Chapter-2 (12 Marks)

a) Define generalization and specialization with its notation and examples. [4]

) Explain the relational model of a database system along with foreign key constraint with appropriate example. [4]

Distinguish between degree and cardinality of a relationship in an ER diagram. Explain generalization in ER diagram with an example. [2+2]

a) Define data independence and explain its significance. What is importance of aggregation in ER design? Discuss with an example. [2+2]

Mention the distinctions among the terms Generalization and Specification with appropriate symbolic representation. [4]

) Define unary relationship along with example. How you convert an ER relationship into relation schema? Explain with examples of different cardinalities. [2+4]

(such as 2 to 5).

Explain strong and weak entity sets along with example.

a) Define discriminator in ER diagram. Explain different keys used in database design. [4]

a) What are data models? Explain various types of data models. [1+3]

instances.

2. Differentiate total and partial participation with suitable example

Explain how network data model is different from relation data model. [4]

Draw an ER-diagram for the following mini-case. What is the difference between strong and weak entity sets?

Explain generalization and specialization in ER diagram along with an example? [8 + 4]

What is the difference between the degree and cardinality of a relationship? [8 + 4]

Data Models

A data model is an abstract representation of how data is structured, stored, and managed in a database. It defines the logical design, relationships, and constraints of data in a structured way.

Types of Data Models

1. Hierarchical Data Model

-> Organizes data in a tree-like structure.
-> Data is stored in parent-child relationships (one parent, multiple children).
-> Example: IBM's Information Management System (IMS).

2. Network Data Model

-> Similar to the hierarchical model but allows many-to-many relationships using pointers.
-> Data is stored using a graph-like structure.
-> Example: CODASYL DBMS.

3. Relational Data Model (RDM)

-> Stores data in tables (relations) consisting of rows (tuples) and columns (attributes).
-> Uses keys to establish relationships.
-> Example: MySQL, PostgreSQL, Oracle DB.

4. Entity-Relationship (ER) Model

-> Represents real-world entities, their attributes, and relationships using ER diagrams.
-> Uses symbols like rectangles (entities), diamonds (relationships), and ovals (attributes).
-> Example: Used in database design before implementing relational databases.

5. Object-Oriented Data Model (OODM)

-> Stores data as objects (similar to OOP concepts).

-> Supports complex data types and relationships.

-> Example: Object-oriented databases like MongoDB (NoSQL).

Perspectives of Data Models: Conceptual, Logical, and Physical

A data model can be viewed from three different perspectives: Conceptual, Logical, and Physical. These perspectives help in designing and structuring a database from a high-level overview to the actual implementation.

1. Conceptual Data Model

-> The conceptual model provides a high-level view of the data, focusing on the business requirements rather than technical details.

-> Represents entities, attributes, and relationships.

-> Does not include storage details or data types.

-> Used mainly by business analysts and stakeholders.

2. Logical Data Model

-> The logical model is a detailed version of the conceptual model, specifying the structure of data but still independent of any specific database management system (DBMS).

-> Defines tables, columns, primary keys, and foreign keys.

-> Specifies data types and constraints.

-> Does not include physical storage details.

3. Physical Data Model

-> The physical model represents how data is actually stored in a specific database management system (DBMS).

-> Specifies database-specific details (e.g., MySQL, PostgreSQL, Oracle).

-> Defines indexes, partitions, and data storage methods.

-> Includes performance optimization like indexing and caching.

ER Model (Entity-Relationship Model) Advantages

-> The ER diagram provides a visual blueprint of data relationships, making it easier to understand database structure.

-> Helps in structuring data efficiently, reducing redundancy.

-> Non-technical stakeholders can understand relationships easily.

-> Helps in minimizing data redundancy and improving consistency.

-> Can be easily converted into relational database schemas.

Strong Entity Set

-> A strong entity set is an entity that has a primary key and can exist independently in a database.

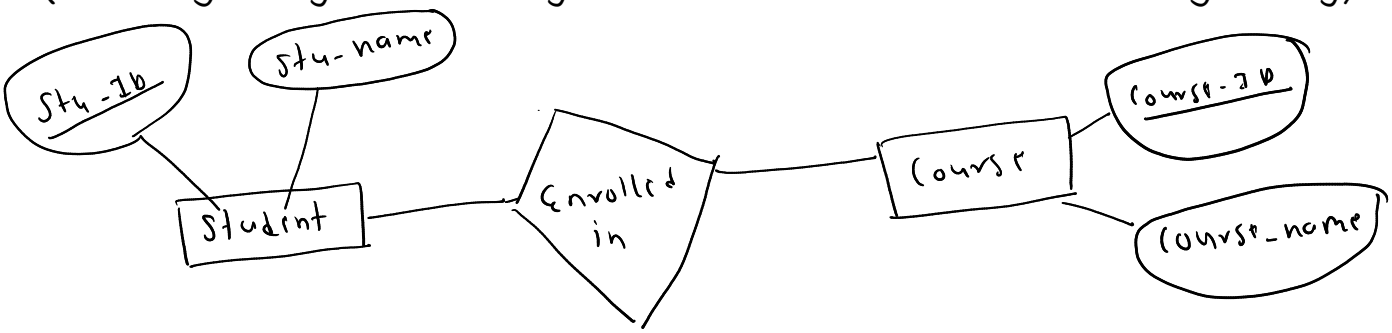
-> It does not depend on any other entity for identification.

Weak Entity Set

-> A weak entity set does not have a primary key and depends on a strong entity for identification.

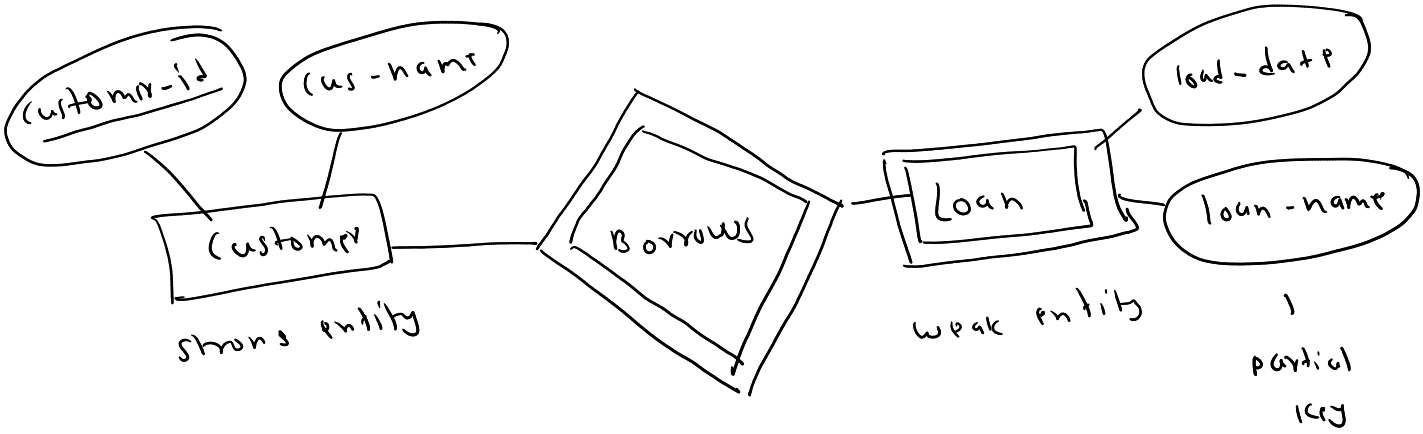
-> It requires a foreign key (called a discriminator or partial key) to uniquely identify its instances.

-> It has a total participation relationship with the strong entity (meaning every weak entity must be associated with a strong entity).



→ Student and Course are Strong entity.

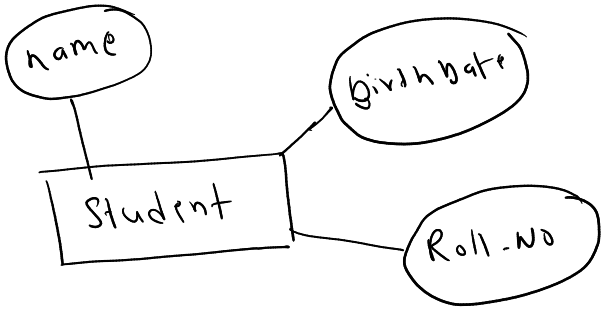
(both have primary key)



Strong Entity Set	Weak Entity Set
Always has a primary key.	Has a partial discriminator key.
Not dependent on any other entity.	Dependent on a strong entity.
Represented by a single rectangle.	Represented by a double rectangle.
Relationship with another strong entity set is represented by a single diamond.	Relationship with a strong entity set is represented by a double diamond.
May have either total or partial participation.	Always has total participation.

Attributes

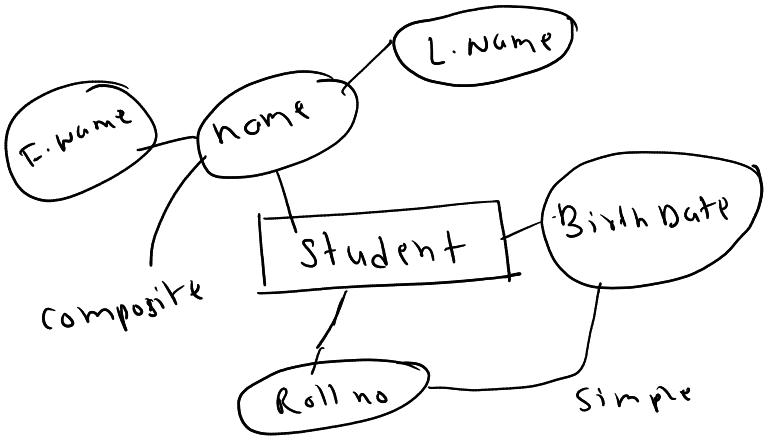
-> descriptive properties of an entity set. represented by an ellipse



