[ASSIGNMENT-2]

1) Explain with sketch the different phases of database design

* Database design can be defined as a set of procedures or collection of tasks involving various steps taken to implement a database.

⇒ Following are the phases / steps that can be taken by a database designer to ensure good database design?

Step 1: Determine the goal of your database, and ensure clear communication with the stakeholders. Understanding the purpose of a database will help in thinking of various use cases & what the problem may arise & how we can prevent it

Step 2: List down all the entities, that will be present in the database & what relationships exis among them

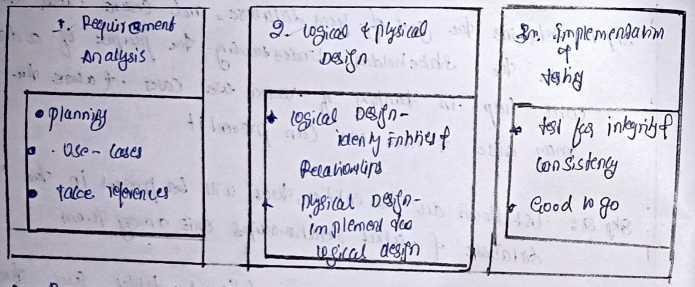
Step 3: Organize the information into different tables such that no or very little redundancy is there.

Step 4: Ensure uniqueness in every table. The uniqueness of records present in any relation is a very crucial part of database design that helps us avoid redundancy. Identify the key attribute to uniquely identify every row from columns. You can use various key constraints. To ensure the uniqueness of your table also keep in mind the uniquely identifying records must consume as little space as possible & should not

contain any null values

Step - 5: After all the tables are structured, and information is organised apply normalization form to identify anomalies that may arise of redundancy that can cause in consistency in the database

Database design lifecycle:



1. Requirement Analysis:- It's very crucial to understand the requirements of our application so that you can frame in productive terms. And imply appropriate integrity - constraints - to maintain the data integrity of consistency.

2. logical & physical design:- This is the actual design phase that involves various steps that are to be taken while designing a database. This phase is further divided into two stages

* Logical data model design:- This phase consists of coming up with a high-level design of our database based on initially gathered requirements to structure & organize our data accordingly.

* Physical Design of data model:- This phase involves the implementation of the logical design made in the previous stage. All the relationships among data and integrity constraints are implemented to maintain consistency & generate the actual database.

3. Data Insertion and testing for various integrity constraints

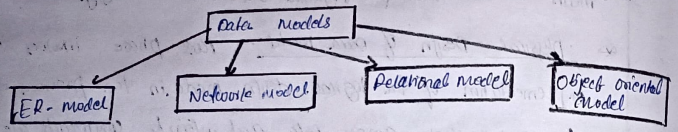
After implementing the physical design of the database, we're ready to input the data & test our integrity. This phase involves testing our database for its integrity - to see if something got left out as anything - need to add & then integrating it with the desired application.

a) Logical data model design:-

The logical data model design defines the structure of data and what relationship exists among those data. The following are the major components of the logical design

⟹ Data models:- Data modeling is a visual modeling technique used to get a high-level overview of our database. Data models help us understand the needs and requirements of our database by defining the design of our database through diagrammatic

representation Ex: model, network model, Relational model, object-oriented data model.



3) Entity: Entities are objects in the real world, which can have certain properties & these properties are referred to as attributes of that particular entity. There are 2 type of entities Strong and weak entity. weak entity do not have a key attribute to identify them, their existence solely depends on one or more specific strong entity & also have full participation in a relationship whereas strong entity does have a key attributes to uniquely identify them.

Ex: weak entity

3) Relationship: How data is logically related to each other defines the relationship of that data with others entities. In simple words, the association of one entity with another is defined here.

A relationship can be further categorized into - unary, binary and ternary relationships.

4) Attributes:- are nothing but properties of a specific entity that define its behaviour.

5) Normalization:- After all the entities are put in place and the relationship among data is defined.

2) Book club has members to whom the books are sold. The books are made available at different places in the city. The books are identified by a book-id, their author and their publishers. An author can write more than one book and a book can have more than one author. Members have information such as membership-id, Name, Phone # and Status. A member can place more than one order. You can choose additional attributes for the schema that seem appropriate. mention any assumption you make. Show minimum and maximum Cardinality values based on your assumptions. Design an E-R schema diagram for this application

Assumptions:-

Unique identifiers:- Each entity such as Book, Author, Member, Order, location has a unique identifier. eg: book-id, author-id, membership-id, rider-id, and location-id, considered as unique primary keys

