Group Ryzen

ARSH,SAMIKA,GURPREET,JAGNOOR

Hospital MANAGEMENT SYSTEM

Contents

[Executive Summary 2](#_Toc127454286)

[Design Analysis Process 2](#_Toc127454287)

[Entity Relationship Diagram 3](#_Toc127454288)

[Table Designs – Data Dictionary 4](#_Toc127454289)

[Contributions 5](#_Toc127454290)

[References 5](#_Toc127454291)

# Executive Summary

**A Hospital Management System (HMS) is essential in today’s healthcare as it helps improve patient care and reduce mistakes. It acts as a central hub that automates various tasks like patient registration, appointment scheduling, and record-keeping. This system streamlines hospital operations, enhances communication between staff, and ensures better overall care for patients. By centralizing and automating processes, an HMS makes healthcare more efficient and effective.**

# Design Analysis Process

**1. Introduction**

This document outlines the system analysis and design process for the healthcare management system, covering the various phases of the project, including user stories, use cases, and design decisions. The system aims to manage patient treatment, billing, and healthcare personnel efficiently.

**2. Analysis Tasks and Evidence**

**2.1 User Stories**

User stories were developed to capture the needs and requirements of the system from the perspective of different users (patients, doctors, nurses, and administrative staff). These stories are crucial in understanding how each user interacts with the system.

* **As a patient**, I want to view my treatment history so that I can track my health progress.
* **As a doctor**, I need to manage and update treatment plans for patients so that I can ensure they receive appropriate care.
* **As a nurse**, I want to assist in administering treatments to patients so that I can help doctors in patient care.
* **As an administrator**, I need to generate bills for treatments so that patients can be charged correctly.

**Justification**: These user stories help ensure that the system is user-centered and meets the needs of all stakeholders involved.

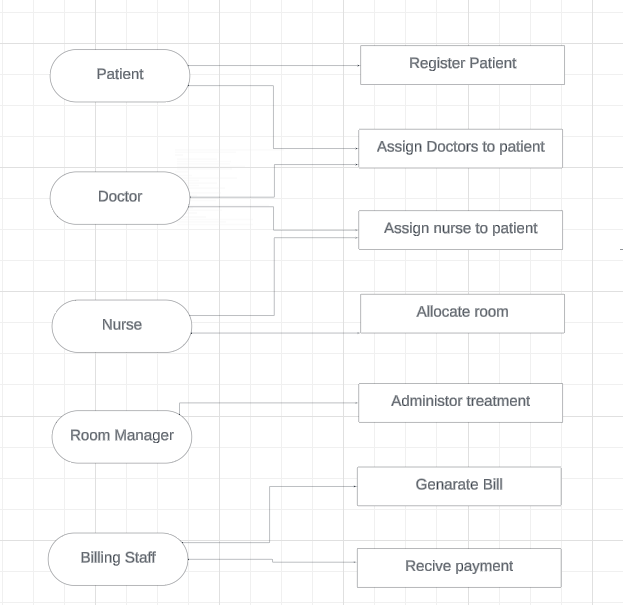
**2.2 Use Cases**

Use cases were derived from user stories and represent specific functionalities the system must support. Each use case was carefully considered to ensure it aligns with the goals of the system.

* **Use Case 1**: Register a new patient in the system.
* **Use Case 2**: Administer treatment to a patient.
* **Use Case 3**: Assist a doctor in a patient’s treatment.
* **Use Case 4**: Assign a room to a patient for treatment.
* **Use Case 5**: Generate a bill for a patient after treatment.
* **Use Case 6**: Update patient treatment history.
* **Use Case 7**: Assign a nurse to assist with treatment.
* **Use Case 8**: Record the completion of a treatment session.
* **Use Case 9**: Track patient’s treatment history.
* **Use Case 10**: Notify the patient about the bill payment.
* **Use Case 11**: Allocate or deallocate rooms based on availability.
* **Use Case 12**: Monitor and update room statuses.

**Justification**: These use cases ensure comprehensive coverage of the system’s functionality and support critical operations required by healthcare staff and patients.