Simple and Composite Attributes

-> simple can't be divided into subparts

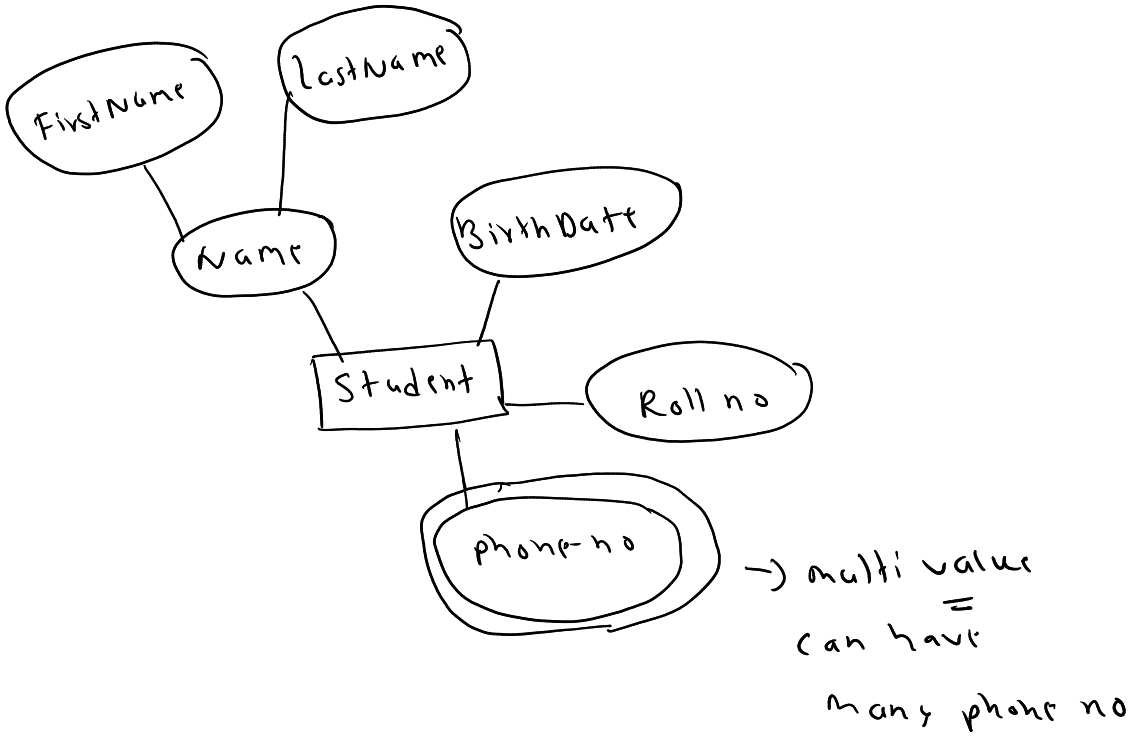
-> composite can be divided into subparts



Single-Valued and Multi-Valued Attributes

-> single can only have one value

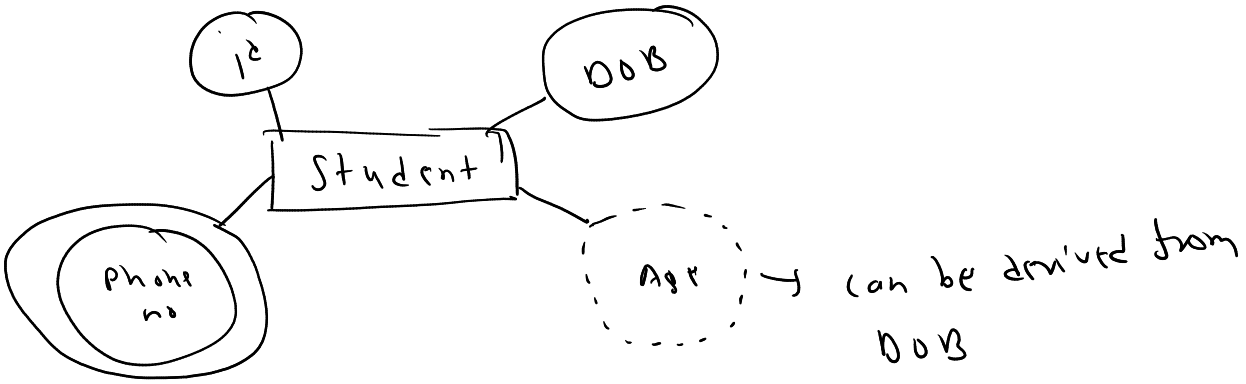
-> multi can have more than one values. Represented by double ellipse.



Derived Attribute

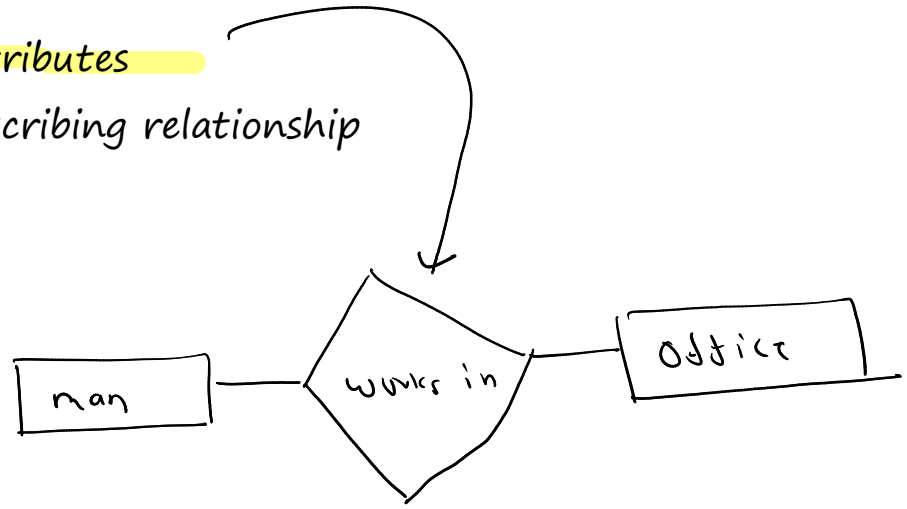
-> whose value can be derived from values of other related attributes.

-> represented by dashed ellipse



Descriptive Attributes

-> used for describing relationship



Keys

In DBMS (Database Management System), keys are used to uniquely identify records in a table and establish relationships between tables.

1. Primary Key

-> A column (or set of columns) that uniquely identifies each row in a table.

-> Cannot have NULL values and must contain unique values.

2. Candidate Key

-> A set of attributes that could be a primary key (i.e., unique and not NULL).

-> A table can have multiple candidate keys, but only one primary key is chosen from them.

-> Example: If a Students table has StudentID and Email, both can be candidate keys.

3. Super Key

-> A superset of a candidate key.

-> It includes additional attributes but still uniquely identifies rows.

-> Example: {StudentID, Name} is a super key, but {StudentID} alone is a candidate key.

4. Foreign Key

-> A column that creates a relationship between two tables by referring to the primary key of another table.

-> Here, StudentID in Orders is a foreign key that refers to StudentID in Students.

5. Composite Key

-> A key made up of two or more columns that uniquely identify a row.

-> Used when a single column is not sufficient to ensure uniqueness.

-> Here, (StudentID, CourseID) together form a composite primary key.

6. Alternate Key

-> The candidate keys that are not chosen as the primary key.

-> Example: If both StudentID and Email are candidate keys, but StudentID is the primary key, then Email is an alternate key.

7. Unique Key

-> Ensures all values in a column are unique, but unlike the primary key, it can contain NULL values.

-> Here, Email is unique but can have NULL values.

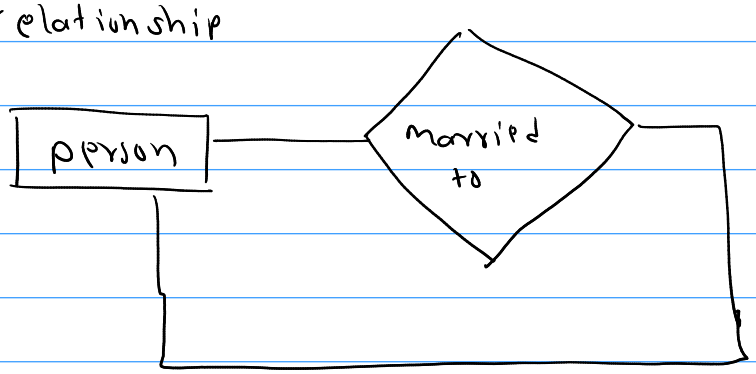
8. Surrogate Key

-> A system-generated key (like an auto-incremented ID) used as the primary key.

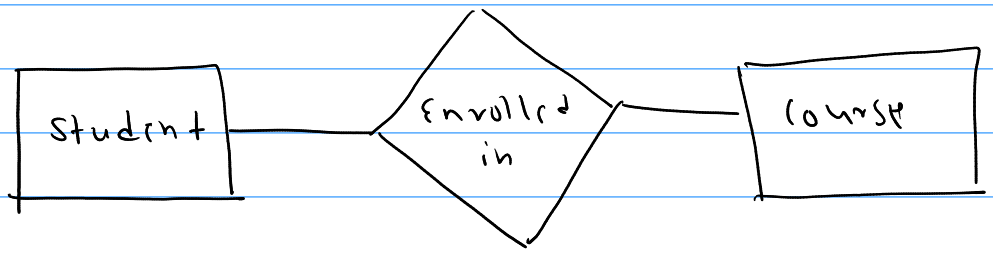
Degree of Relationship

→ Different type of entity set participating in a relation

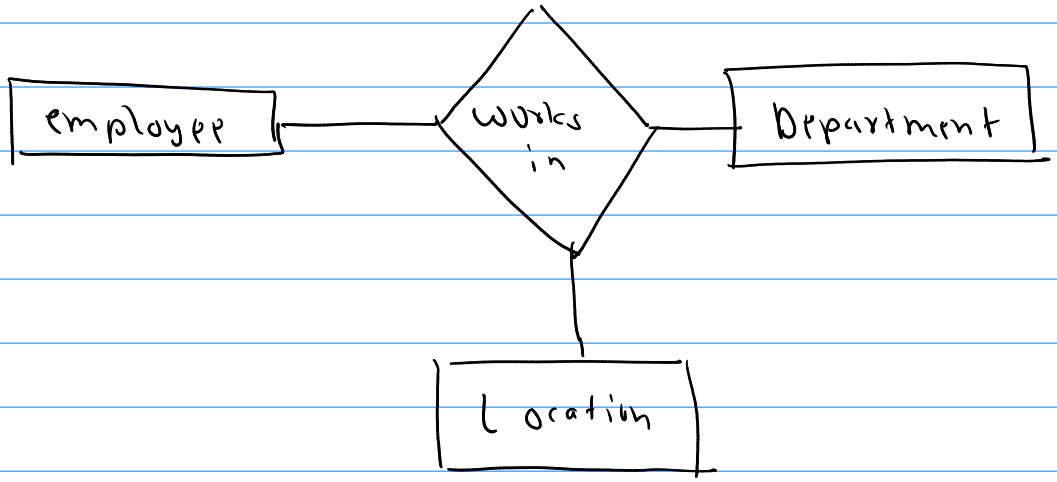
① Unary relationship



② Binary Relationship Set



③ Ternary Relationship set



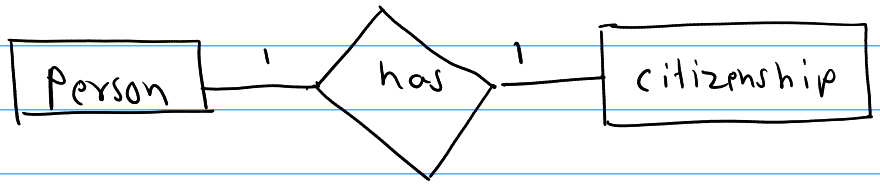
Mapping Constraints

① Cardinality Constraints
② Participation Constraints

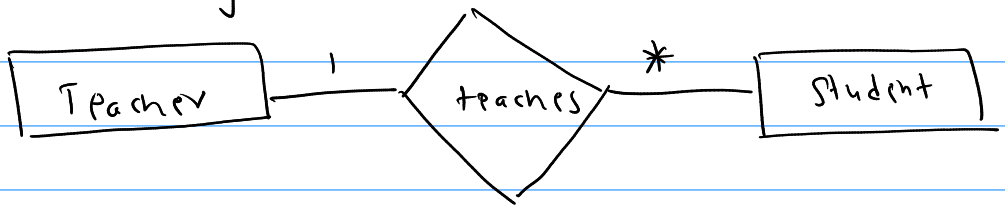
Cardinality constraints

→ Expresses the ratio of number of entity participating in the relation.

