



Entity Types

1. Bed as a Strong Entity:

We define Bed as a strong entity because we need to frequently perform independent operations and record information about beds. For example, we need to track details like bed grade, associated equipment, comfort level, dimensions, and bed cost. These attributes are not entirely dependent on the wards, so it makes more sense to consider bed as a strong entity with its own significance. Therefore, we define bedID as the unique identifier for each bed, acting as a unique and complete identifier.

2. PatientAssessment as a Weak Entity:

PatientAssessment is a weak entity that depends on Admission because every patient assessment is inherently linked to a specific admission record, whether it's for an emergency or a planned admission. The assessment, performed by staff to evaluate the patient's condition and make necessary adjustments, occurs after the admission process is initiated. Therefore, PatientAssessment cannot exist independently and must be associated with an admission record to capture the context and details of the evaluation.

Relationship Types

1. Disjoint ISA Relationship for Staff Categorization:

The disjoint ISA relationship is used to categorize Staff into Doctor, Nurse, and Other because each staff member falls into one, and only one, of these categories. This design ensures that a staff member cannot simultaneously belong to multiple categories, reflecting the unique role assignment within the organization. By using a disjoint ISA relationship, we accurately represent the exclusive classification of staff roles and maintain a clear hierarchical structure.

2. Many-to-Many Relationship between OperatingTheatre and Department:

The many-to-many relationship between OperatingTheatre and Department is appropriate because each operating theatre can be utilized by multiple departments, and conversely, each department can have access to multiple operating theatres. This reflects the shared use of operating theatres across departments, with no exclusive ownership by a single department, and the flexibility for departments to use various theatres as needed.

3. One-to-Many Relationship between Admission and Doctor:

The one-to-many relationship between Admission and Doctor is appropriate because each admission can involve a single doctor (CSH doctor) who is directly associated with the patient's care during an emergency admission. For planned admissions, the referring health practitioner is recorded, which could be either a doctor or nurse. However, in an emergency admission, a specific doctor and nurse are involved, but only one doctor is directly associated with each admission. Therefore, each admission is linked to a single doctor, making it a one-to-many relationship where each doctor can oversee multiple admissions.

Attributes

1. NurseID and DoctorID in Admission:

Given that Admission requires recording details of the nurse and CSH doctor involved, we use NurseID and DoctorID in the Admission attributes to clearly specify these roles. Although both NurseID and DoctorID ultimately reference StaffID, using distinct identifiers for nurses and doctors ensures clarity in specifying their respective roles within the admission process. This approach simplifies tracking and managing the specific involvement of each staff member in admissions.

2. Attributes Instead of Entities:

Attributes like DoctorSpecialty, Invoice, NurseWWCC, and AdmissionType are defined as attributes rather than entities because they exhibit weak independence and are strongly dependent on key entities. By treating them as attributes, we simplify the data structure and enhance performance. For instance, DoctorSpecialty and NurseWWCC are closely tied to specific doctors and nurses respectively, and AdmissionType is directly associated with an admission record. Defining these as attributes avoids unnecessary complexity and maintains efficiency in the database design.

Primary Keys

1. Use of IDs as Primary Keys:

We use numeric IDs as primary keys because they are more efficient for indexing and reduce information redundancy. Numeric keys streamline data management and improve performance by providing a compact and consistent way to uniquely identify records.

2. DepartmentID as Primary Key:

Using departmentID as the primary key for Department is preferred because numeric identifiers are more efficient for indexing and searching within the database compared to string values. Although each department has a unique name, departmentID provides a reliable and efficient way to uniquely identify and reference departments across the entire ER model. The name attribute is retained to allow for meaningful classification and human-readable identification of departments, ensuring clarity and usability.

Constraints

1. Foreign key constraints

foreign key constraints create essential links between entities like Admission, Staff, and PatientAssessment, ensuring that assessments and admissions are always tied to valid entities in the system. Additionally, check constraints are used to restrict certain attributes, such as injury severity and staff proficiency levels,

Design Specialities

1. Weak Entity Relationships

Using weak entities (like PatientAssessment) that depend on other entities (like Admission) to ensure that assessments are always tied to a valid admission record.

2. Inheritance for Staff Roles

The way you have separated doctors, nurses, and other staff roles using inheritance, which allows you to maintain common staff details while also allowing each role to have its own specific attributes (like specialty or certification).