**2.3 Use Case Diagrams**

****

Use case diagrams were created to visualize the interactions between actors (patients, doctors, nurses, administrators) and the system. These diagrams provide a clear overview of the system’s functionalities and their relationships.

**Justification**: Use case diagrams help in identifying system boundaries and key functionalities, making it easier to communicate the system’s design to stakeholders.

**2.4 Functional and Non-Functional Requirements**

* **Functional Requirements**:
  + The system must allow doctors to create and update treatment plans.
  + Nurses should be able to view and assist with treatments.
  + The system must generate bills after treatments.
  + Rooms should be assignable to patients.
* **Non-Functional Requirements**:
  + The system should be accessible 24/7 to accommodate healthcare needs at any time.
  + Data confidentiality and integrity must be ensured to protect patient information.
  + The system should be scalable to support an increasing number of patients and treatments.

**Justification**: Defining these requirements ensures that the system meets both operational needs and quality expectations.

**3. Design and Implementation Decisions**

**3.1 Entity-Relationship Diagram (ERD)**

The ERD, as presented earlier, defines the key entities (Patient, Doctor, Nurse, Room, Bill, Treatment) and their relationships. The cardinalities ensure that the relationships are modeled .realistically, reflecting real-world scenarios in healthcare management.

**Justification**: The ERD is designed to support a normalized database structure, reducing redundancy and ensuring data integrity. This design choice enhances the system’s performance and maintainability.

**3.2 Design Choices and Their Rationale**

* **Normalization**: The decision to normalize the database to the third normal form (3NF) was made to eliminate redundancy and improve data integrity.
* **User-Centered Design**: User stories and use cases were central to the design process, ensuring that the system meets the real needs of its users.
* **Scalability**: The system is designed to be scalable, allowing for easy expansion as the number of patients and treatments grows.

**Why This Implementation is Better**:

* **Efficiency**: The normalized design reduces data duplication, leading to faster queries and updates.
* **Flexibility**: The use cases cover a wide range of functionalities, ensuring that the system is flexible enough to accommodate future needs.
* **Security**: Non-functional requirements emphasize data protection, ensuring that patient information is kept secure.

