Lecture-1
Introduction to Database System

Basic Definitions

Data

Recorded facts with implicit meaning

Database

Coherent collection of related data built for a purpose.

Mini-world

A segment of the real world stored in the database. [Ex → student grades and transcripts at a university]

Database Management System (DBMS)

Software for creating/maintaining database

Database System

DBMS software plus data (may include applications).

Types of Databases and Databases App.

Traditional Apps → Numeric and textual databases

More Recent Apps

(i) Multimedia db

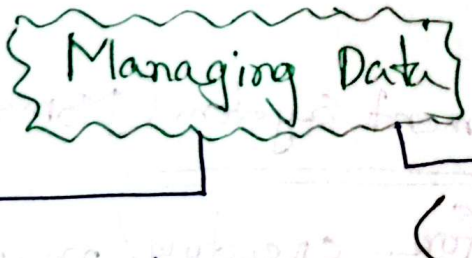
(ii) Geographic Information systems (GIS)

(iii) Biological and genome db

(iv) Data Warehouses

(v) Mobile db

(vi) Real time and active db



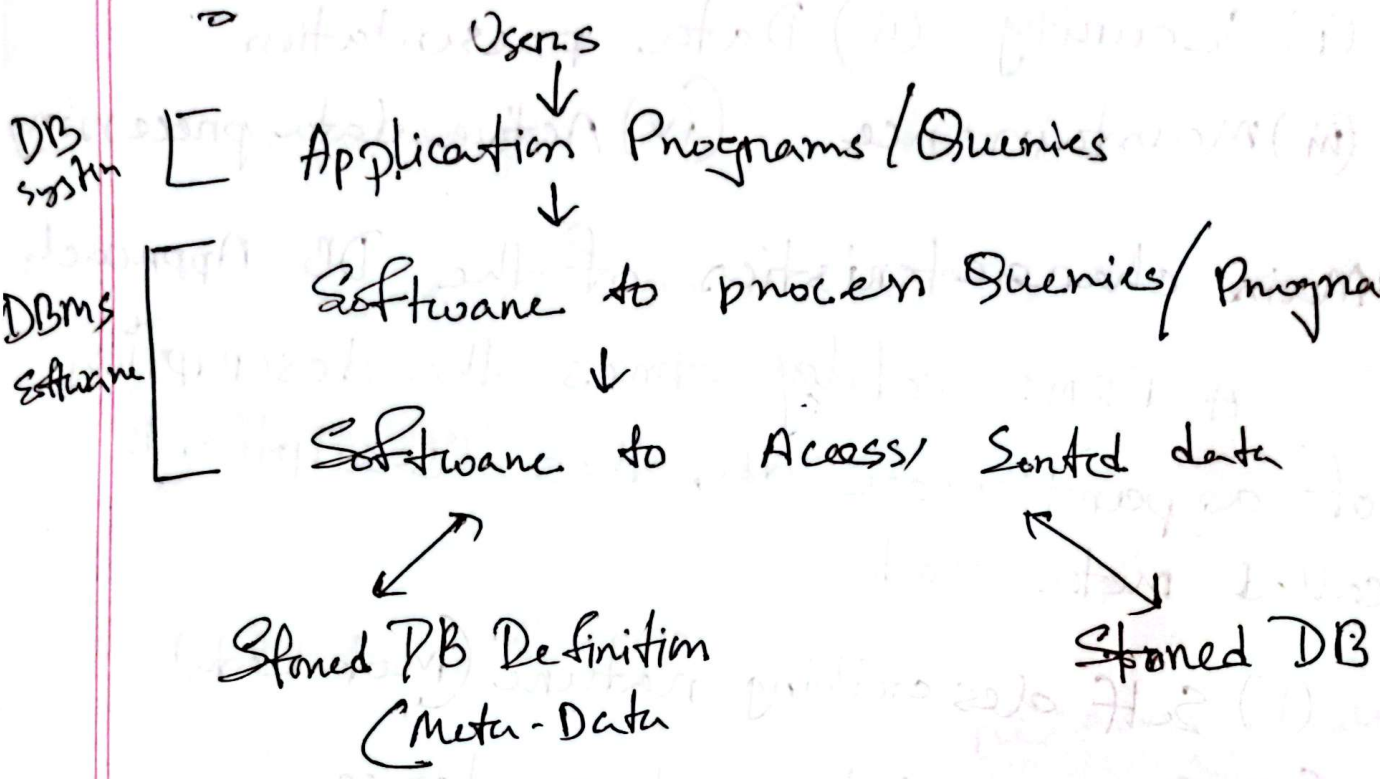
File Based Approach

Uses specific application programs which performs services to end users, each managing its own data

Database Approach

Uses DBMS for data collection and sharing among multiple programs.

