Database Management Systems

Subject Teacher: Zartasha Baloch

Entity-relationship model

Lecture # 4

Disclaimer: The material used in this presentation to deliver the lecture i.e., definitions/text and pictures/graphs etc. does not solely belong to the author/presenter. The presenter has gathered this lecture material from various sources on web/textbooks. Following sources are especially acknowledged:

1. Connolly, Thomas M., and Carolyn E. Begg. *Database systems: a practical approach to design, implementation, and management.* Pearson Education, 2005.

2. 2. Hoffer, Jeffrey A., Venkataraman Ramesh, and Heikki Topi. *Modern database management.* Upper Saddle River, NJ: Prentice Hall,, 2011.

In this Lecture you will Learn about:

▶ **What is E-R Modeling**

▶ **What is Entity, Attribute & Relationship**

E-R Model

▶ An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and can be used as the foundation for a relational database.

▶ The Entity-Relationship (ER) data model allows us to describe the data involved in a real-world enterprise in terms of objects and their relationships and is widely used to develop an initial database design.

▶ It provides useful concepts that allow us to move from an informal description of what users want from their database to a more detailed, precise description that can be implemented in a DBMS.

E-R Model

▶ An entity-relationship diagram (ERD) has following main components.

> ▶ **ENTITY**

> ▶ **ATTRIBUTE**

> ▶ **RELATIONSHIP**

▶ **ENTITY,** An entity can be a real-world object, either animate or inanimate, that can be easily identifiable. For example, in a school database, students, teachers, classes, and courses offered can be considered as entities. Entities are represented by rectangle.

▶ **WEAK ENTITY** An entity whose existence depends on some other entity.

▶ *weak entity* must define by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

E-R Model

What Should an Entity be?

- ▶ Should be:
  - ▶ An object that will have many instances in the database
  - ▶ An object that will be composed of multiple attributes
  - ▶ An object that we are trying to model

- ▶ Should not be:
  - ▶ A user of the database system
  - ▶ An output of the database system (e.g., a report)

E-R Model

| Entity | Weak Entity | Attribute |

- **ATTRIBUTE,** the properties or characteristics of an entity are called **attributes**. All attributes have values.
  - A key attribute is the unique, distinguishing characteristic of the entity.
  - Attributes are represented by ovals.
  - Attribute can be,
  - *Composite attribute*, that can be further subdivided.
    - For example address can be divided into city, street, state.

E-R Model (cont'd)

Multivalued attribute that have more than one value. For example, an employee entity can have multiple skill values.

Multivalued Attribute

Derived Attribute

Derived attribute, the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average salary in a department should not be saved directly in the database, instead it can be derived. For another example, age can be derived from date_of_birth.

E-R Model (cont'd)

RELATIONSHIP, The association among entities is called a relationship. A relationship is how the data is shared between entities.

For example, an employee **works at** a department, a student **enrolls** in a course. Here, **Works at** and **enrolls** are called relationships.
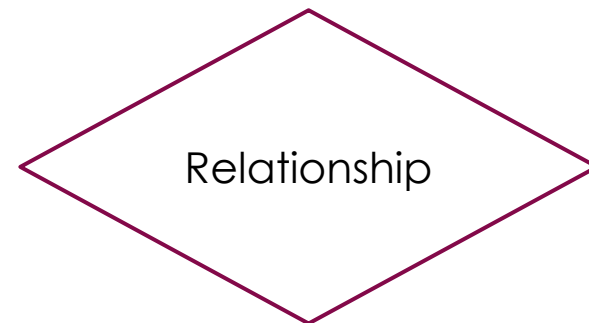
Relationships are represented by diamond shapes.

Degree of Relationship, The number of participating entities in a relationship defines the degree of the relationship.

Unary   = degree 1

Binary   = degree 2

Ternary = degree 3

Relationship

Cardinality

Cardinality, it defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.

The three main cardinal relationships are,

One-to-One (1:1). One entity from entity set A can be associated with at most one entity of entity set B and vice versa.

For example, if each customer in a database is associated with one mailing address.