**3.3 Final System vs. Initial Proposal**

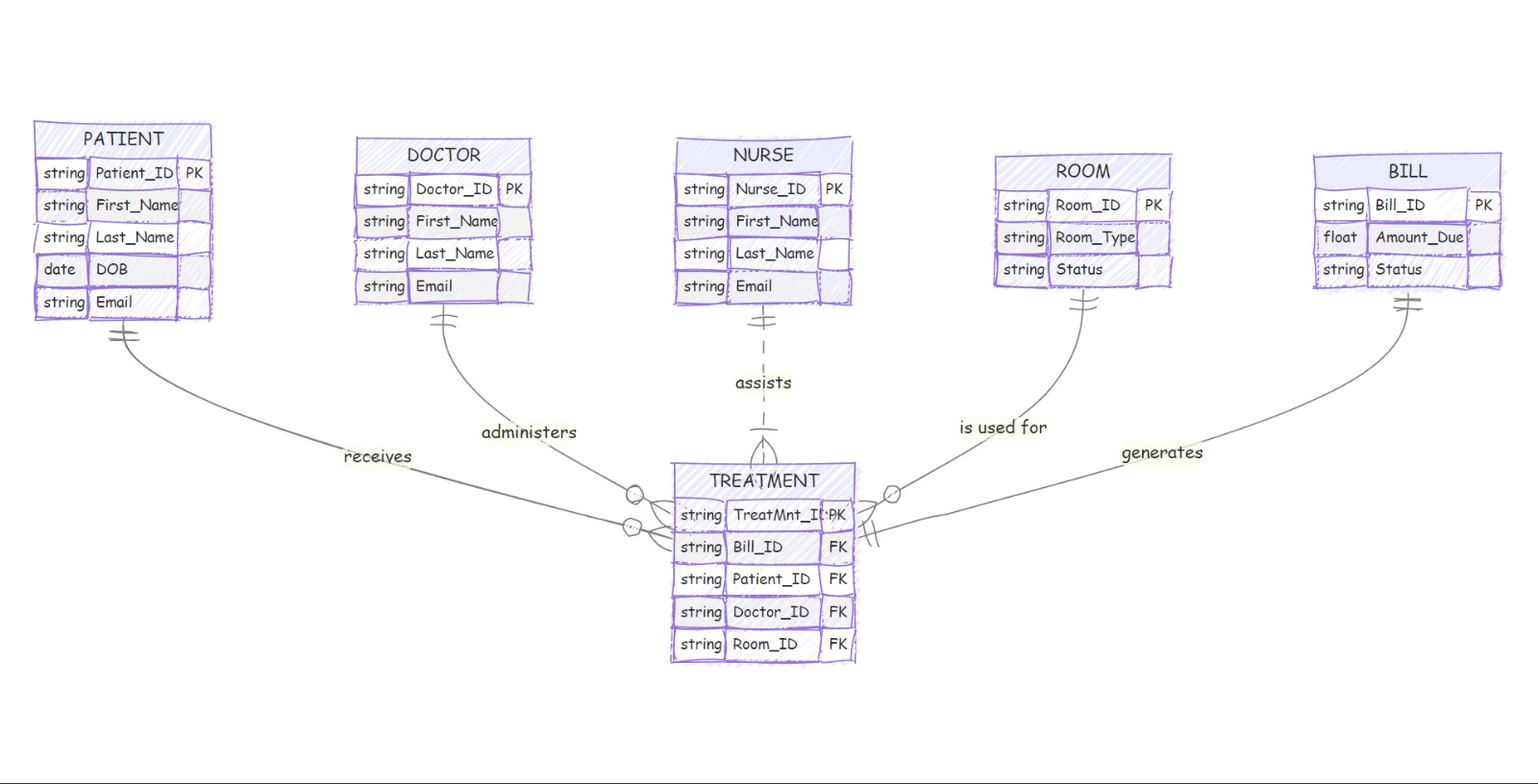
The final system might differ from the initial proposal due to iterative improvements based on feedback from stakeholders and testing. Design changes were made to enhance system usability, performance, or security.

**Justification**: Iterative design allows for continuous improvement, ensuring that the final system is more robust and user-friendly than the initial proposal.

**4. Conclusion**

The design and implementation of this healthcare management system were driven by a user-centered approach, ensuring that all key functionalities are covered while maintaining a focus on performance, scalability, and security. The use of ERD, use cases, and user stories provided a structured approach to system development, resulting in a system that meets the needs of all stakeholders.

# Entity Relationship Diagram



\*Add date& time,diagnoises,treatment(prescription)

1. PATIENT and TREATMENT:

* Relationship: A patient receives treatments.
* Cardinality:
* A patient can have multiple treatments (one-to-many from PATIENT to TREATMENT).
* Each treatment is related to one and only one patient (many-to-one from TREATMENT to PATIENT).

1. DOCTOR and TREATMENT:

* Relationship: A doctor administers treatments.
* Cardinality:
* A doctor can administer multiple treatments (one-to-many from DOCTOR to TREATMENT).
* Each treatment is administered by one doctor (many-to-one from TREATMENT to DOCTOR).

1. NURSE and TREATMENT:

* Relationship: A nurse assists with treatments.
* Cardinality:
* This relationship shows that each treatment can be assisted by multiple nurses, and each nurse can assist in multiple treatments (many-to-many between NURSE and TREATMENT).

1. ROOM and TREATMENT:

* Relationship: A room is used for treatments.
* Cardinality:
* A room can be used for multiple treatments (one-to-many from ROOM to TREATMENT).
* Each treatment takes place in one room (many-to-one from TREATMENT to ROOM).

