Basic Tasks

1.

Animals
├── Primates
│ ├── Monkeys
│ │ └── Humans (as humans are a type of monkey within the primate family, though this classification is debated and often simplified for clarity)
│ └── Other Primates (to represent primates that are not monkeys or humans)
└── Other Animals

Software System
├── Operating System
│ ├── Linux
│ └── Windows (representing two different types of operating systems)
├── Database Management System
│ └── Oracle (Oracle is a specific implementation of a database management system, but MySQL could also be included here as another example, though it's not explicitly mentioned alongside Oracle in the initial list)
│ └── MySQL (as an additional example, even though not listed directly with Oracle)
└── Web Browser / Other Software
└── Chrome (Chrome is a web browser, which falls under the broader category of "Other Software" or could be further specified if needed. It's not directly an operating system or database management system.)

2.

a. Entity: An entity represents a fundamental object, concept, or event that is of interest in the real world and that is modeled in the database. Entities can be tangible (e.g., a person, a book) or intangible (e.g., a transaction, a process). In the ER model, an entity is usually represented by a rectangle with its name inside.

b. Entity Type: An entity type is a generalization or classification of entities that share similar attributes and relationships. For example, "Person" is an entity type that represents all instances of people in the real world. Entity types define the structure of the entities that belong to them, specifying their attributes and relationships with other entities.

c. Entity Instance: An entity instance is a specific occurrence of an entity type in the real world.

For example, "John Doe" is an entity instance of the "Person" entity type. An entity instance represents a single, concrete object or concept that exists within the scope of the database.

d. Optional: In the context of relationships between entities, the term "optional" refers to a relationship that is not required for every instance of an entity type. In other words, some instances of an entity type may participate in the relationship, while others may not. This is often indicated in an ER diagram with a dashed line (or a specific notation) connecting the related entities, signifying that the relationship is optional.

e. Mandatory: In contrast to optional, a mandatory relationship is one that must exist for every instance of an entity type. In other words, all instances of the entity type must participate in the relationship. This is indicated in an ER diagram with a solid line (or a specific notation) connecting the related entities, signifying that the relationship is mandatory.

f. Cardinality: Cardinality refers to the number of instances of one entity type that can be related to a specific instance of another entity type in a given relationship. Cardinality constraints define the minimum and maximum number of related instances allowed. For example, a relationship between "Employee" and "Department" might have a cardinality constraint indicating that each employee must belong to exactly one department (a 1:1 or 1:N relationship, depending on whether a department can have multiple employees). Cardinality is often represented in ER diagrams using specific notation, such as a crow's foot symbol, to indicate the number of instances that can be related.

3.

Data Model: A data model is a conceptual representation of data, data relationships, and data constraints. It is an abstract and structured way of representing entities, attributes, and relationships between them that exist in the real world and need to be stored and manipulated in a database. A data model serves as a blueprint for the design and construction of a database and guides the process of transforming real-world information into a form that can be efficiently stored, retrieved, and processed by a computer system.

Three Reasons Why Data Modelling is Important:

Understanding and Communication: Data modelling helps stakeholders, including database designers, developers, and business analysts, to understand the data requirements of a system. It provides a common language and framework for discussing data structures, relationships, and constraints. This enables better communication and collaboration among team members, leading to more effective and efficient database design.

Data Integrity and Consistency: A well-designed data model ensures that data is stored consistently and accurately. It defines data types, constraints, and relationships that enforce rules for data integrity, such as ensuring that no two entities have the same identifier (primary key constraint) or that certain values are not allowed (check constraint). This helps to maintain the accuracy and reliability of the data, reducing the risk of errors and inconsistencies.