A   B
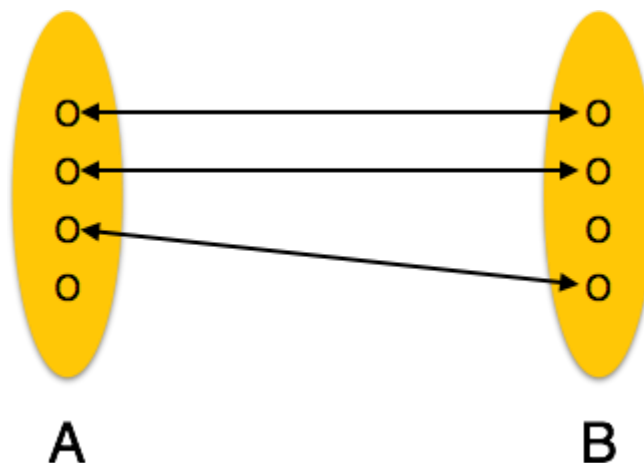
Cardinality

One-to-Many (1:M). One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.

For example, a single customer might place an order for multiple products. The customer is associated with multiple entities, but all those entities have a single connection back to the same customer.

A   B
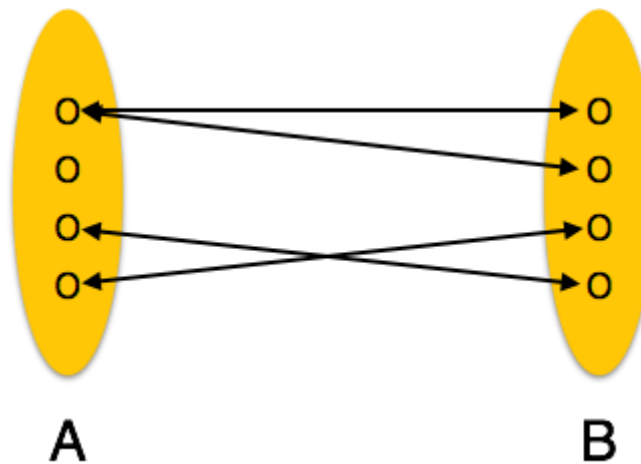
Cardinality

Many-to-Many (M:N). One entity from A can be associated with more than one entity from B and vice versa.
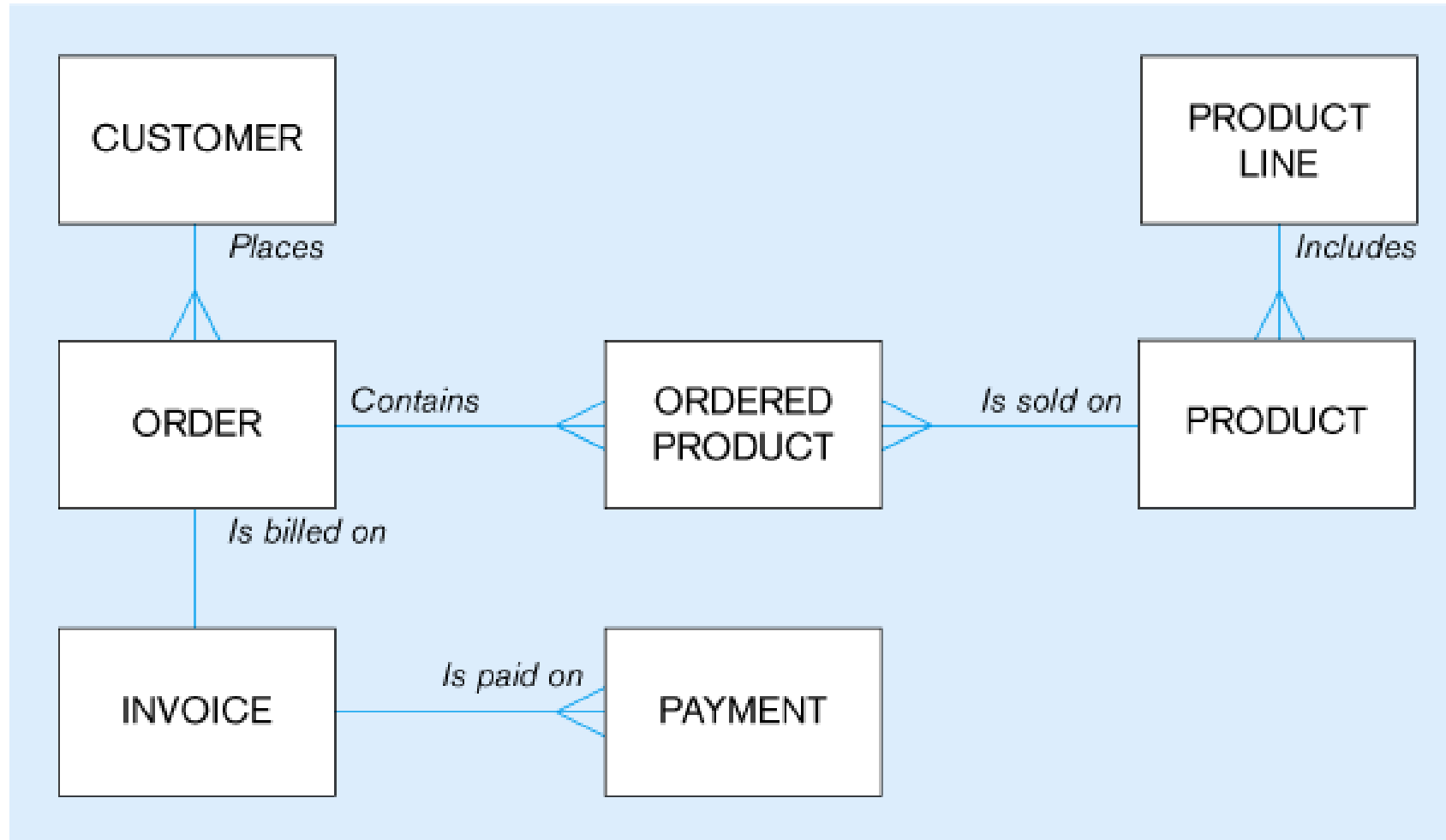
For example, at a company where all call center agents work with multiple customers, each agent is associated with multiple customers, and multiple customers might also be associated with multiple agents.