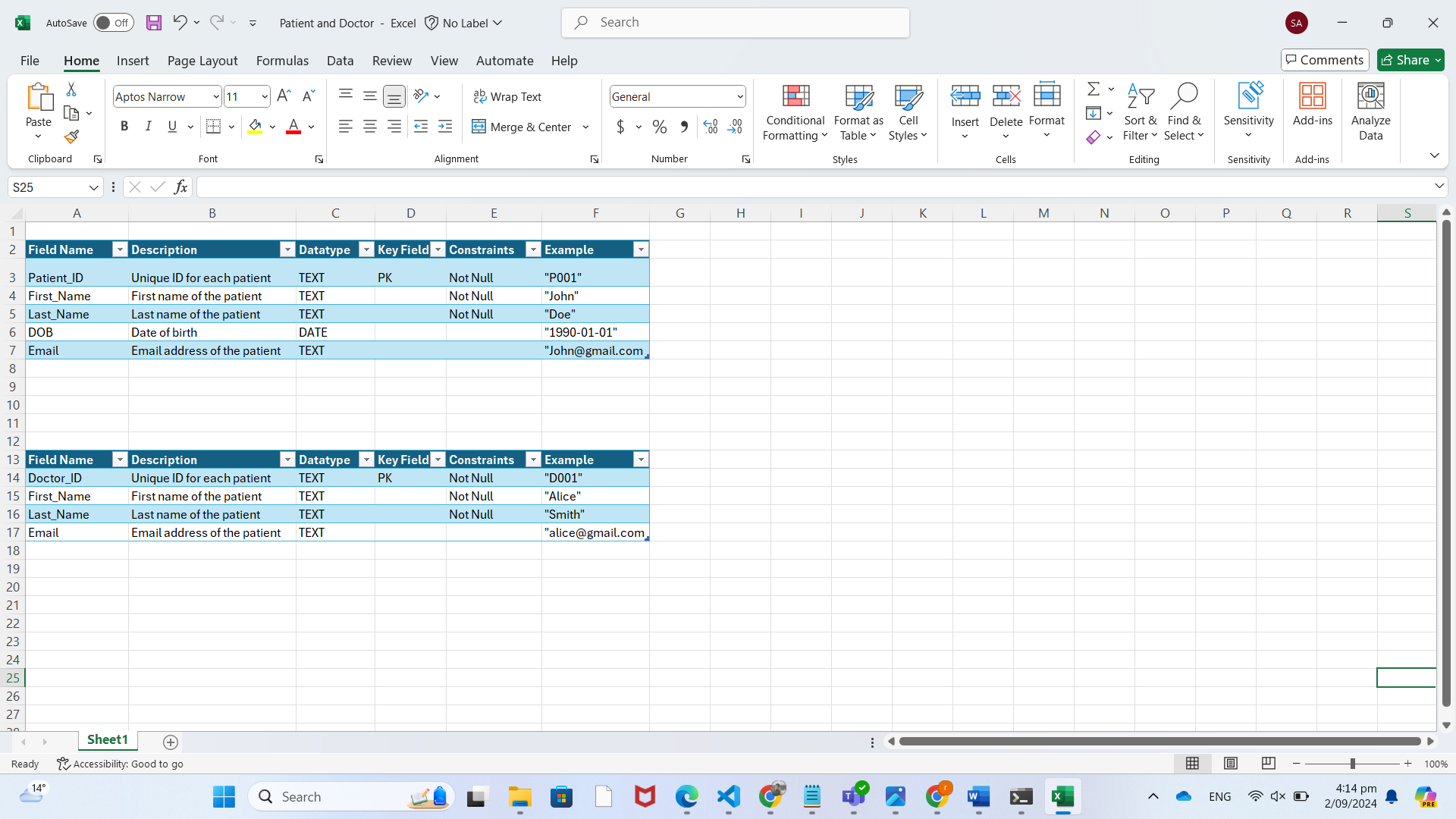
1. BILL and TREATMENT:

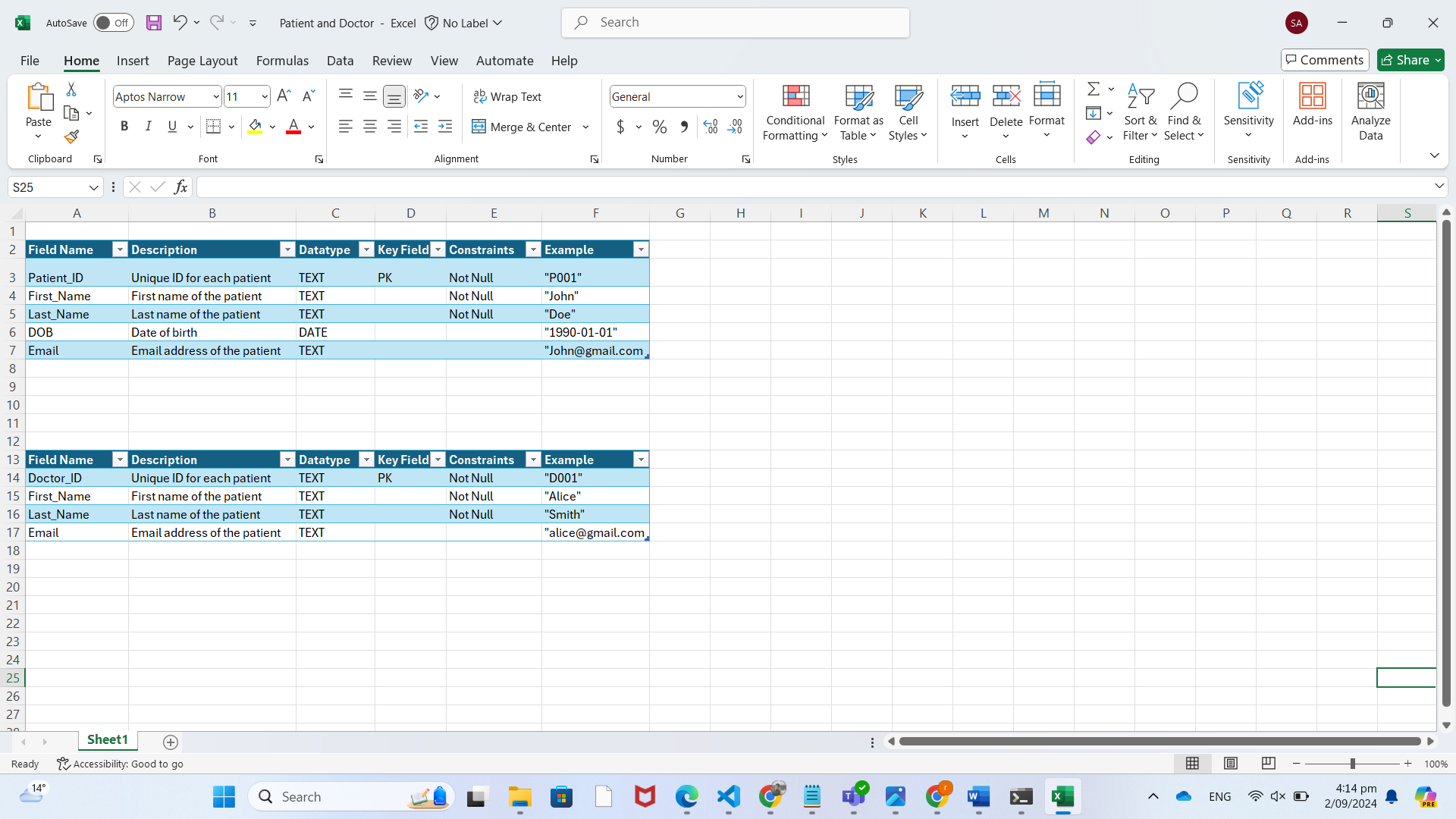
* Relationship: A treatment generates a bill.
* Cardinality:
* A treatment can generate one bill (one-to-one from TREATMENT to BILL).
* Each bill is associated with exactly one treatment (one-to-one from BILL to TREATMENT).