A simplified db system environment



DBMS Facilates

(i) Defining DB's

(ii) Construct or load the initial db contents on a secondary storage medium

(iii) Data retrieval, modification, web access.

(iv) Multi-user processing

DBMS Functionalities

(i) Security (ii) Data presentation
(iii) Maintainance (iv) Active data processing

Main characteristics of the DB Approach

A DBMS catalog stores the description of a particular db, this descriptin is called meta-data

(i) Self describing nature (Meta-data)
(ii) Program data independence
(iii) Data abstraction
(iv) Support for multiple data views
(v) Multi users transactions
(vi) Concurrency control.
(vii) Restricting unauthorized access to data
(viii) Providing storage structures for efficient query processing.

OLTP (Online Transaction Processing) is a major part of db applications.

Advantages

(i) Data redundancy control

(ii) Data sharing

(iii) Restricted Access

(iv) Query optimization

(v) Providing backup and recovery services.

(vi) Providing multiple interfaces to different classes of users

(vii) Representing complex relationships among data.

(viii) Enforcing integrity constraints on the db.

Database Users

Actors on the Scene

যারা db use করে

Workers behind the scene

DBMS তৈরি and যারা রক্ষণাবেক্ষণ করে।

Actors on the scene

① Database Administrators: DB তে access প্রদান and নিয়ন্ত্রণের দায়িত্ব থাকে। DB এর নিরাপত্তা, কার্যকারিতা এবং Data সুবিন্যস্তকরণে দায়িত্ব থাকে। They update and control DB.

② Database Designers: They design db and the structure. Db এর logical and physical design তৈরি করে- according to end-users

③ Software Engineers: DB ভিত্তিক apps তৈরি করে এবং apps এর সাথে db এর connection নিশ্চিত করে।

DB End Users

They use the data for queries, reports and some of them update the db content. End-users can be categorized into:

(i) Casual Users: এরা মাঝে মাঝে db ব্যবহার করেন. তবে আগ্রহের অতিরিক্ত তথ্য সংগ্রহের জন্য Query-র সাহায্য নেন।

(ii) Naive: এরা primary users. যারা দায়িত্বের প্রয়োজনে app স্ক্রিন or interface ব্যবহার করে db access করেন।

(iii) Sophisticated: They know what they're doing. এরা db তে জটিল query চালাতে পারেন এবং উন্নত data বিশ্লেষণ করতে সক্ষম।

(iv) Stand-alone: Mostly maintain personal db using ready-to-use packaged apps.

When not to use DBMS

(i) Too costly

(i) High setup costs and Hardware needs.

(ii) Overhead for security, recovery and concurrency.

(ii) Unnecessary

(i) simple, unchanging db.

(ii) If access to data by multiple users is not required

(iii) Infeasible (অসম্ভব)

(i) Limited storage (embedded systems)

(iv) Won't work

(i) If real time speed needed (Phone systems)

(ii) Complex data (Genome db)

(iii) If special tasks needed to be handled (GIS tools)

Data Models

A set of concepts to describe the structure of a db, the operations for manipulating these structures, and certain constraints that the db should obey.

Categories

(i) Conceptual (High-level, Semantic)

এটা ব্যবহারকারীদের দৃষ্টিভঙ্গি থেকে data কে উপস্থাপন করে. [Entity-based বা obj based মডেল]

(ii) Physical (low-level, Internal)

কম্পিউটারে how data is being secured. এগুলো সাধারণত DBMS design ও administration এর জন্য উদ্দেশ্য করা হয়

(iii) Implementation (Representational)

In between Conceptual and Physical Model.

Business Related DBMS এ use হয় বেশি

(iv) Self describing data Models

Combine the description of data with the data values. [XML, key-value stone]

Database Schema

(i) The description of a db. (ii) Includes descriptions of the db structure, data types, and the ~~cont~~ constraints on the db.

(iii) Changes very infrequently

(iv) Schema is called intension.

Student

Name	number	class

Course

Name	credit	Dept

Database State

(i) The current data in the db at any time

(ii) Collection of all data in the db.

(iii) db state changes every time ~~the~~ db is updated

(iv) is called extension

Name	Number	Class
Rafi	01	10
Abnan	02	10
Faiyaz	03	10

Three Schema Architecture

- Proposed to support DBMS characteristics of.
  (i) Program-data independence
  (ii) Support of multiple views of the data
- has been useful in explaining db system organization.

