DBMS

2020-21

2(b)

Strong entity set can exist independently and has its own primary key.

Example: Bank with attributes like bank_id (PK) and name.

weak entity set depends on a strong entity set for its existence. It does not have a primary key of its own. It uses a foreign key from the related strong entity, along with its own attributes, to form a composite key.

Example: Account related to a bank, with attributes account_number and balance.

Relationship between weak and strong entity:

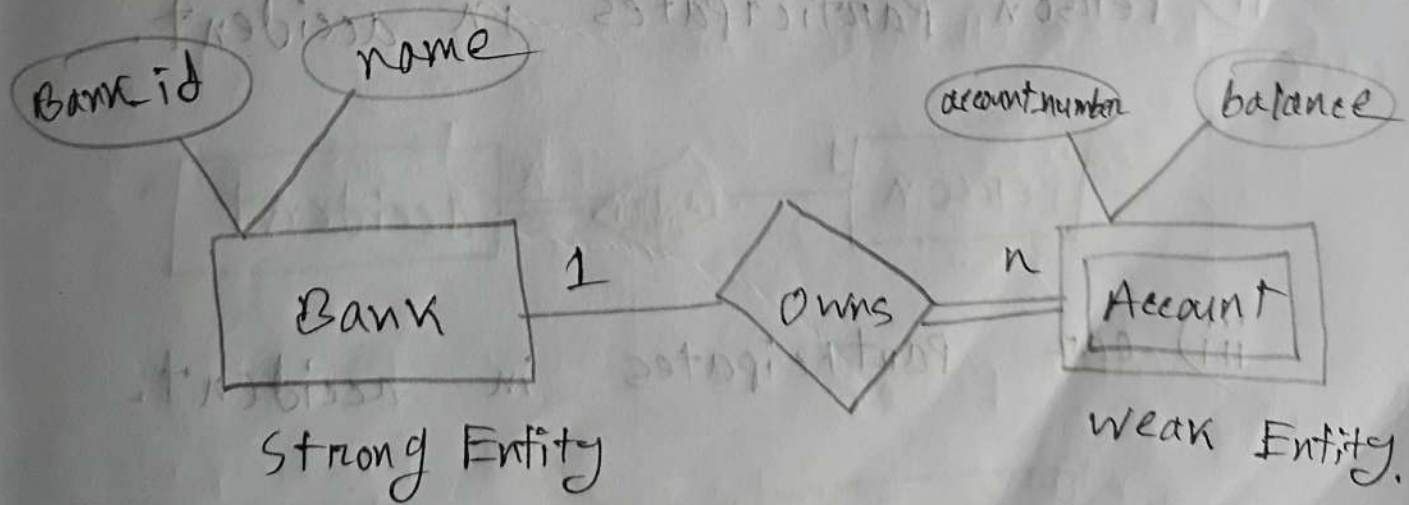
entities: i) Bank (strong entity)
ii) Account (weak entity)

Attributes: Bank; Bank_id, name.

Account: account_number, balance.

Relationships: i) Bank owns account.

ER Diagram:



2 (d)

Relationships:

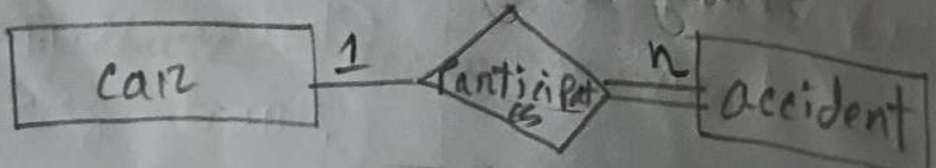
i) Person owns car.
ii) Person participates in accident.
iii) Car participates in accident.

Cardinality ratio and participation:

i) Person owns car.

ii) Person participates in accident.

iii) Car participates in accident.