A   B
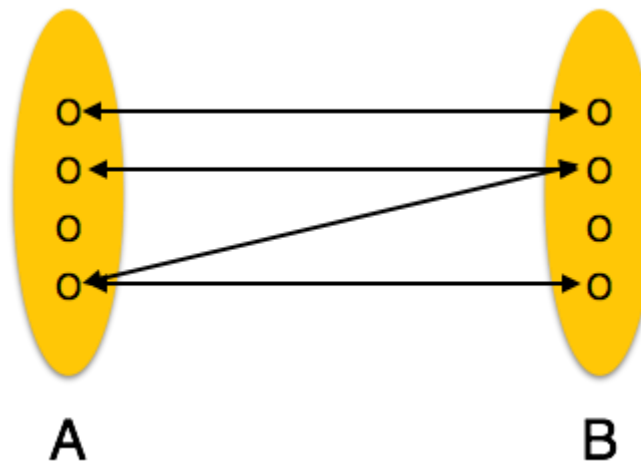
Cardinality
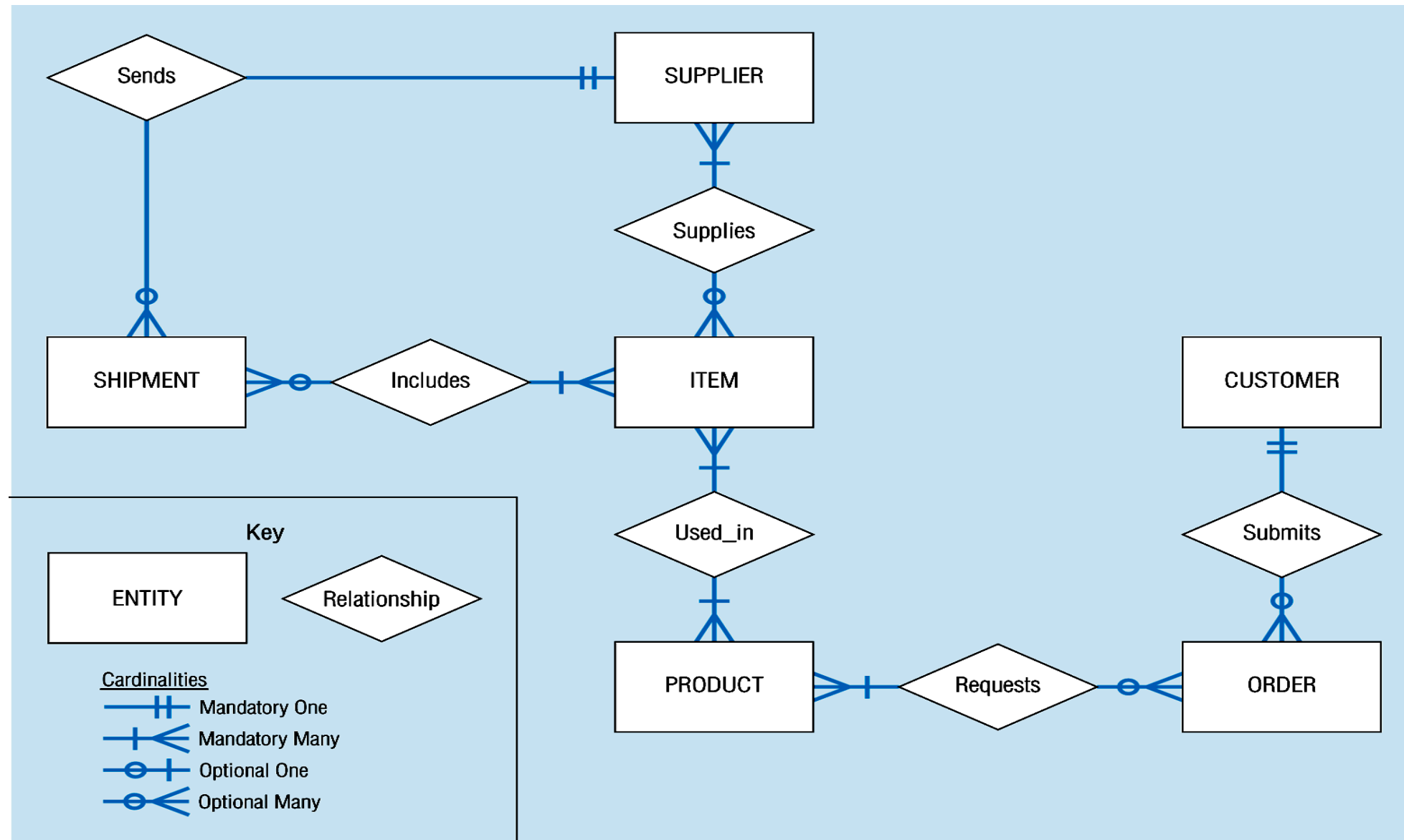