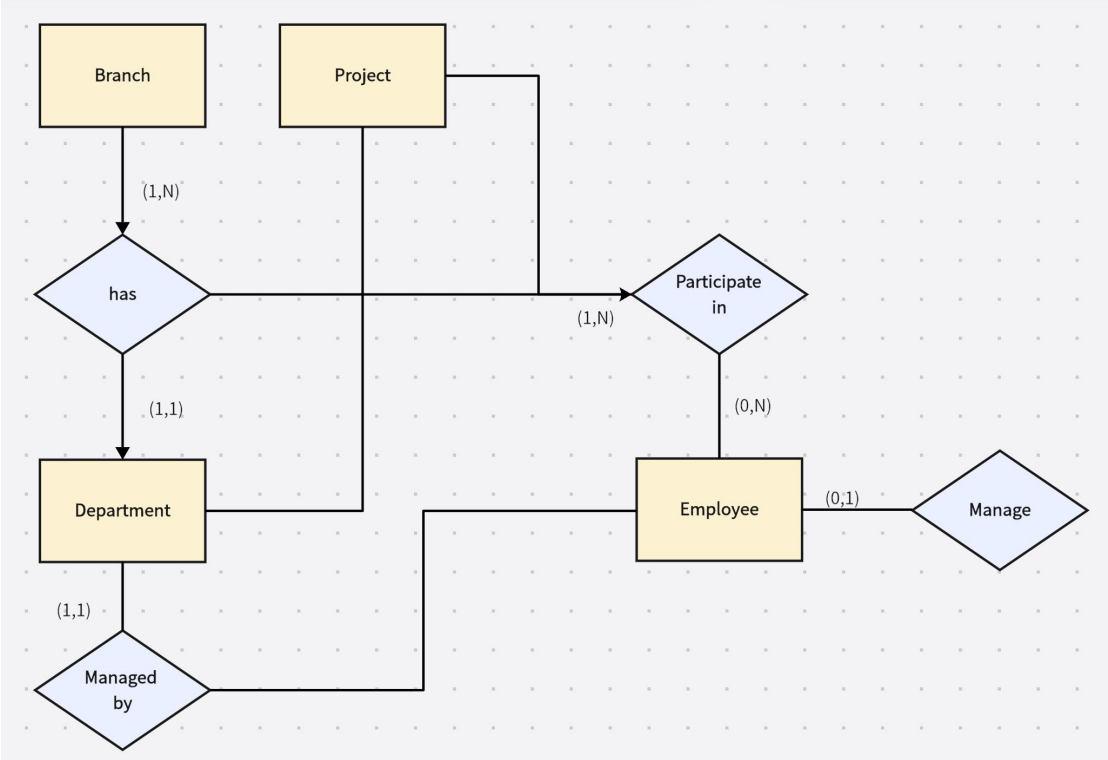
Efficiency and Scalability: Data modelling is essential for optimizing the performance and scalability of a database. By understanding the data requirements and relationships, designers can create efficient data structures and indexes that support fast data retrieval and manipulation. Additionally, a flexible data model can accommodate changes in data requirements over time, enabling the database to scale and adapt to growing needs without significant rework or redesign. This is particularly important in large-scale, enterprise-level systems where data volumes and user demands can be significant.

4.

a.

- A Firm consists of multiple Branches. Each branch represents a physical location or a division of the firm.
- Each Branch has one or more Departments. A department is a functional unit within a branch that focuses on a specific area of work.
- Each Department is managed by a single Employee who serves as the department head. Additionally, employees are assigned to a specific department and can only be members of one department at a time.
- Employees within the firm may participate in one or more Projects. A project is a temporary endeavor undertaken to create a unique product, service, or result. Each project must have at least one employee working on it.

b.



5.

Entities Identification:

Hospital: Represents the overall institution where medical services are provided.
Doctor: Represents the medical professionals who provide consultations and treatments to patients.
Patient: Represents the individuals who visit the hospital for medical services.
Test: Represents the medical examinations or procedures suggested by doctors to identify a patient's medical condition.

Attributes for Each Entity:

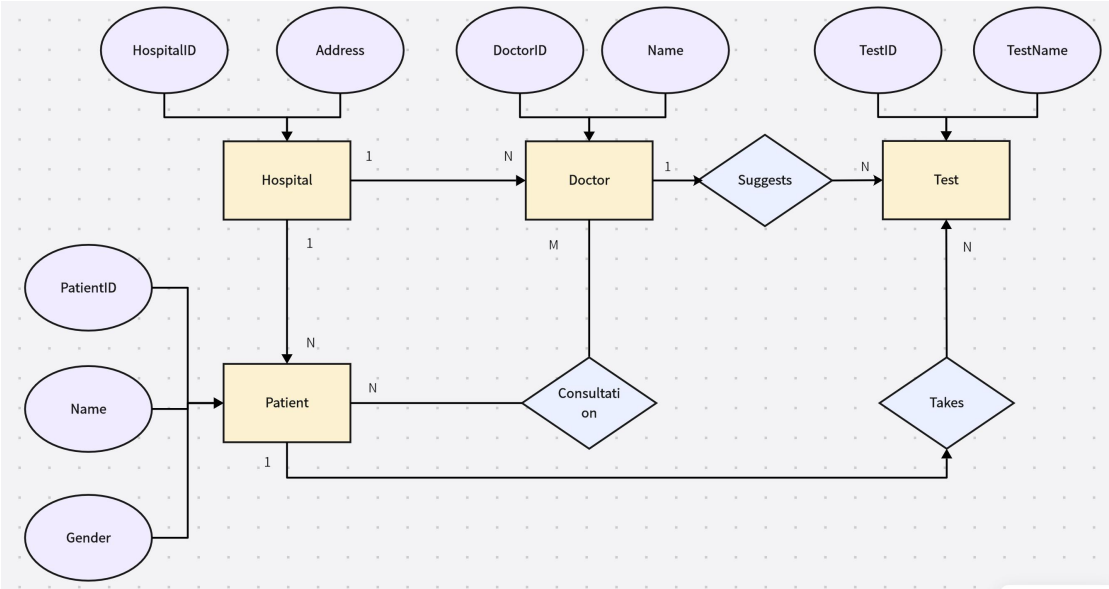
1.Hospital:
HospitalID (Primary Key)
HospitalName
Address
ContactNumber
Email

2.Doctor:

DoctorID (Primary Key)
Name
Specialization
Qualification
ContactNumber
Email
HospitalID (Foreign Key)

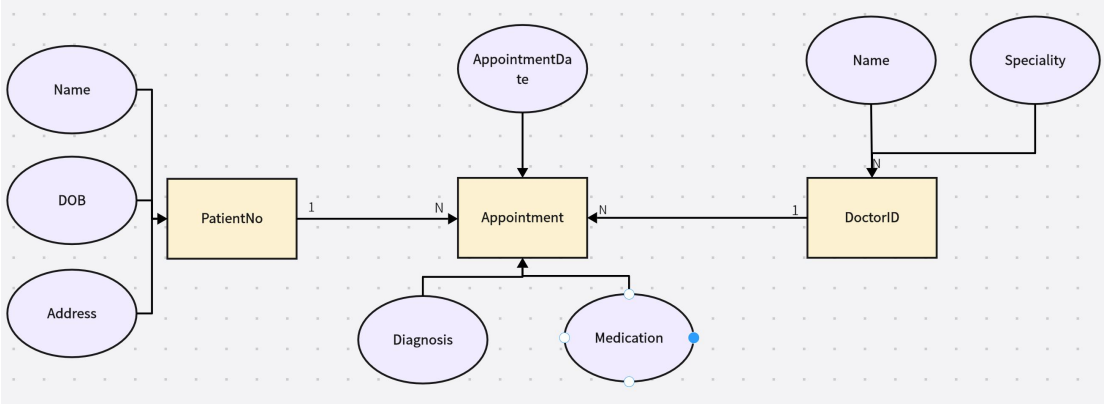
3.Patient:
PatientID (Primary Key)
Name
Age
Gender
ContactNumber
Email
Address