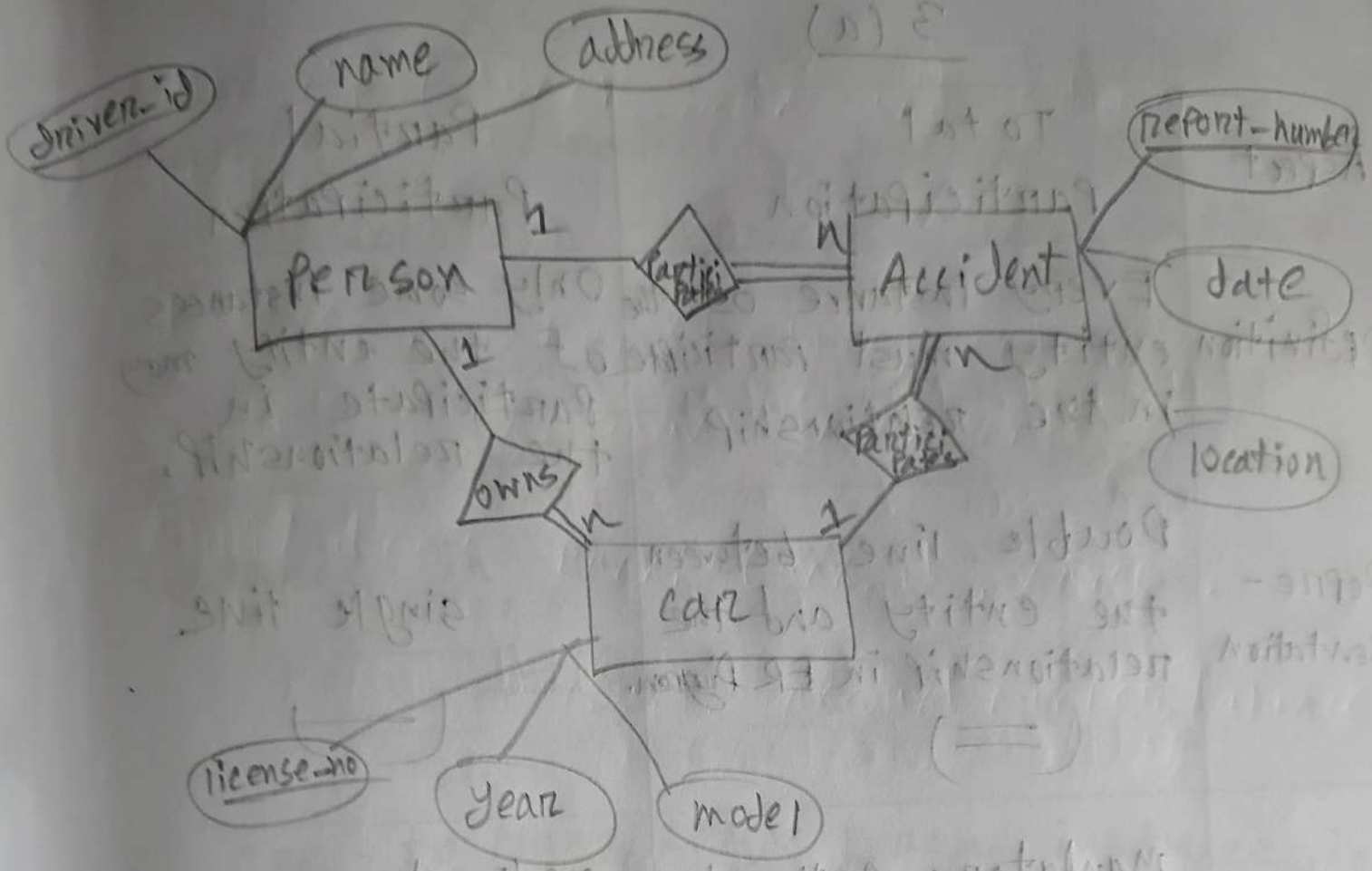


fig: ER Diagram

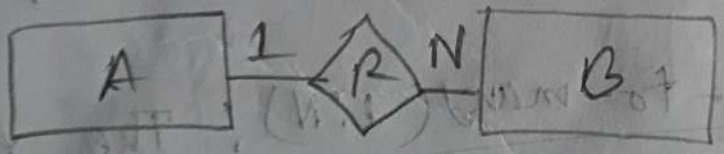
3 (a)

Aspect	Total Participation	Partial Participation
Definition	Every instance of the entity must participate in the relationship	Only some instances of the entity may participate in the relationship.
Representation	Double line between the entity and the relationship in ER Digram. (=)	single line. (—)
Requirement	mandatory participation for all instances	optional participation of some instances.
Use case	A "student" must be enrolled in one "course".	A "customer" may or may not have placed an "order".
Example	All "person" entities must own a "car".	Some "person" entities may participate in an "Accident".

i) Mapping cardinalities describe the type of association between two entities.

1) One-to-on (1:1): Each instance in A is associated with at most one instance in B, and vice versa.