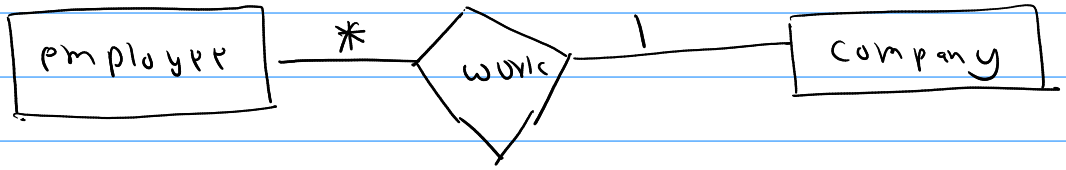
ⓐ one to one cardinality



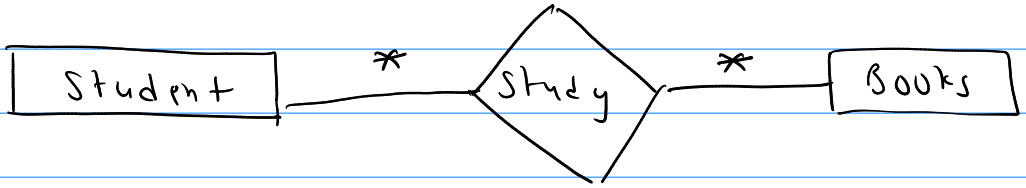
ⓑ One to many cardinality



ⓒ Many to one cardinality



ⓓ Many to many cardinality



Participation Constraints

Participation constraints define whether all or only some entities from an entity set must participate in a relationship in an ER (Entity-Relationship) Model. There are two types:

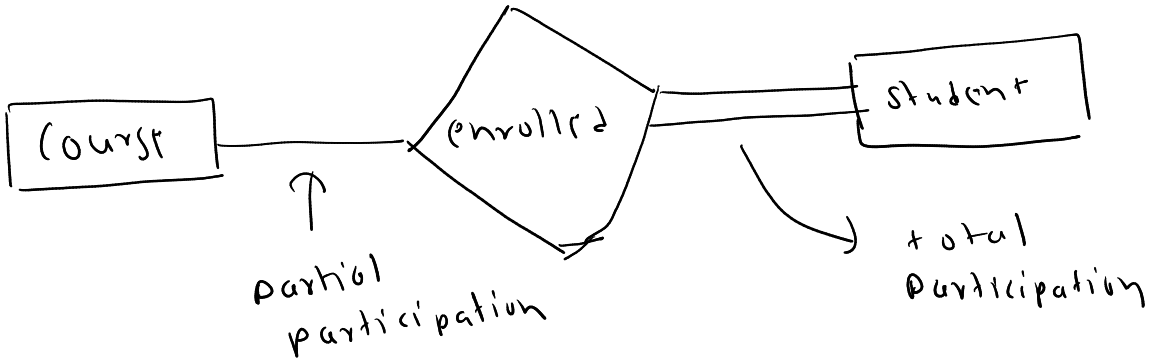
1. Total Participation

-> All entities in the entity set must be involved in the relationship.

-> Represented by a double line in the ER diagram.

-> Every student in a university must be enrolled in at least one course.

-> So, the Student entity has total participation in the Enrollment relationship.



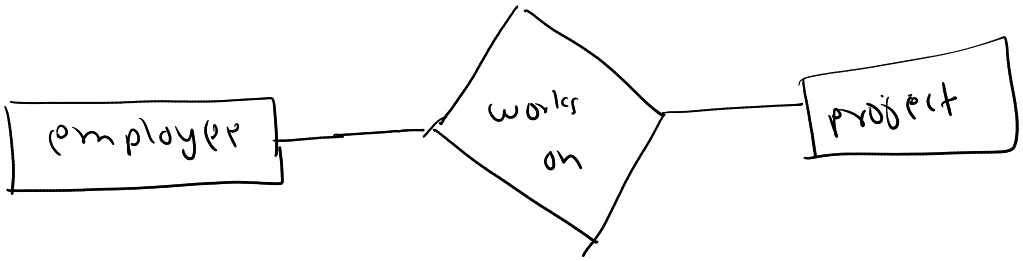
2. Partial Participation

-> Only some entities from the entity set participate in the relationship.

-> Represented by a single line in the ER diagram.

-> Not all employees in a company are assigned to projects.
Some employees may not be part of any project.

-> So, the Employee entity has partial participation in the Works_On relationship.



Specialization

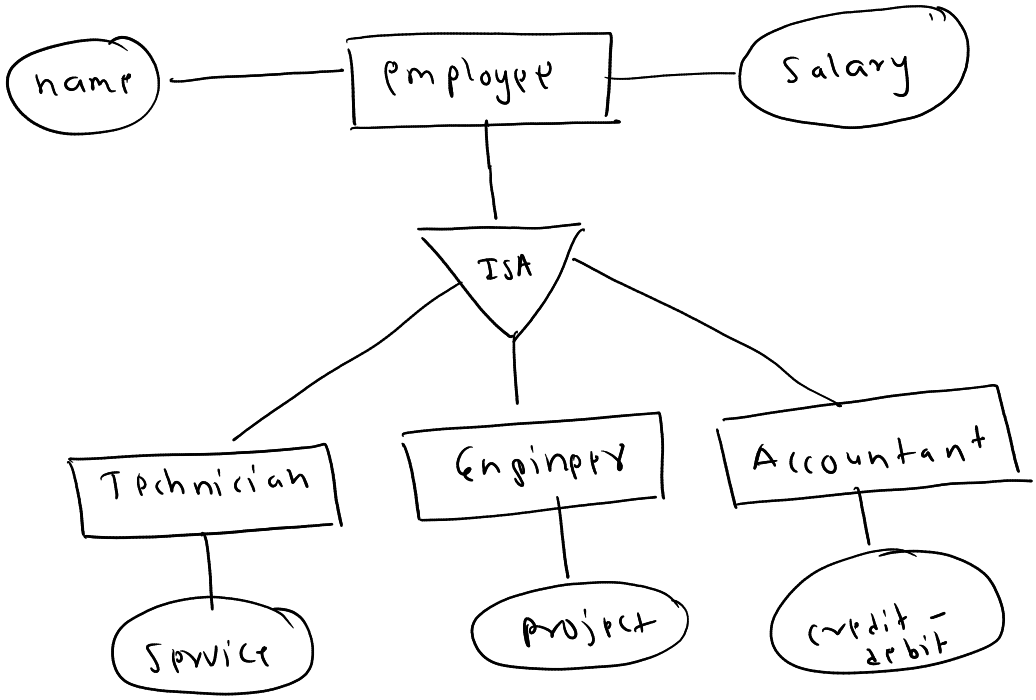
-> Breaking down a general entity into more specific sub-entities.

-> It divides a higher-level entity into two or more specialized lower-level entities.

-> Used when some entities have additional attributes or behaviors that others don't.

-> Consider an entity Employee. Some employees are Teachers, and some are accountant, some are engineers.

-> Employeed is specialized into teacher, accountant and engineer



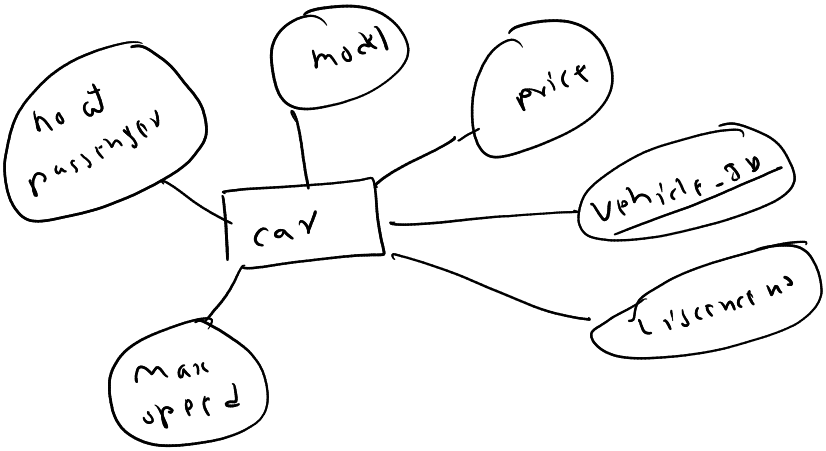
2. Generalization

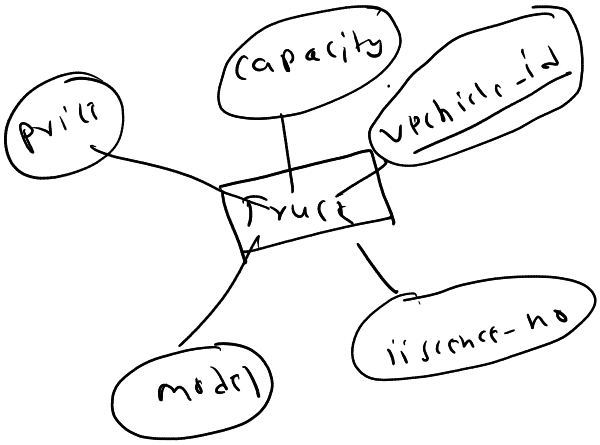
-> Combining multiple related entities into a single generalized entity.