4.Test:
TestID (Primary Key)
TestName
Description
Cost

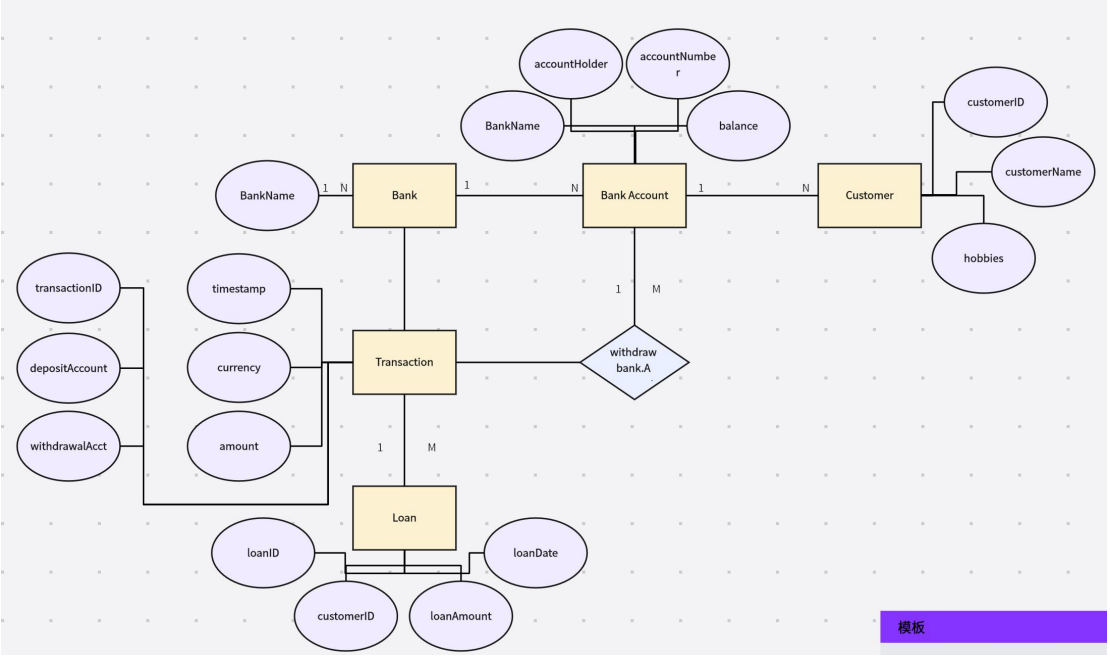


Medium Tasks

6.

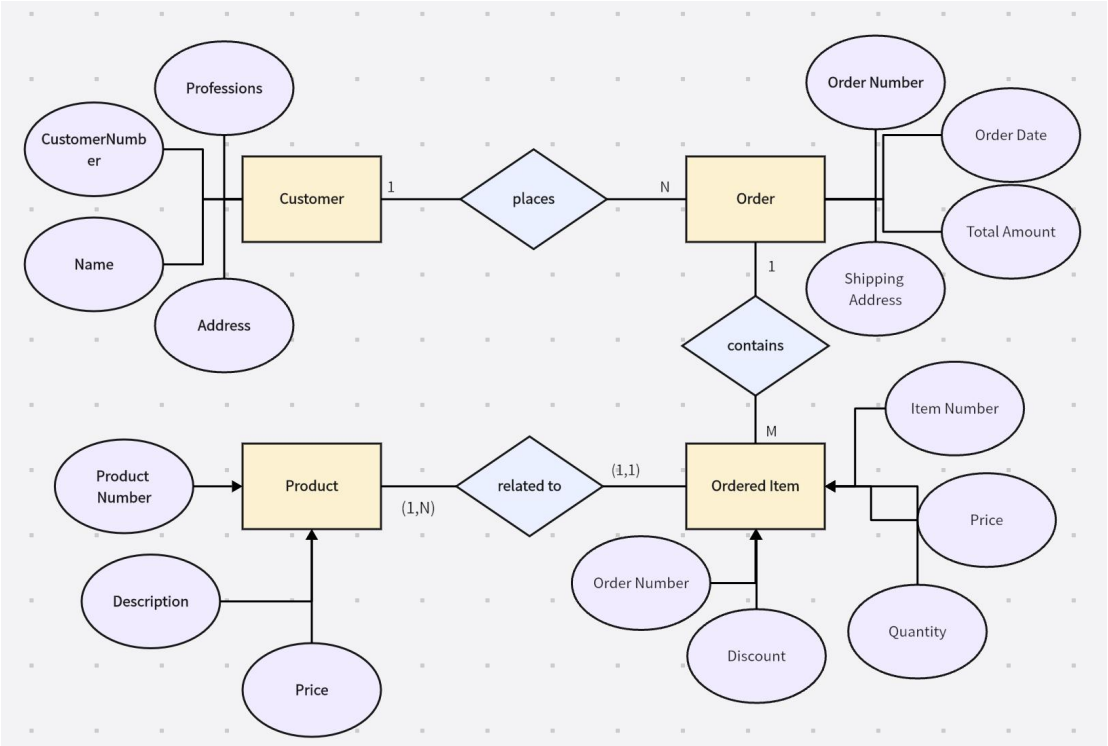


7.