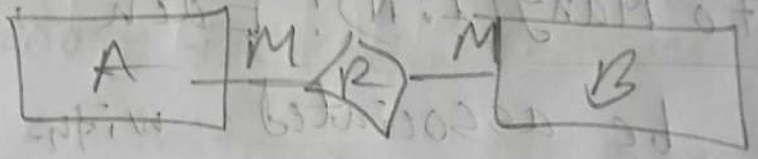
2) One-to Many (1:N): Each instance in A can be associated with multiple instances in B, but each instance in B is associated with at most one instance in A.



3) Many-to-One (N:1): Each instance in B can be associated with multiple instances in A, but each instance in A is associated with at most one

instance in B.

4) Many-to-Many (M:M): Each instance in A can be associated with multiple instances in B, and each instance in B can be associated with multiple instances in A.



ii)

1) One-to-one (1:1): Either the primary key of A or B can be used as the primary key for R.

2) One-to-many (1:N): The primary key of B and a reference to the primary key of A can define R.

3) Many-to-one (N:1): The primary key of A and a reference to the primary key of B can define R.

4) Many-to-Many (M:M): The primary key for R would be a composite key consisting of bothe the primary keys from A and B.

iii) combining tables can depend on the mapping cardinality:

1) one-to-one: tables A and B can often be combined into a single table, as there is a unique pairing.

2) one-to-many: Include a foreign key in the B table referencing A, but it is generally better to keep them separate due to the one-to-many relationship.

3) Many-to-one (N:1) Include a foreign key in the A table referencing B, while keeping them separate.

4) Many-to-Many (M:M): Use a separate associative (Junction) table to represent R, with foreign keys referencing both A and B tables.

4 (a)

Entities

i) Book
ii) Library
iii) Copy
iv) Borrower
v) ~~Loan~~

Attributes

Book: ISBN, Title, Authors.

Library: Name, Type.

Copy: copy-number, Price, loan_date.

Borrower: ID, Name.

~~Loan: Loan-Date,~~

Relationships

i) Library owns copy.

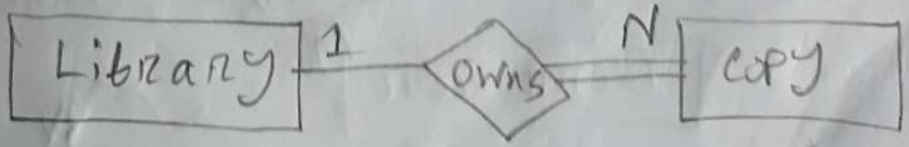
ii) ~~Main~~ Library manages Library.

iii) Borrower borrows a copy of book.

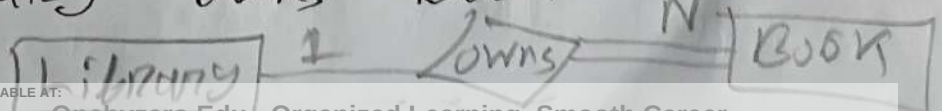
~~iv) Borrower takes loan~~

cardinality ratio and participation

i) Library owns copy.