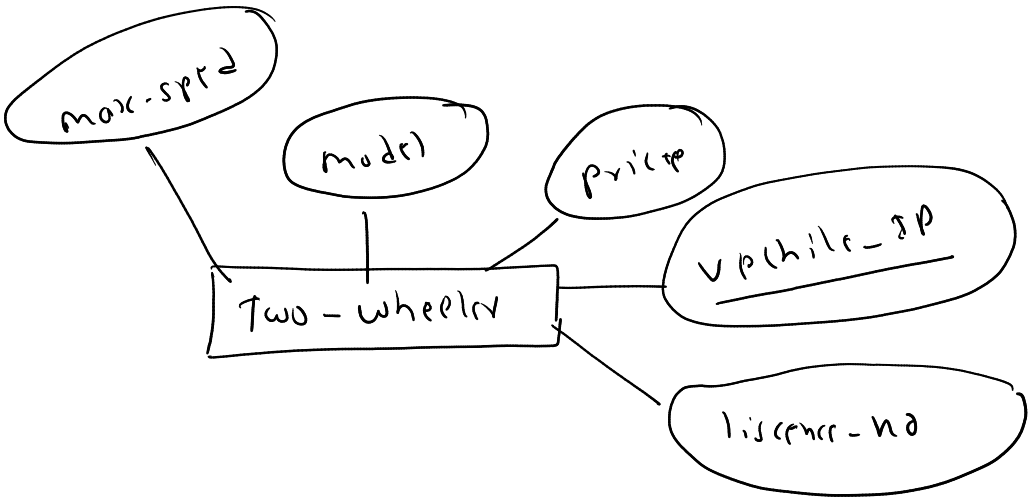
-> It is the reverse of specialization.

-> Used to reduce redundancy by merging common attributes into a higher-level entity.

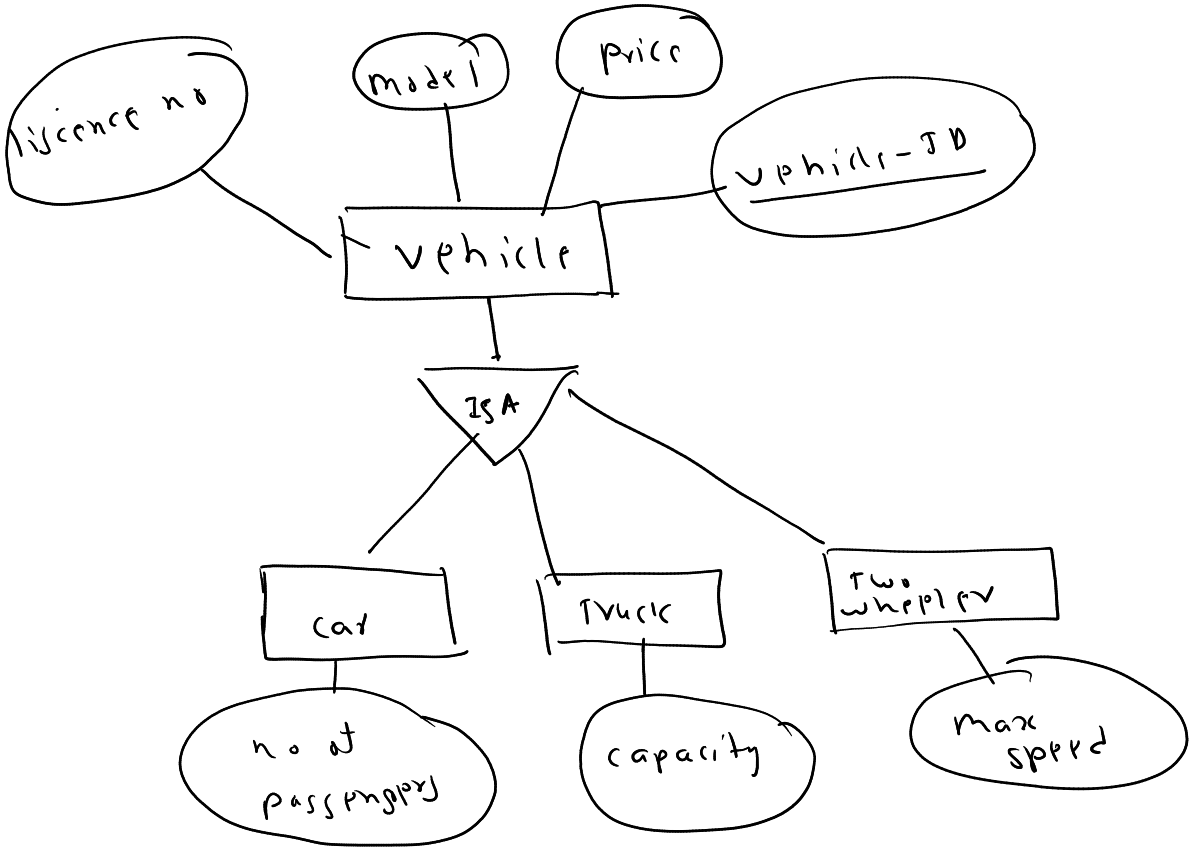
-> Entities Car, Bike, and Truck can be generalized into Vehicle.







Before generalization



(after generalizing)

More example of specialization

1. **Entity type Employee before specialization**

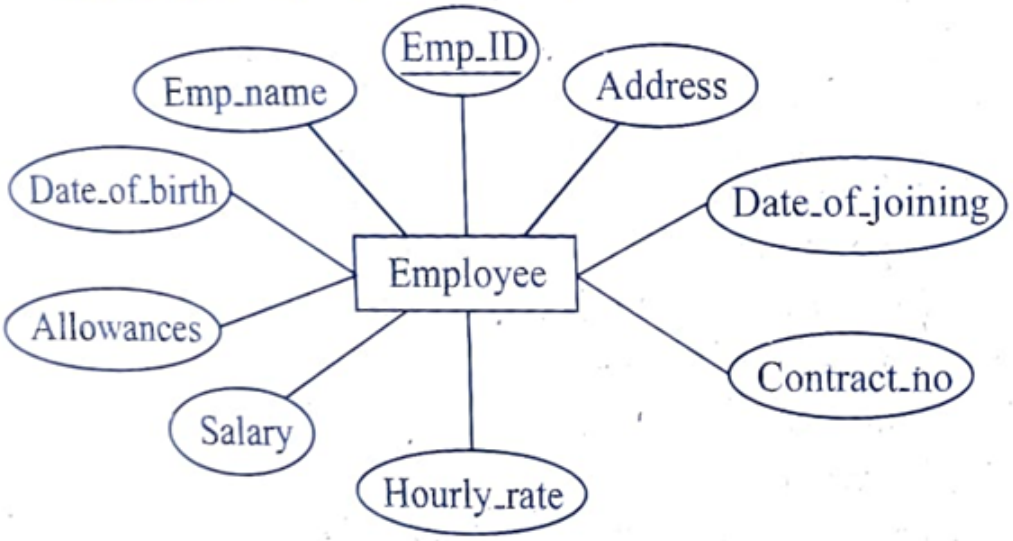


Figure: Employee before specialization

2. **Entity type Employee after specialization**

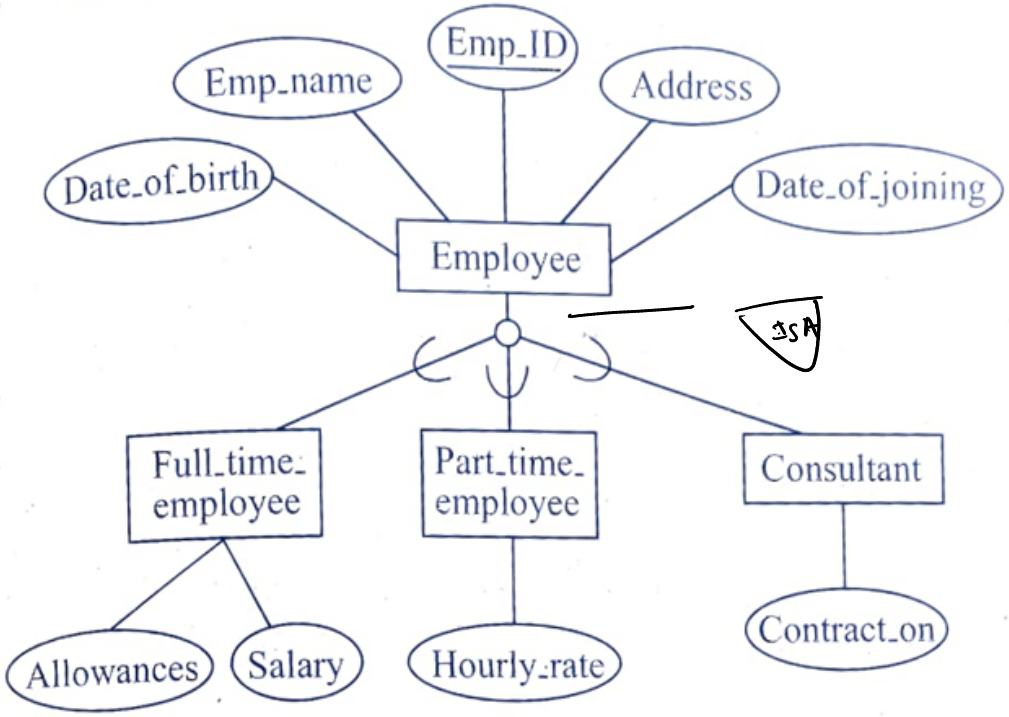


Figure: Employee after specialization

3. Aggregation

-> Aggregation is an advanced ER model concept where a relationship itself acts as an entity. It is used when a relationship needs to be connected to another entity.

-> Sometimes, a relationship between two entities needs to participate in another relationship.