Book and Author Relationship:-

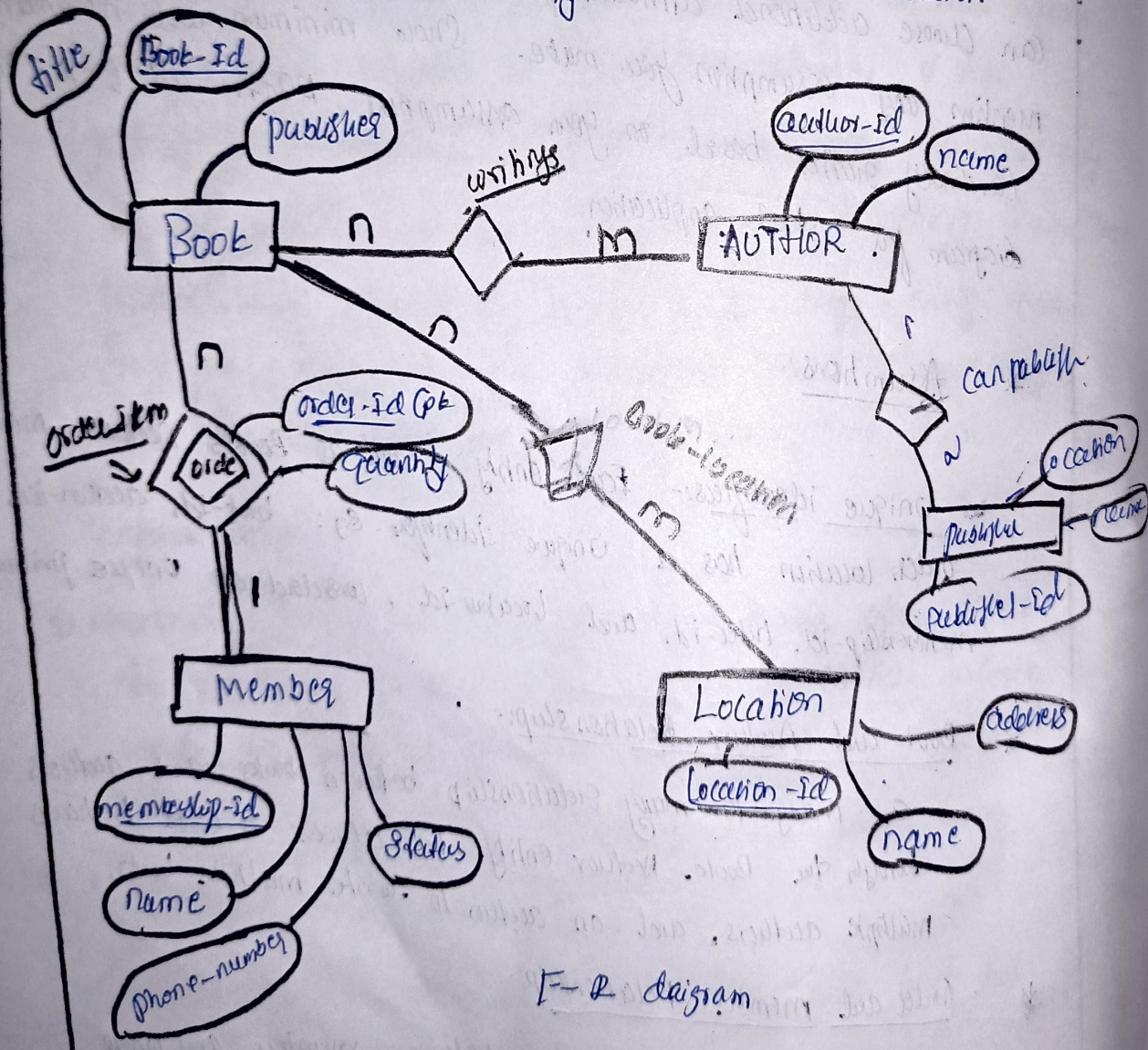
a many-to-many relationship between books and authors through the Book-Author entity. This allows a book to have multiple authors, and an author to write multiple books.

Order and member Relationship:-

A one-to-many relationship between members and orders. This means a member can place multiple orders but each order is associated with only one member.

