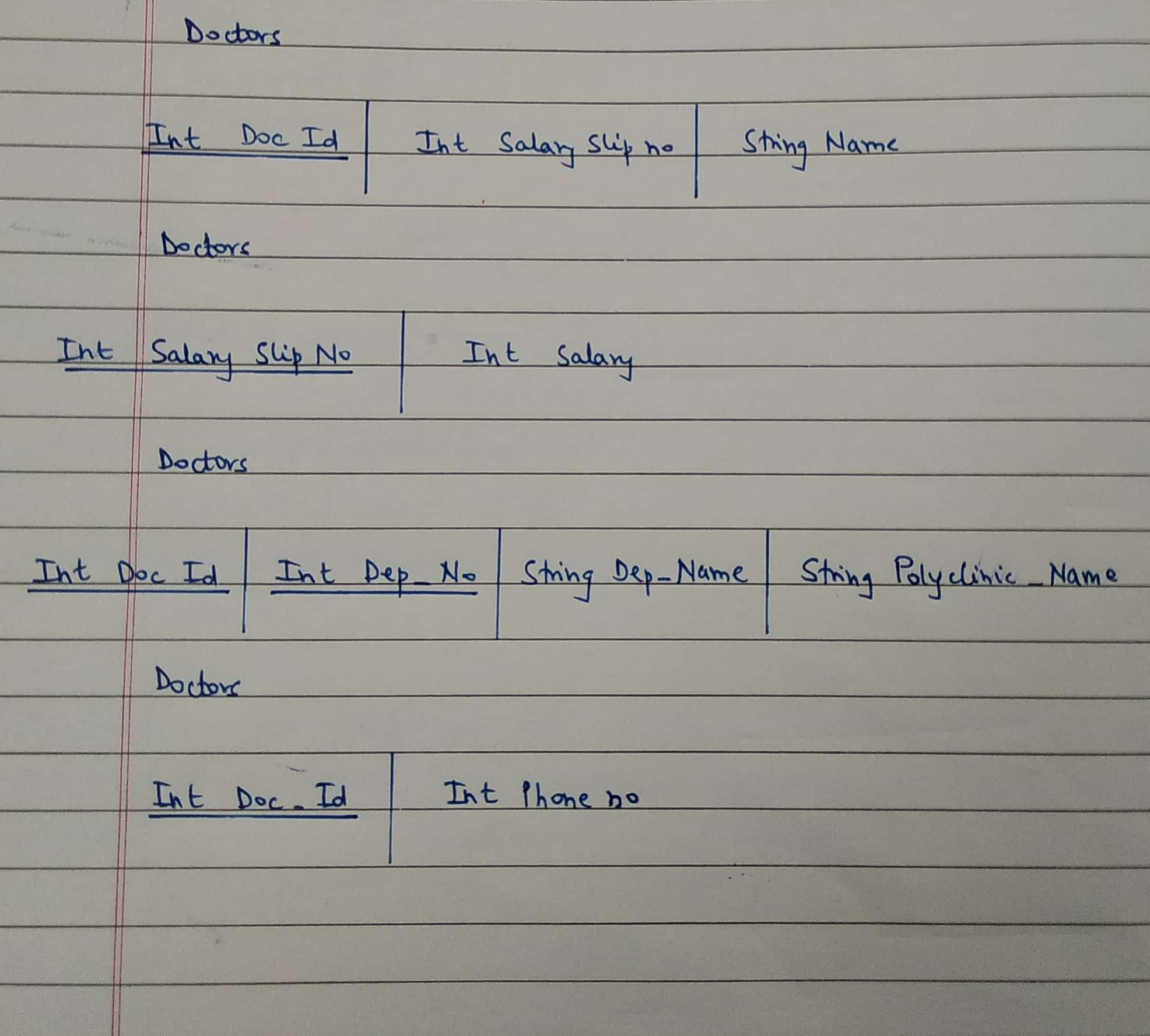


5.2)Doctors









**5.3)Explanation of Normalization**

For a table to be in 1NF, there should be no multivalued attributes.

For a table to be in 2NF, no partial dependency should be present and the table should be in 1NF. Partial Dependency is when:

α🡪β (prime attribute 🡪 non-prime attribute) . This must be removed to get the table in 2NF.

For a table to be in 3NF, no transitive dependency should be present and the table should be in 2NF. Transitive Dependency is when:

α🡪β ( non-prime 🡪 non-prime). This must be removed to get the table in 3NF. For 3NF, in the relation α🡪β, either α should be superkey or β should be prime. Even if one doesn’t satisfy, then it won’t be in 3NF.

For a table to be in BCNF, in α🡪β, α must be superkey.

We decompose the tables in a way such that it is a Lossless form of decomposition attributes of R1 ∪ attributes of R2 = attributes of R or

attributes of R1 ∩ attributes of R2 ≠ null or

attributes of R1 ∩ attributes of R2 🡪 attributes of R1 or

attributes of R1 ∩ attributes of R2 🡪 attributes of R2