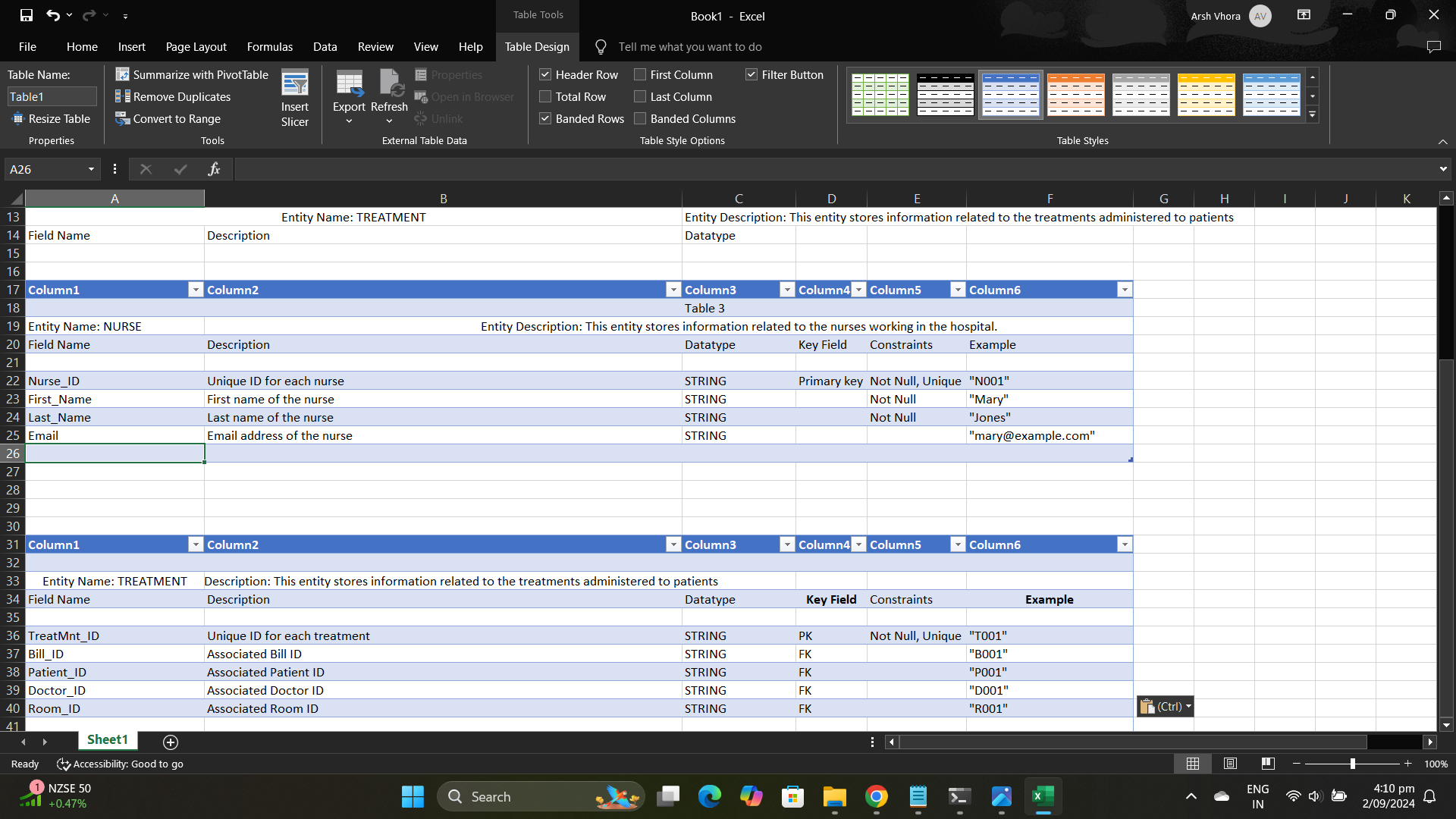
In summary:

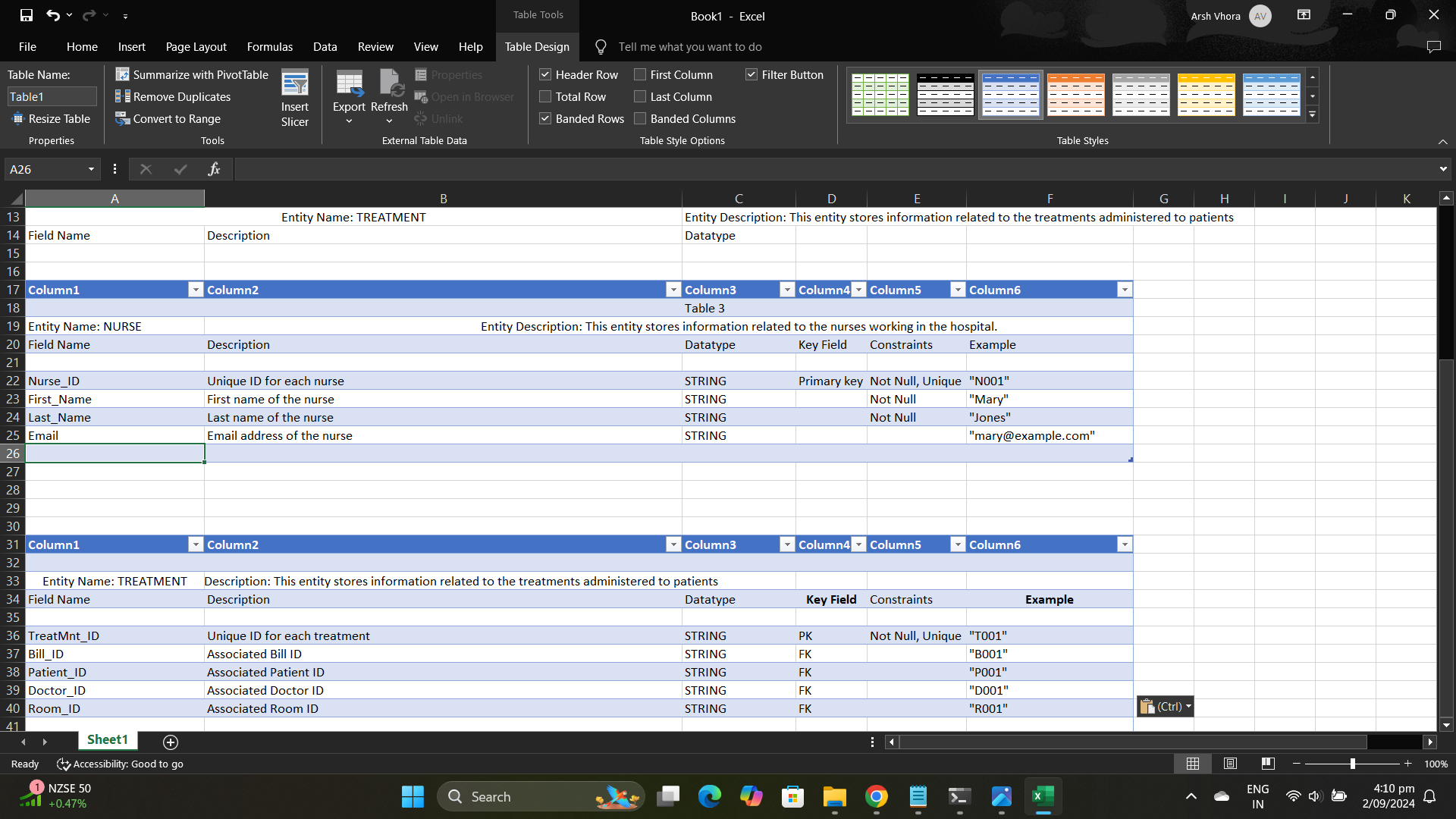
* One-to-Many Relationships: PATIENT, DOCTOR, and ROOM to TREATMENT.
* Many-to-Many Relationship: NURSE and TREATMENT.
* One-to-One Relationship: TREATMENT and BILL.