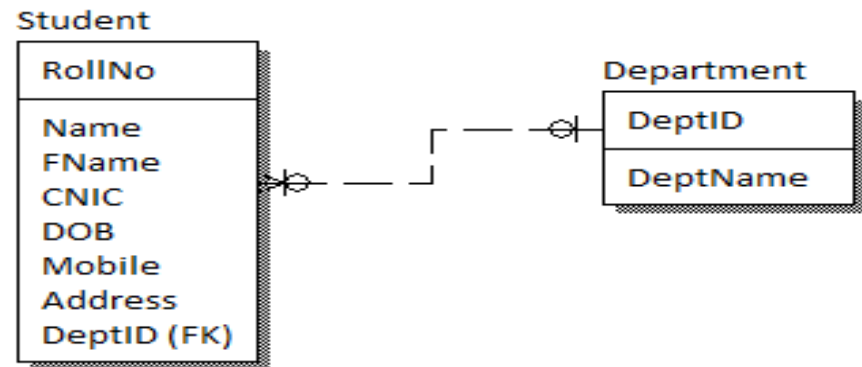|	
⊢	One
⟨	Many
⊦⊦	One (and only one)
○⊢	Zero or one
⟨⊦	One or many
○⟨	Zero or many

Cardinality Notations, can be optional or mandatory.

Simple E-R Model of Customer & Product

E-R Model

Student Department E-R Model

Student
RollNo
Name
FName
CNIC
DOB
Mobile
Address
DeptID (FK)

Department
DeptID
DeptName

Doctor Patient E-R Model

Ward
WardNumber
WardName

Bed
BedNumber
WardNumber (FK)
PatientID (FK)
NurseID (FK)

Doctor
DoctorID
DoctorName
CNIC
Mobile
WardNumber (FK)