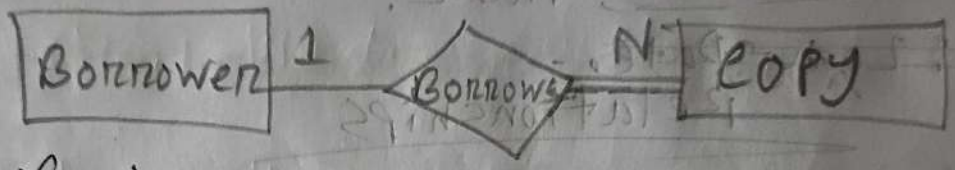


ii) Library owns Book

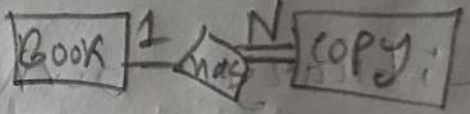


iii) Library manages Library

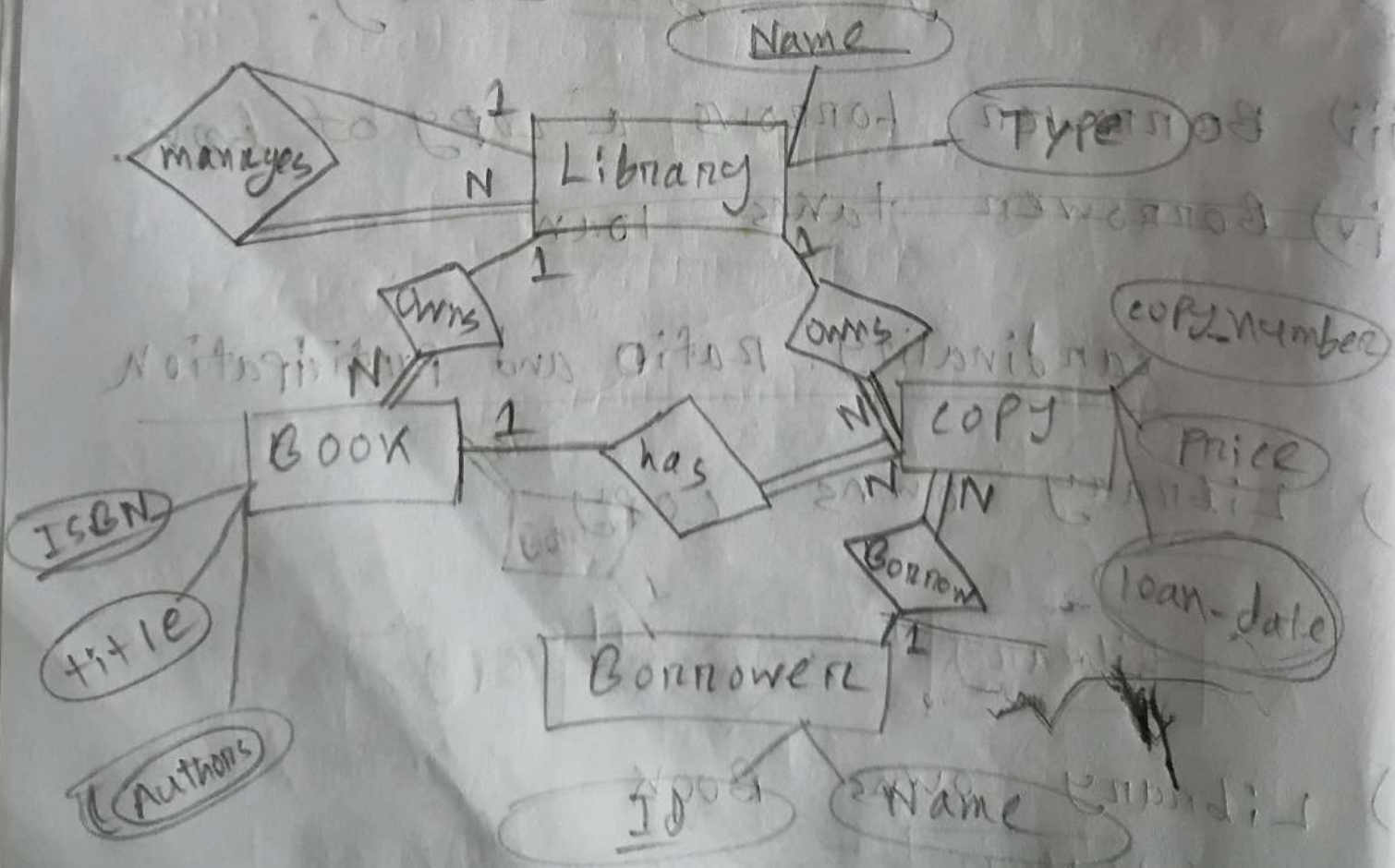
iv) Borrower Borrows copy



v) Book has copy



E·R Diagram



4(b)

Entities:

i) Lot
ii) ProductionUnits
iii) RawMaterials
~~iv)~~

~~Attributes~~ Schema:

i) Lot: LotNumber, CreationDate, Cost of Materi
ii) ProductionUnits: Unit DD, InstallDate, Producttype.
iii) RawMaterials: MaterialID, Type, Unitcost.

~~Relationship~~

iv) Includes: Lotnumber, Unit ID.

v) created from: LotNumber, Material ID, Units

Referential Integrity constraints:

Primary keys: Each table has a primary key to uniquely identify each row.

Foreign keys:

i) Includes.LotNumber references Lot.Lotnumber.

ii) Includes.UnitID references ProductionUnits.UnitID.

iii) CreatedFrom.LotNumber references Lot.LotNumber.

iv) CreatedFrom.MaterialID references RawMaterials.MaterialID.