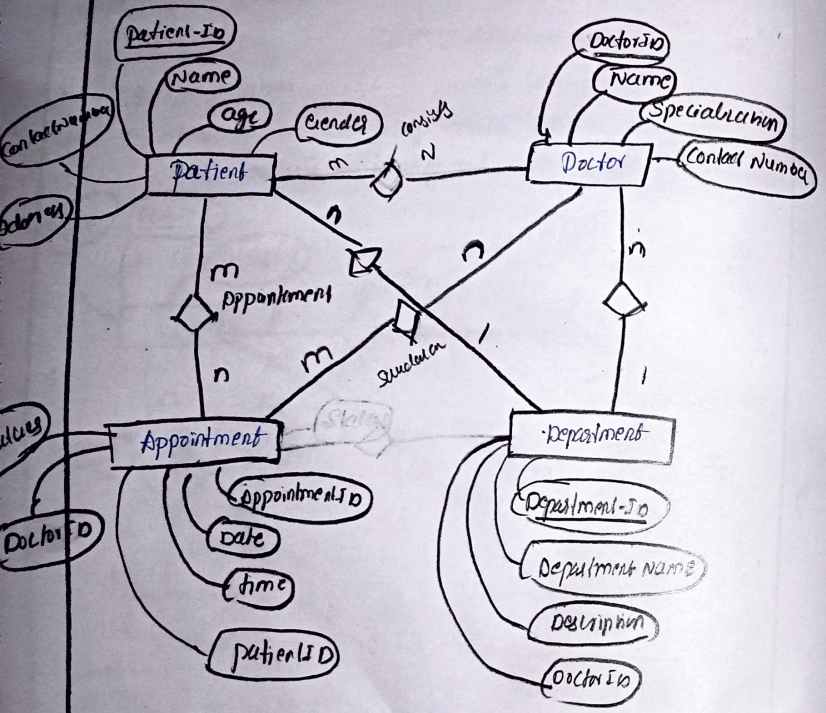
* Book and location Relationship: a many-to-many relationship between books and locations through the Book_Location entity. This allows a book to be available at multiple locations and a location to have multiple books.

* Location information:
  * Each location entity includes attributes such as name and address to identify and describe each location.



E-R diagram

3. write an ER diagram for hospital management system. Assume minimum 4 entities, attributes and relations.



4] Consider the following relation schema

Project (P-no, P-name, P-incharge)

Employee (E-No, E-name)

Assigned-to (P-no, E-No)

→ Create table project (
P-no Int primary key,
P-name varchar (200),
P-incharge varchar (200)
);

Create table employee (
E-no Int primary key,
E-name varchar (255)
);

→ Create table Employee (
E-no Int primary key,
E-name varchar(200)
);

→ Create table Assigned-to (
P-no Int,
E-no Int,
primary key (P-no, E-no),
foreign key (P-no) references Project (P-no),
foreign key (E-no) references Employee (E-no));

- Inserting into project table

Insert into project (p-no, p-name, p-incharge) values (1, 'project A', 'manger');

Insert into project values (2, 'projectB', 'mangerB');

Insert into project values (3, 'projectC', 'mangerC');

- Inserting into Employee table

Insert into Employee values (101, 'John');

Insert into Employee values (102, 'Tom');

Insert into Employee values (103, 'Alice');

- Inserting into Assigned-to table.

Insert into Assigned-to values (1, 101);

Insert into Assigned-to values (1, 102);

Insert into Assigned-to values (2, 103);

Insert into Assigned-to values (3, 101);

Insert into Assigned-to values (3, 102);

- Write the Relational Algebra queries for the following:

i) List details of employee who are working on all the projects

Π Employee (E-no, E-name) (

Employee - Π employee.E-no (

Employee $\times$ (Employee

$R_1 \leftarrow$ (Assigned-to $\bowtie$ Employee.E-no = Assigned-to.E-no)

$R_2 \leftarrow$ $\neg$ not Exists Assigned-to.E-no (R_1)

$R_3 \leftarrow$ ($R_1 \bowtie$ Assigned-to P-no = project-no)

$R_4 \leftarrow$ Π Employee

$R_4 \leftarrow$ $\neg$ not exists project.p-no (R_3)

$R_5 \leftarrow$ Π Employee.E-no, Employee.E-name (R_4)

E_No	E-Name
103	Alice

ii) List E-no of employee who are not working on project number 2k

$R_1 \leftarrow \sigma_{p\_no = '2k'}(Assigned\_to)$

$R_2 \leftarrow \Pi_{Assign\_to.Eno}$

$R_3 \leftarrow$ Employee

$R_4 \leftarrow \Pi_{Employee.E\_no}(R_3 - (R_2)(R_1))$

or $R_4 \leftarrow \Pi_{Employee.E\_no}(Employee - \Pi_{Assign\_to.Fno}(\sigma_{p\_no = '2k'}(Assigned\_to)))$

E_no
101
102
103

iii) List the names of employee who are working in the same project as employee named 'Tom'.

$R_1 \leftarrow \sigma_{Employee.E\_name = 'Tom'}(Employee)$

$R_2 \leftarrow (R_1 \bowtie Assigned\_to \; | \; Employee.E\_no = Assigned\_to.E\_no)$

$R_3 \leftarrow (R_2 \bowtie project \; | \; Assigned\_to.p\_no = project.p\_no)$

$R_4 \leftarrow \Pi_{Employee.E\_name}(R_3)$
E_no

E_no	E_name
101	John
102	tom
103	Alice