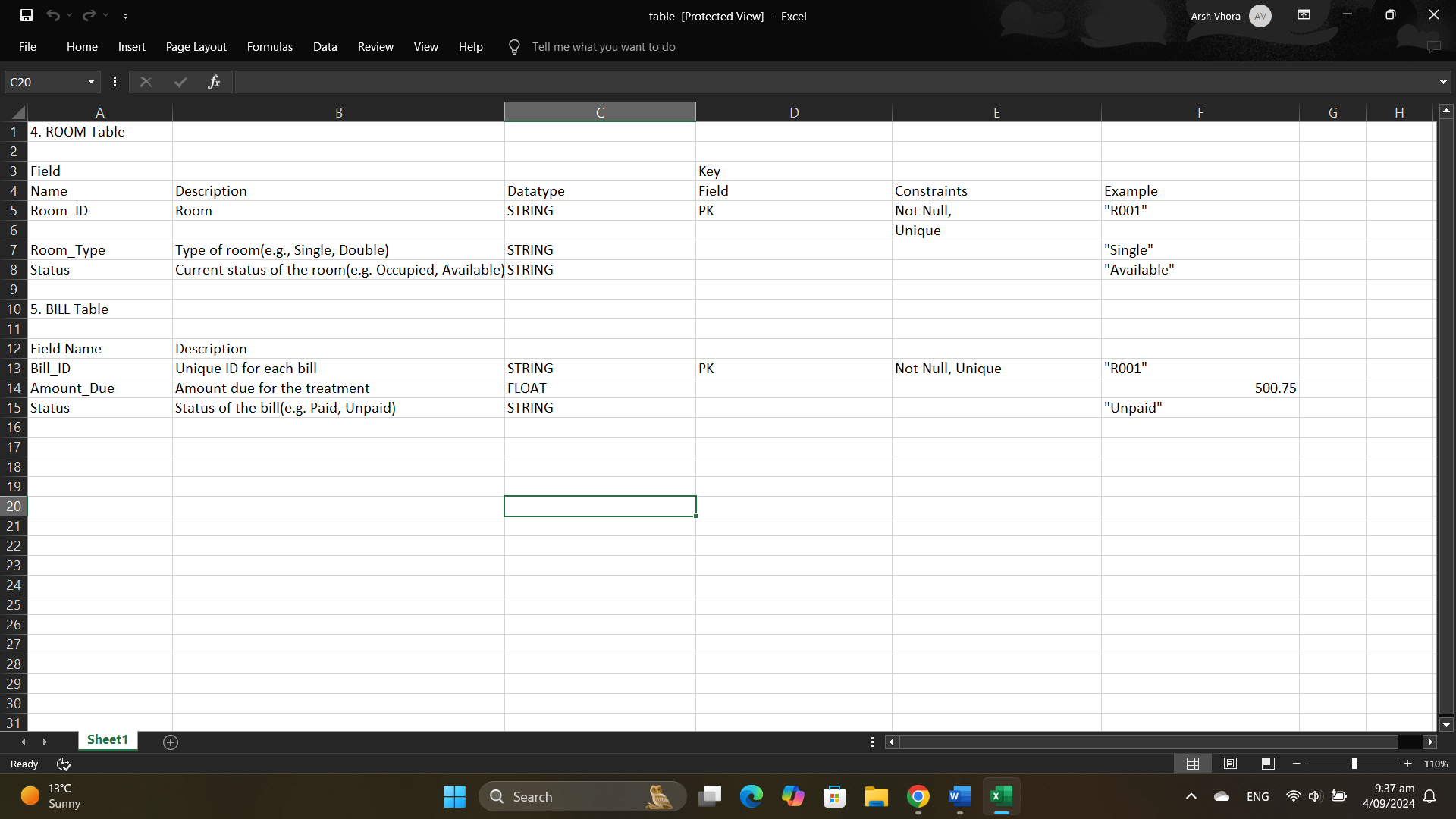
# Table Designs – Data Dictionary











# 

# Contributions

|  |  |
| --- | --- |
| Executive Summary | ARSH |
| Design Analysis Process | ARSH |
| Entity Relationship Diagram | ARSH |
| Table Designs – Data Dictionary | ARSH,SAMIKA & GURPREET |
| Contributions | ARSH |
| References | ARSH |

Also add the link to your GitHub Repository

# References

Add references here if you have any.