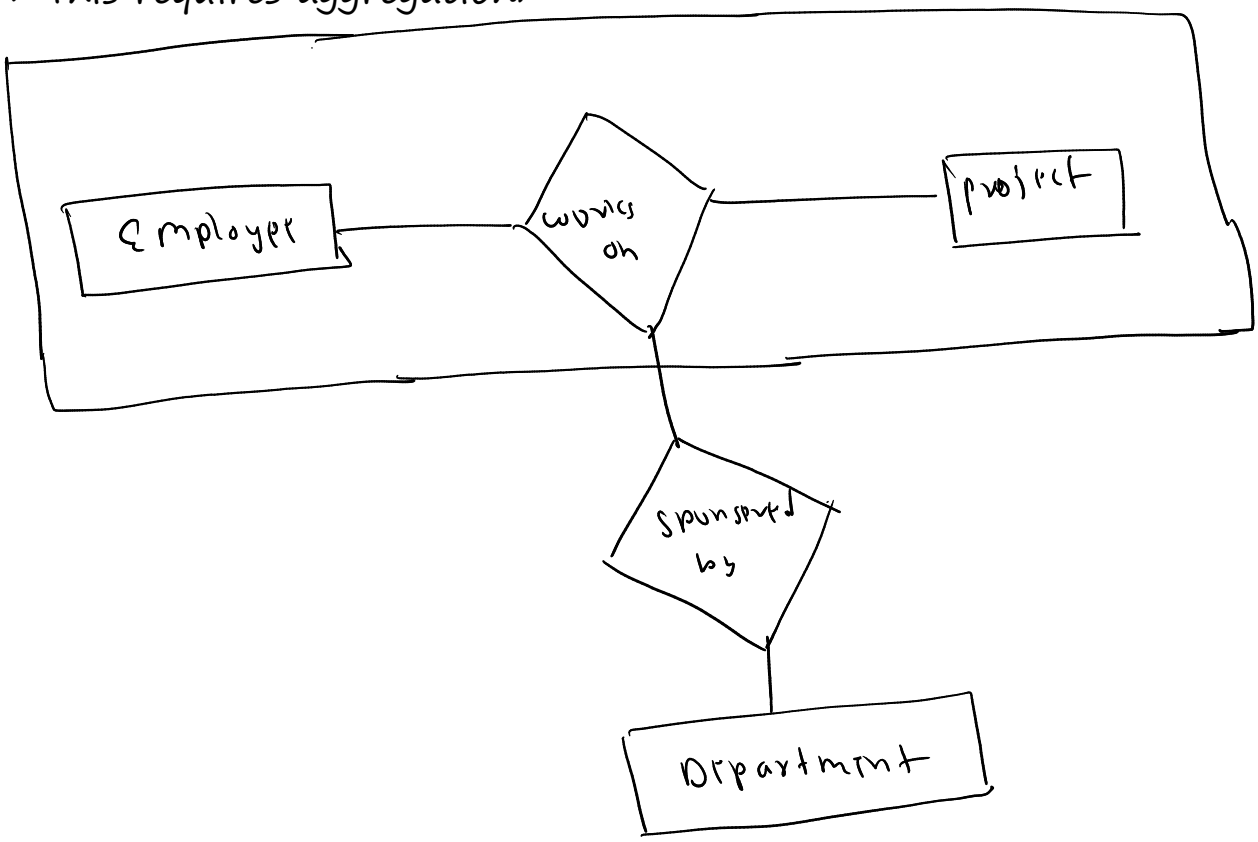
-> Normal ER models only allow entities to be related, but with aggregation, we can relate relationships to other entities.

-> 👤💻 Employee works on a 🗂️ Project, and that project is sponsored by a Department.

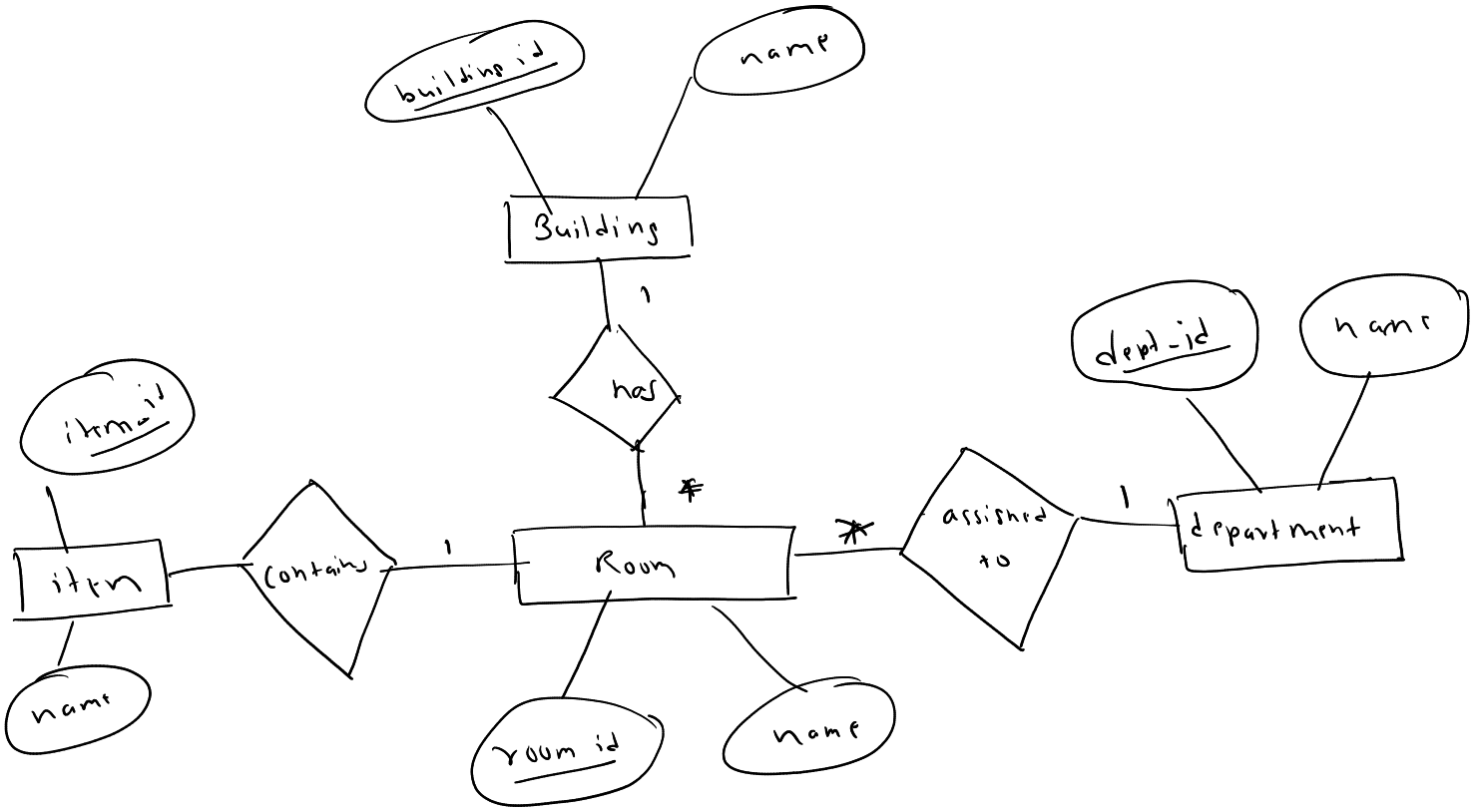
-> Employee and Project are related via the Works_On relationship.

-> The Works_On relationship itself needs to be associated with Department (which sponsors the project).

-> This requires aggregation.



1. Draw an ER diagram for the mini-case 'Procurement department of an Organization (ABC) keeps track of all the items such as Furniture and equipment in the offices. There are several buildings of the ABC and each one is given a different name to identify it. Each item is assigned a unique ID when it is purchased. This ID is used to keep track of the item, which is assigned to a room within a building. Each room within a building is assigned to a department'. Document all assumptions that you make about the mapping constraints.



→ Building has many rooms

→ many rooms are assigned to 1 depart

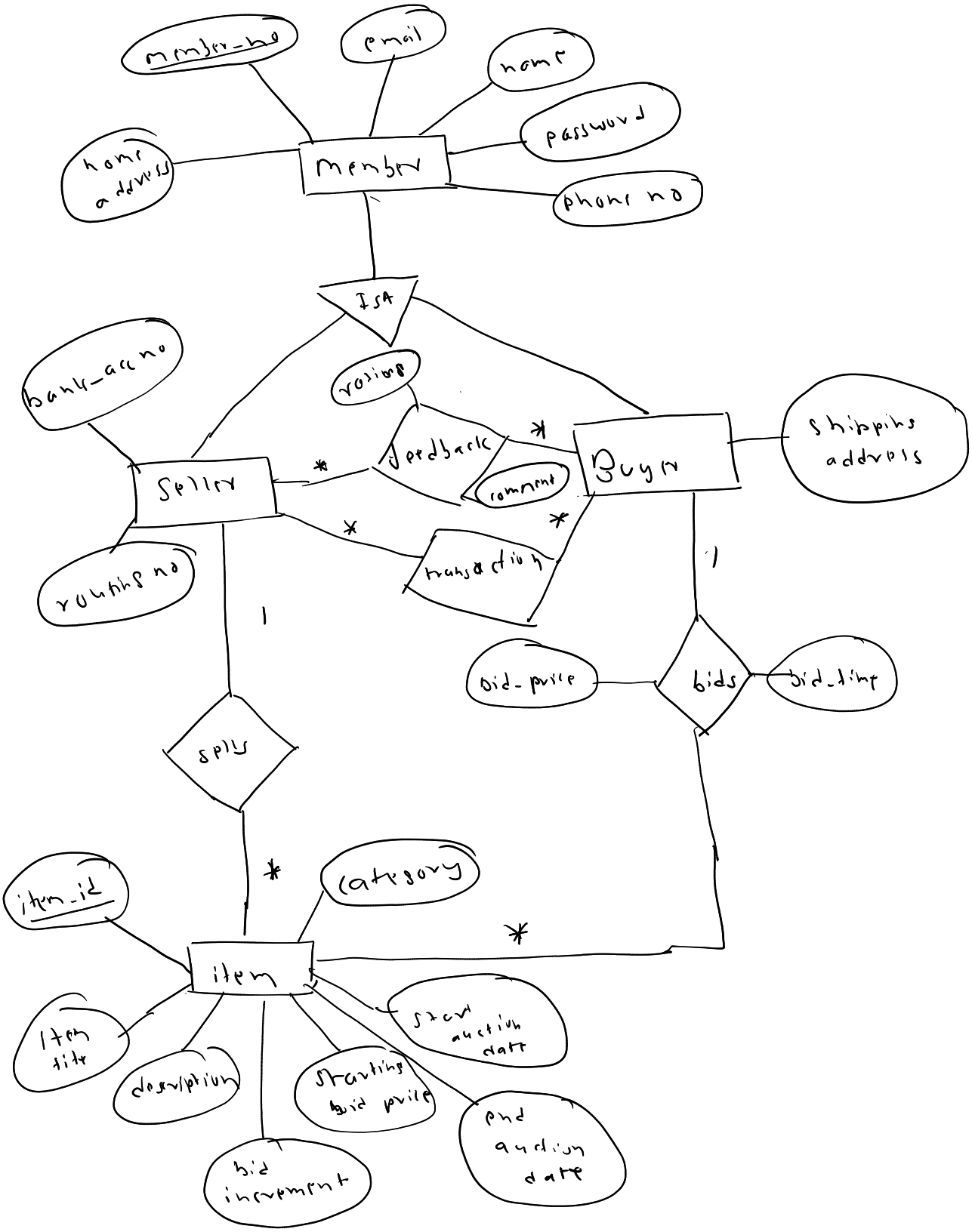
→ Room contains many items

2. Consider an ONLINE AUCTION database system in which members participa in the sale of items. The data requirements for this system are summarized as follows:

The online site has members, each of whom is identified by a unique member number and is described by an e-mail address, name, password, home address and phone number. A member may be a buyer and seller. A buyer has a shipping address recorded in the database. A seller has a bank account number and routing number recorded in the database. Items are placed by a seller for sale and are identified by a unique item number assigned by the system. Items are also described by an item title, a description, starting bid price, bidding increment, the start date of the auction and the end date of the auction. Items are also categorized based on a fixed classification hierarchy. Buyers make bids for items they are interested in. Bid price and time of bid is recorded. The bidder at the end of the auction with the highest bid price is declared the winner and a transaction between buyer and seller may then proceed. The buyer and seller may record feedback regarding their completed transactions. Feedback contains a rating of the other party participating in the transaction (1-10) and a comment. Design an Entity-Relationship diagram for the ONLINE AUCTION database.