2019-20

3(a)

Entities

i) Customer
ii) Order
iii) Book
iv) Review
v) Payment.

Attributes

i) Customer: CustomerID, Name, Email, Phone, Address.

ii) Order: OrderID, OrderDate, Total Amount.

iii) Book: BookID, Title, Author, Price, Stock.

iv) Review: ReviewID, Rating, ReviewDate.

v) Payment: PaymentID, PaymentDate, Amount, payment Method.

Relationships

i) customer places order.

ii) customer makes payment

iii) customer writes review

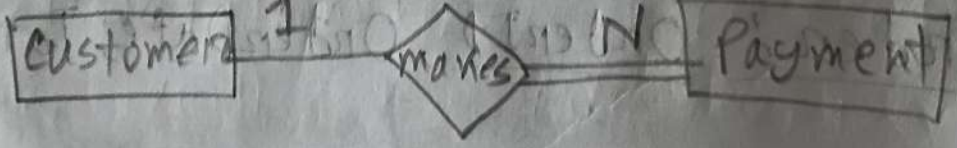
iv) order contains Book

v) Book has review

Cardinality Ratio and Participation ~~Relationship~~

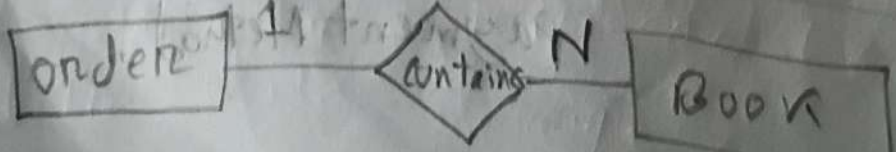
i) customer places order.

ii) customer makes payment.