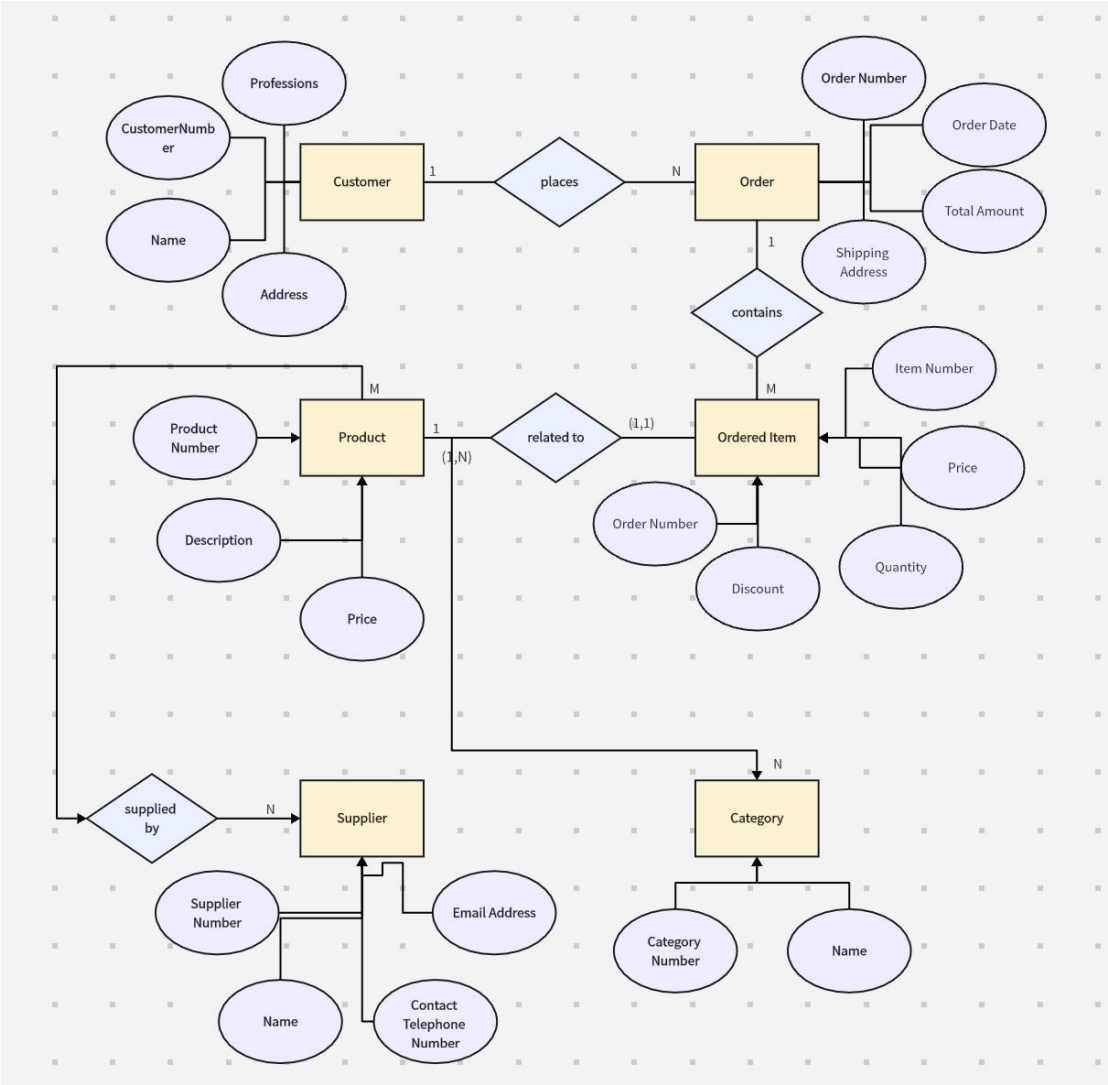


Advanced Tasks

8.



9.


Professions
CustomerNumber
Name
Address
Customer
1
places
N
Order
Order Number
Order Date
Total Amount
Shipping Address
1
contains
M
Item Number
Price
Ordered Item
M
(1,1)
related to
1
(1,N)
Product
Product Number
Description
Price
Order Number
Discount
Quantity
N
supplied by
N
Supplier
Category
Supplier Number
Email Address
Name
Contact Telephone Number
Category Number
Name