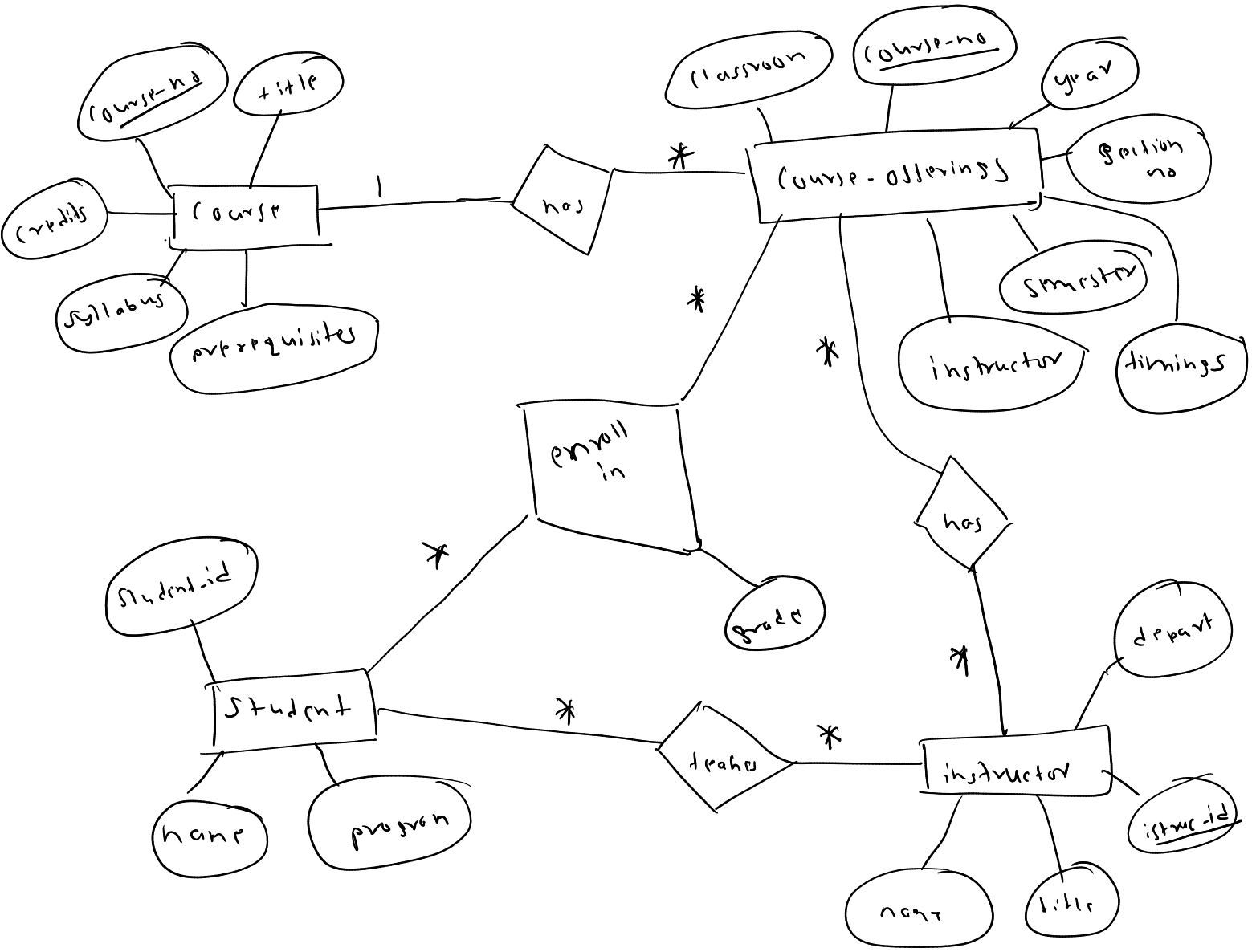
Key Relationships:

Seller lists Items.
Buyer makes Bids on Items.
Winning Bid → Transaction.
Transaction → Feedback (buyer and seller rate each other).
Items categorized in a hierarchy.

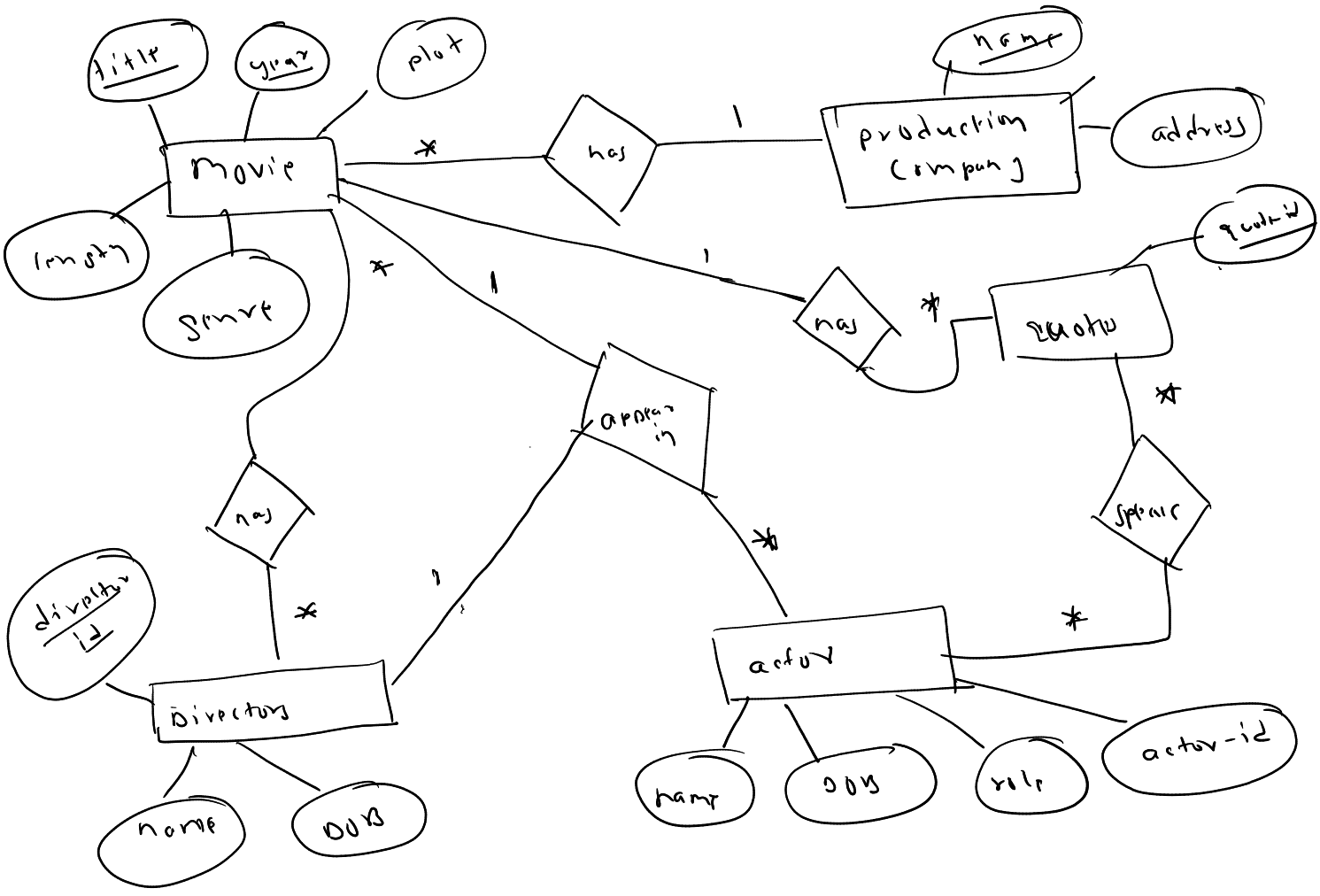

member_id
email
name
password
home address
Member
phone no
ISA
bank_acc no
rating
Seller
feedback
comment
Buyer
shipping address
routing no
transaction
lists
bids
bid_price
bid_time
item_id
category
item
item title
description
starting bid price
bid increment
end auction date
start auction date

3. An University Registrar's Office maintains data about the following entities:
   i. Courses, including number, title, credits, syllabus, and prerequisites;
   ii. Course Offerings, including course number, year, semester, section number, instructors, timings, and classroom;
   iii. Students, including student id, name, and program;
   iv. Instructors, including identification number, name department, and title.

   Further, the enrollment of students in the course and the grade awarded to students in each course they are enrolled in must be appropriately modeled. Construct an E-R diagram for the Registrar's office. Document all assumptions that you make about the mapping constraints. [2078 Chaitra]