Nurse
NurseID
NurseName
CNIC
Mobile

Shift
ShiftID
ShiftName
StartTime
EndTime

Patient
PatientID
PatientName
CNIC
Mobile
DoctorID (FK)
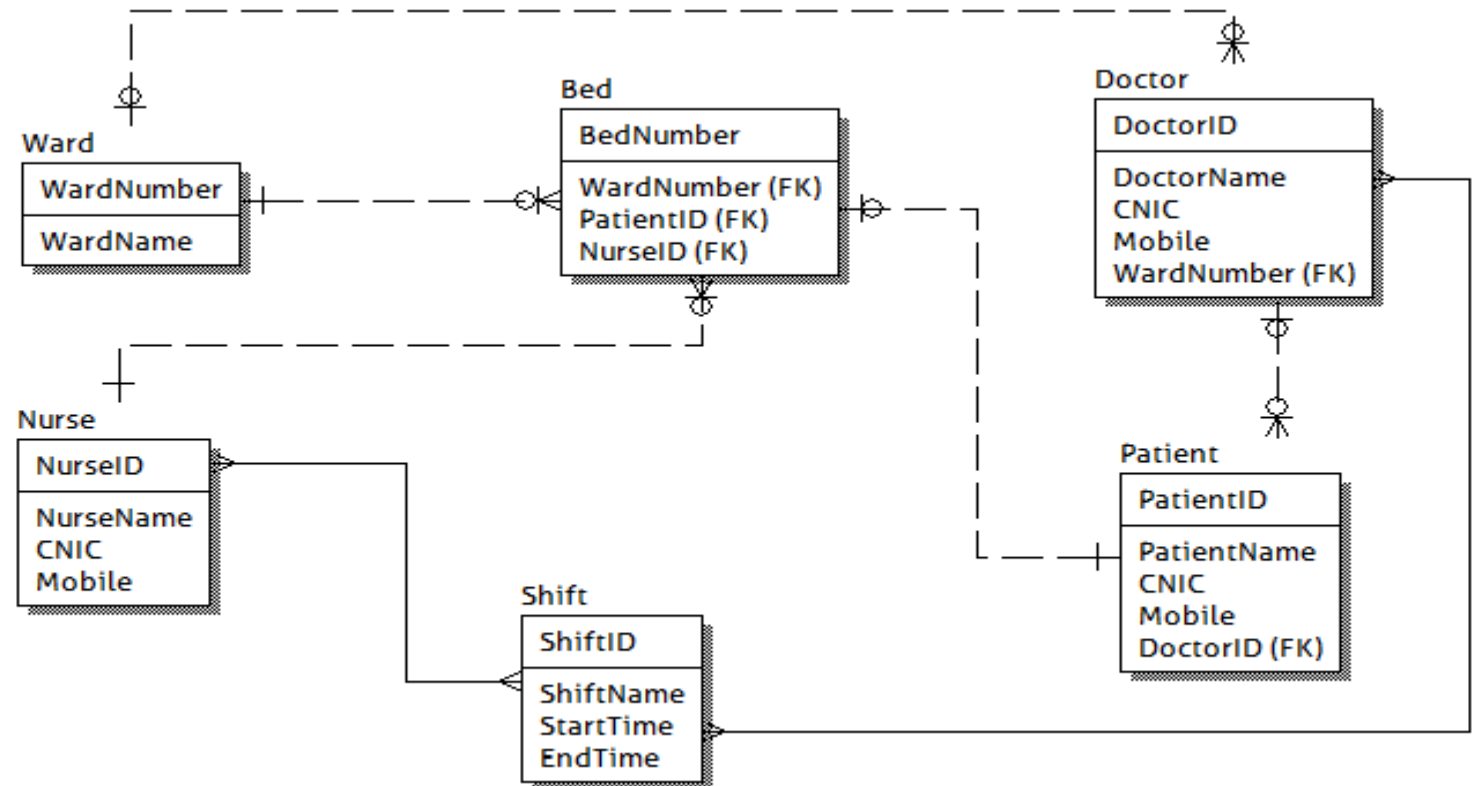
Database keys

DATABASE KEYS

Primary Key, It is a candidate key that is chosen by the database designer to identify entities with in an entity set. Primary key is the minimal super keys. In the ER diagram primary key is represented by underlining the primary key attribute. Ideally a primary key is composed of only a single attribute. But it is possible to have a primary key composed of more than one attribute.

Foreign Key, Fields in a table that refer to the primary key in another table.

The data in this field must exactly match data contained in the primary key field.

Composite Key, Composite key consists of more than one attributes.