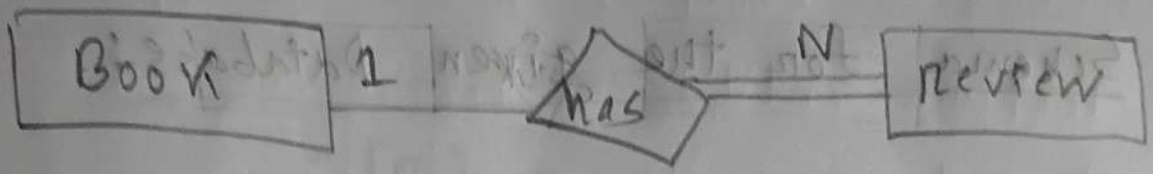


iii) customer writes review

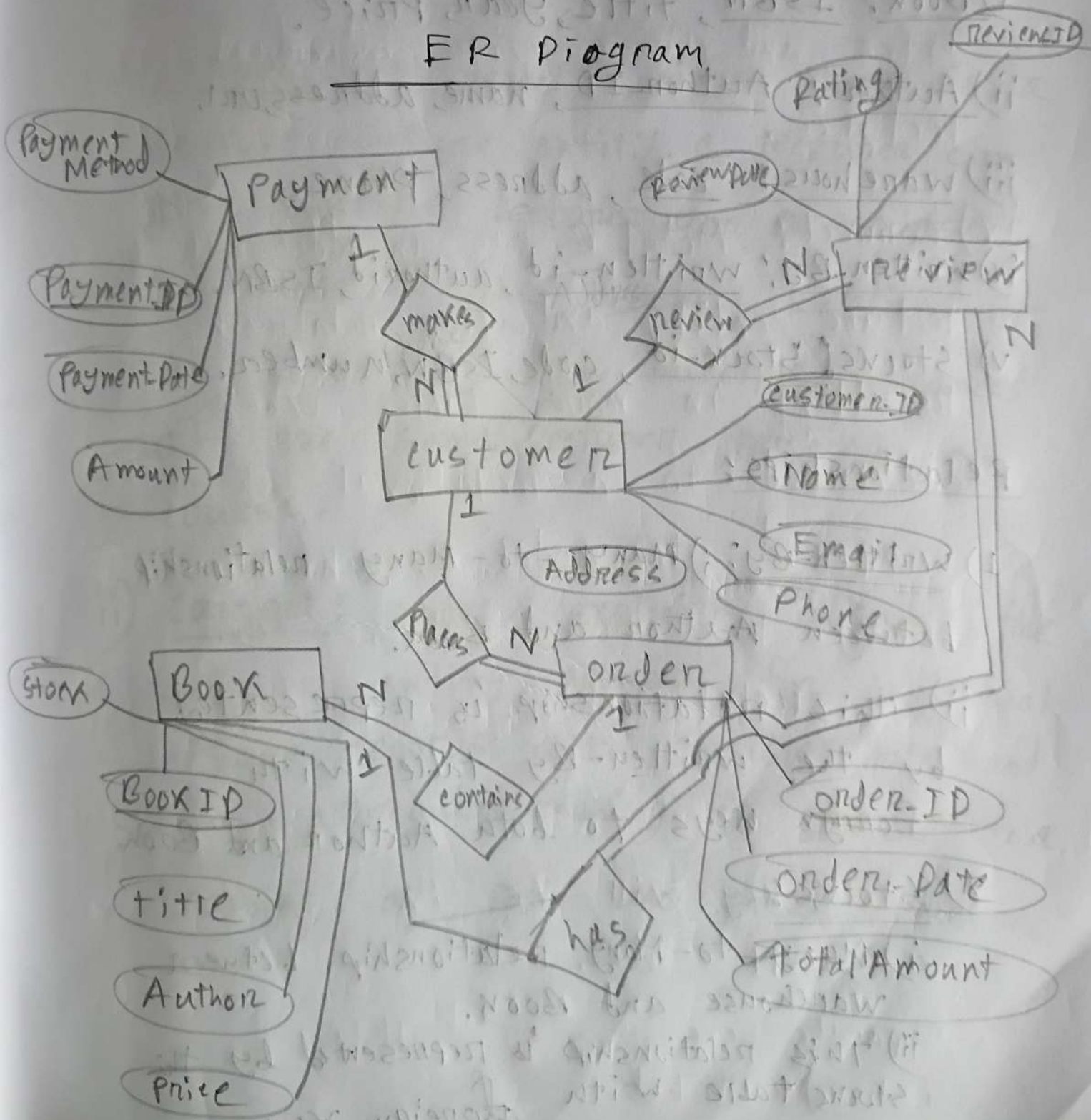
iv) order contains Book



v) Book has review.



ER Diagram



3(c)

Schema for the given Database:

i) Book: ISBN, title, years Price.

ii) Author: Author_ID, name, address, url.

iii) warehous: code, address, phone.

iv) written_By: written_id, authorid, ISBN.

v) stocks[Stock-id, code, ISBN, number.

Relationships:

1) written-By: i) Many-to-Many relationship between Author and Book.

ii) This relationship is represented by the written-By table with foreign keys to both Author and Book.

2) Stocks:

i) Many-to-Many relationship between warehouse and Book.

ii) This relationship is represented by the stocks table with foreign keys

to both warehouse and Book.

4(a)

i) connect: In Model B, since there is a distinct lecture entity, a lecturer can give multiple lectures for a course. In Model A, the teaches relationship can potentially allow multiple instances of the same lecturer teaching a course.

ii) connect; Model B explicitly includes the lecture entity, allowing for individual lecture records with possible date and time attributes, making it suitable for tracking all past and present lectures.

iii) Correct: Model A lacks a separate lecture entity, so there is no direct way to associate specific dates and times with individual lectures.

iv) Correct: Model B has an additional lecture entity and two relationships, resulting in more tables when converted to a relational schema compared to model A, which only has two entities and one relationship.

---

* foreign key will be in (many) N side.

* separate schema for double valued attributes.

4(c)

A composite attribute is an attribute that can be broken down into smaller sub-parts, each with independent meaning.

here, createdDate could be a composite attribute because it may contain parts such as day, month and year.

4(d)

Data redundancy occurs when the same piece of information is stored in multiple places, leading to inconsistency.

In this ER diagram, a redundancy issue might exist between Raw Materials and Production Units. Specifically, attributes like Unit of measurement could be duplicated across these entities if they are being managed in both places.

17-18

4(b)

Entities

i) Employee

ii) Department

iii) child

Attributes

i) Employee: ssn, name, salary, phone

ii) Department: dno, dname, budget.

iii) child: name, ssn, age.

Relationship

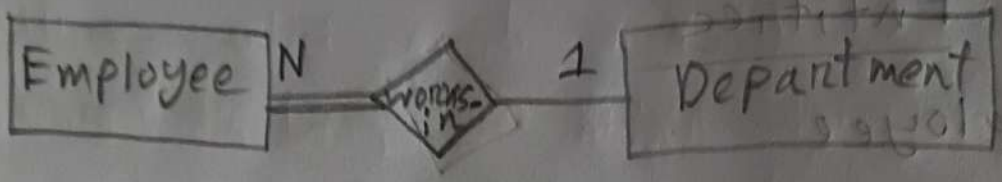
i) Employee works_in Department.

ii) Employee manages Department.

iii) Employee has child.