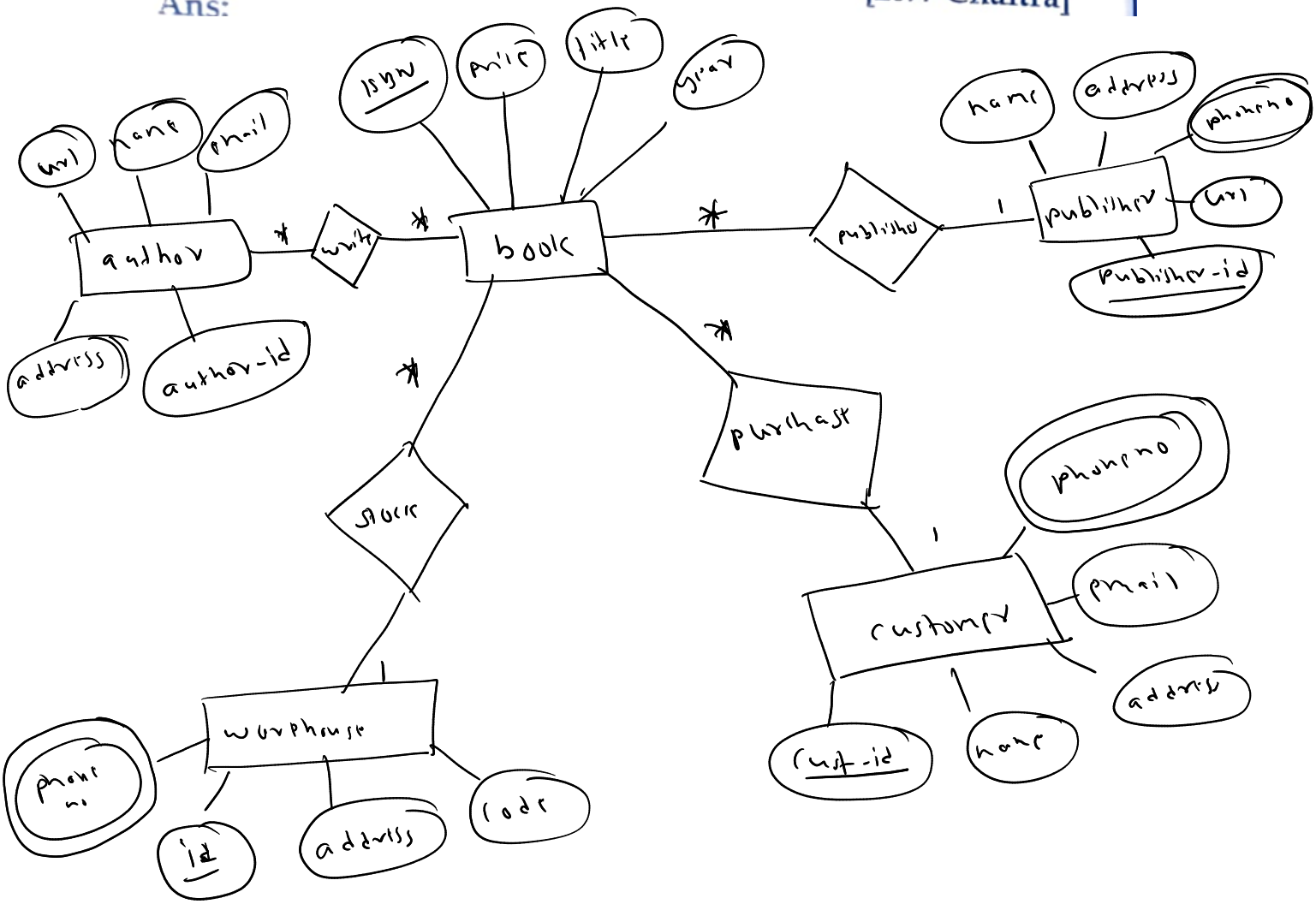
4. List the entities and their relationships and draw an ER diagram for the movie database. Consider a MOVIE database in which data is recorded about the movie industry. The data requirements are summarized as follows: "Each movie is identified by title and year of release. Each movie has a length in minutes. Each has a production company and each is classified under one or more genres (such as horror, action, drama and so forth). Each movie has one or more directors and one or more actors appear in it. Each movie also has a plot outline. Finally, each movie has zero or more quotable quotes, each of which is spoken by a particular actor appearing in the movie. Actors are identified by name and date of birth and appear in one or more movies. Each actor has a role in the movie. Directors are also identified by name and date of birth and direct one or more movies. It is possible for a director to act in a movie (including one that he or she may also direct). Production companies are identified by name and each has an address. A production company produces one or more movies." [2078 Bhadra]



5. Draw an ER diagram for an online book store that contains written by authors and published by the publisher. Also the book is stocked in the warehouse. The store also maintains record about the selling of books online via shopping basket (shopping cart), ie. customers purchase the books online via the shopping basket. Books have ISBN, price, title and year whereas authors may have url, name, address and email. Also the publishers name, address, phoneno and url are stored. The customer's name, address, email and phone are stored. The warehouse may have code, phoneno and address. Assume other parameters accrodingly.

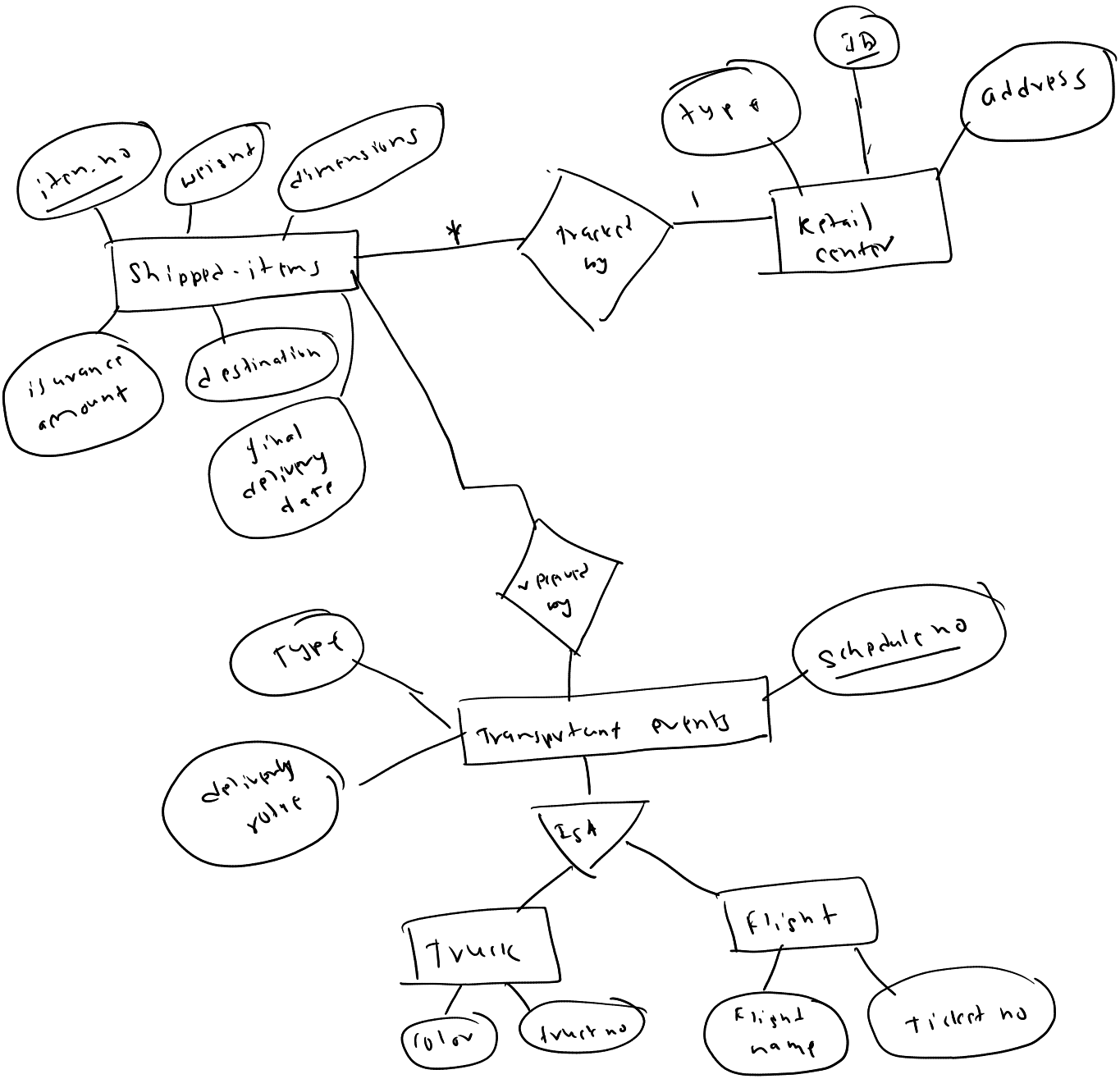
[2077 Chaitra]

Ans:



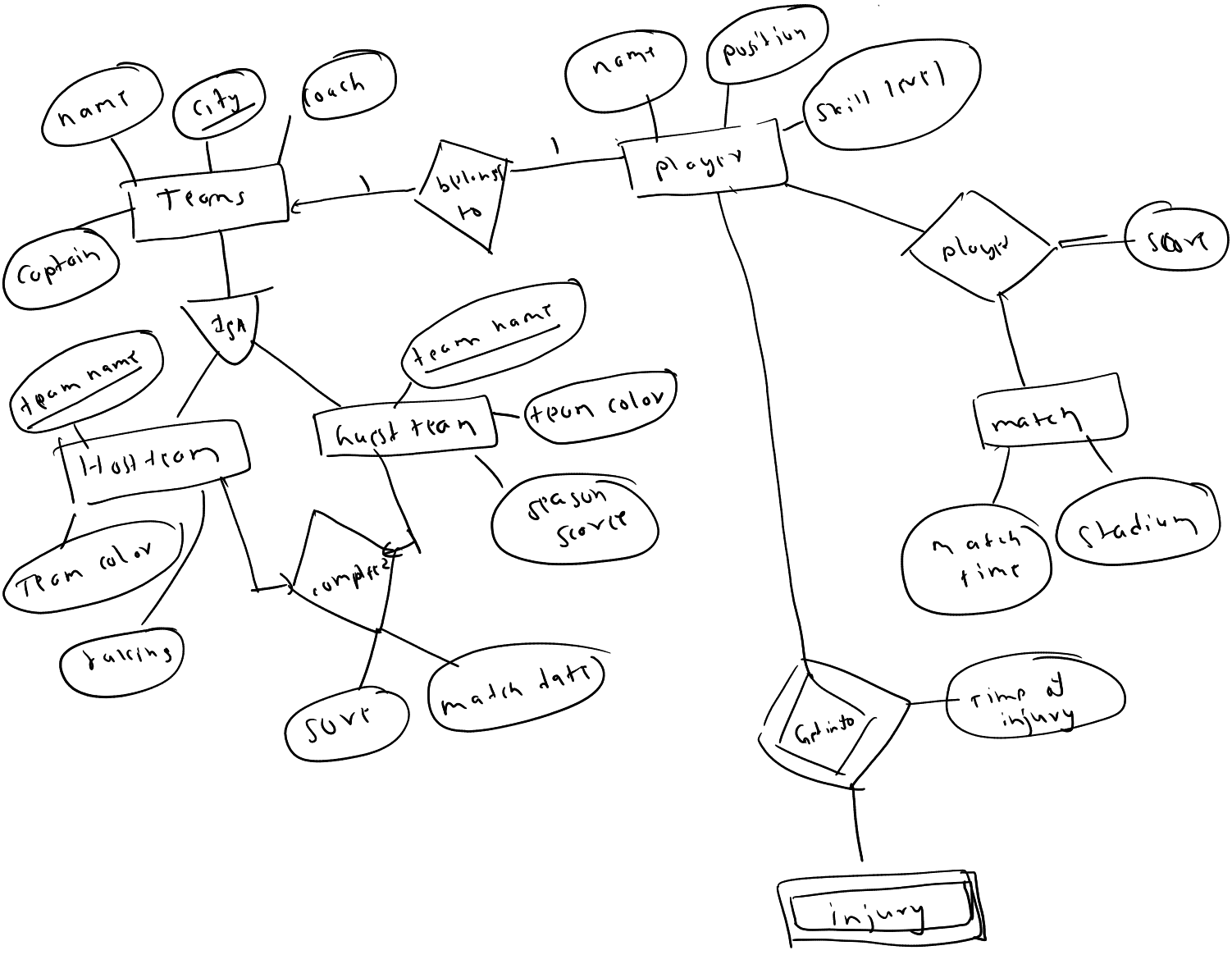
6. UPS prides itself on having up-to-date information on the processing and current location of each shipped item. To do this, UPS relies on a company-wide information system. Shipped items are the heart of the UPS product tracking information system. Shipped items can be characterized by item number (unique), weight, dimensions, insurance amount, destination, and final delivery date. Shipped items are received into the UPS system at a single retail center. Retail centers are characterized by their type, uniqueID, and address. Shipped items make their way to their destination via one or more standard UPS transportation events (i.e., flights, truck deliveries). These transportation events are characterized by a unique scheduleNumber, a type (e.g, flight, truck), and a deliveryRoute.

Ans:



You are given the requirement for a simple database for the National Football League (NFL). The NFL has many teams, and each team has a name, a city, a coach, a captain and set of players. Each player belong to only one team and each player has a name, a position (such as left wing, mid fielder or a goalkeeper) a skill level and a set of injury records. A team captain is also a player and a game is played between two teams (referred as host team and guest team) and has a match date(such as June 11,2018) and (score such as 2 to 5).

ns: [2075 Baisakh]



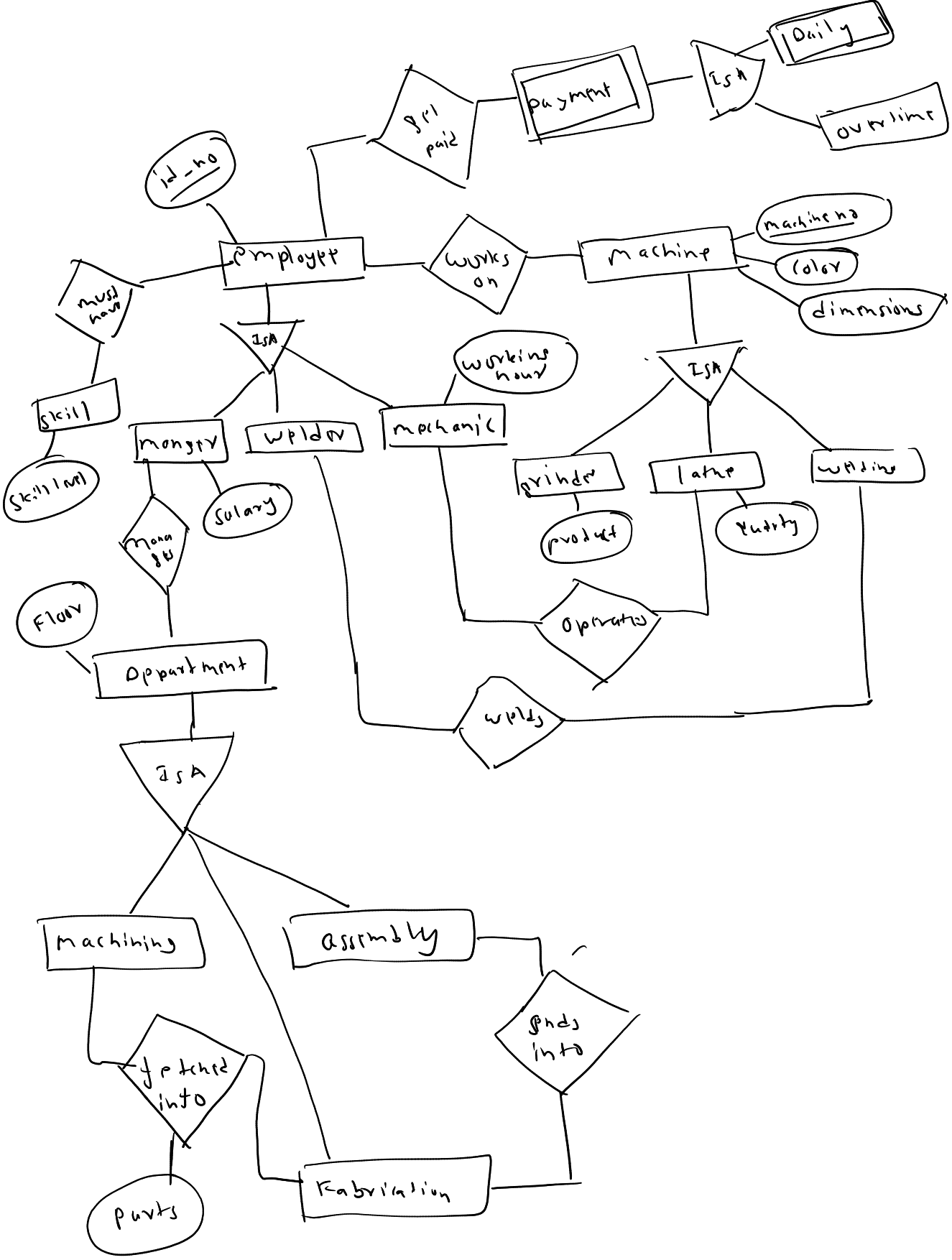
8. A Production company consists of a machining, fabrication and assembly department. Employees are assigned to different departments. Each department is managed by a manager. Each employee has at most one recognized skill, but a given skill may be possessed by several employees. An employee is able to operate a given machine type (e.g., lathe, grinder, welding) of each department. Some of the employees are paid overtime and some of them are paid with daily basis. According to their designation (e.g., machine, welder) are supposed to maintain at least one machine type of

their department. Raw materials are bought from different vendors and fetched to the machining department. Parts from machining department are fetched to fabrication department and so on. Many parts are assembled together to form a product. The final products from assembly department are stored in the ware house. Products are labeled with different specifications (e.g., Product_ID, Product_type, MRP etc).

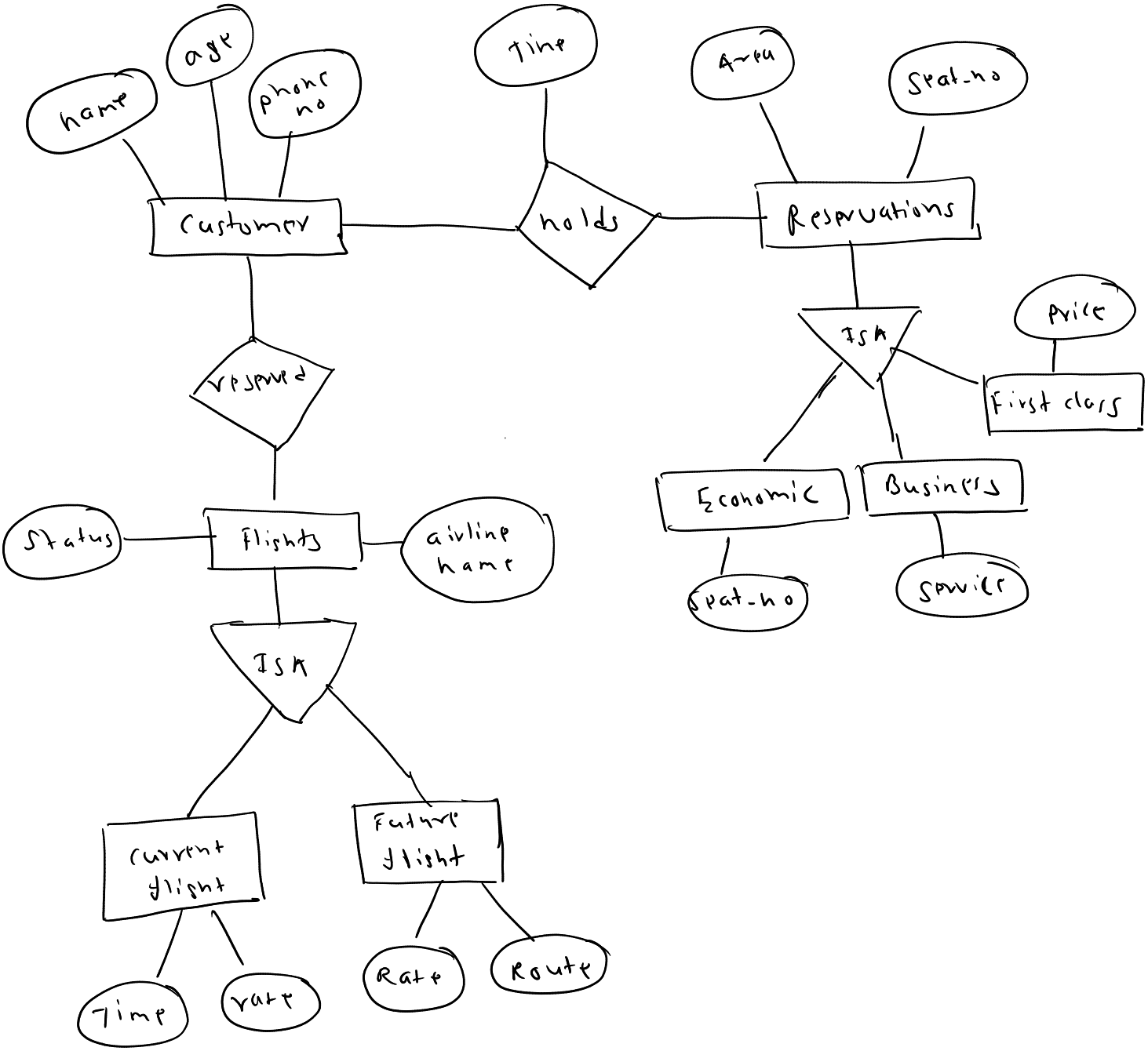
[2074 Bhadra]

Ans:

p.no


Daily
payment
ISA
overtime
id_no
Employee
works on
machine
machine no
color
dimensions
must have
ISA
working hours
ISA
skill
skill lvl
mangr
welder
mechanic
grinder
lathe
welding
salary
product
duty
Manages
Floor
Operates
Department
welds
ISA
machining
assembly
fabs into
fetched into
parts
Fabrication

Design an ER diagram for a database for an airlines system. The database must keep track of customers and their reservations, flights and their status, seat assignments on individual flights and schedule and routing of future flights. Apply all the database design constraints as much as possible. [2073 Magh]



Draw an ER diagram for the airport database. Be sure to indicate the various attributes of each entity. Every airplane has a registration number and each airplane is of specific model. The airport accommodates a number of airplane model and each model is identified by a model number (eg DC-10) and has a capacity and a weight. A number of technicians works at the airport. You need to store the name, SSN, address, phone number and salary of each technician. Each technician is an expert on one or more plane model(s) and his or her expertise may overlap with that of other technicians. This information about technicians must also be recorded. Traffic controllers must have an annual medical examination. For each traffic controller you must store the data of the most recent exam.