Cardinality Ratio and Participation

i) Employee works_in Department.



ii) Employee manages Department.

iii) Employee has child.

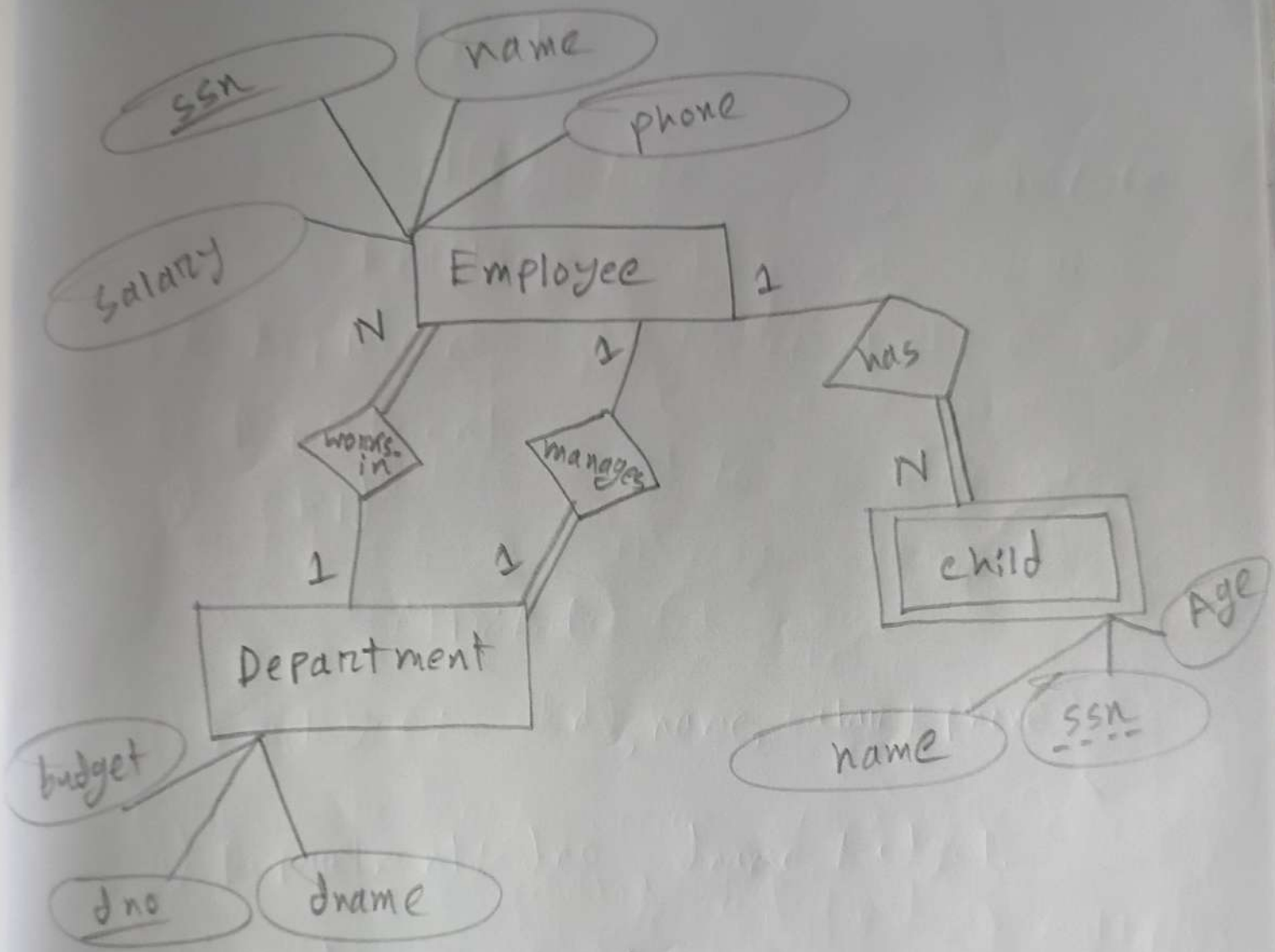


fig: ER Diagram.