(i) Internal Schema (Physical Level)

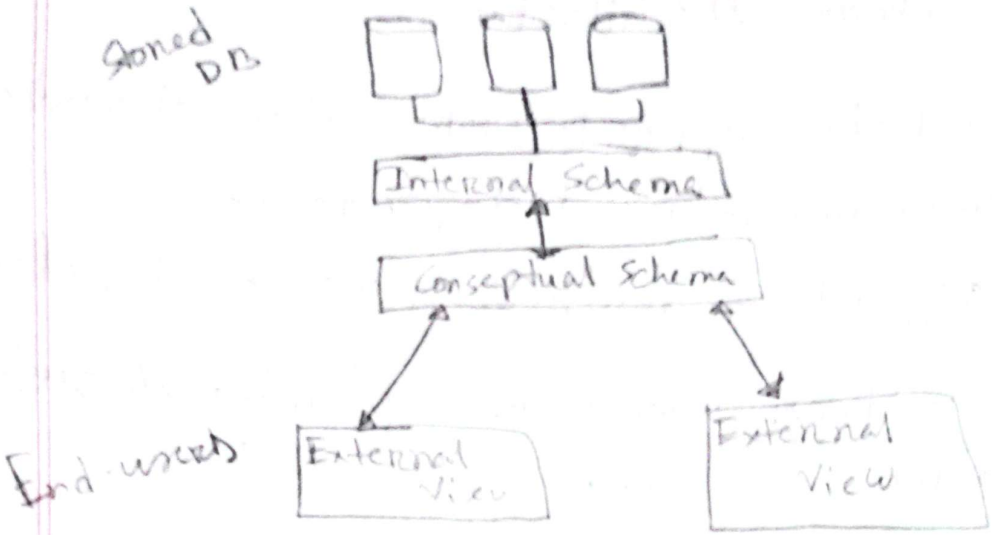
It is the physical storage of the data, detailing how data is stored in the computer.

(ii) Conceptual Schema. (logical level)

This level represents the overall structure of the entire db for the organization. It is like a blueprint showing all entities, relationships, and data types without concern for how it is physically stored.

(iii) External schema (View Level)

The external schema is the user's view of the db. It allows different users to see only the data relevant to them.



Data Independence

The ability to change the schema at one level without impacting the schema at the next higher level. 2 types.

(i) Logical Data Independence :

The capacity to change the conceptual schema without having to change the external schemas and their associated application programs. [db তে new column যোগ করলে, or datatype change করলে app এ কোন প্রভাব পড়ে না]

(ii) Physical Data Independence

Internal schema এর পরিবর্তন যাতে conceptual schema এর উপর প্রভাব না ফেলে. [db এর বিভিন্ন ফাইল স্টোর করার পদ্ধতি পরিবর্তন করা হয়, বা কোন new index create করা হয়, এগুলো conceptual schema unchanged রাখে.

DBMS Language

(i) Data Definition Language (DDL)

To define db schema. DDL ব্যবহার করে db এর বিভিন্ন কাঠামো যেমন table → creation, dropping tables, adding col and এটার datatype পরিবর্তন করা যায়।

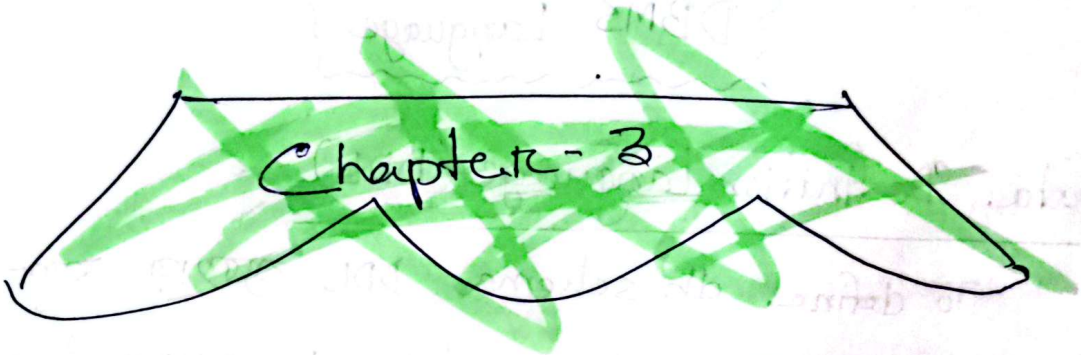
(ii) Data Manipulation Language (DML)

Used to manipulate data by inserting, deleting, updating and retrieving data. DML কোনো general-purpose a programming language (Java) এর সাথে embed করা হয়, এবং এর মাধ্যমে program থেকে db পরিচালনা করা যায়.

Ex → Insert, Delete, Update, Select → Data

DBMS Interfaces

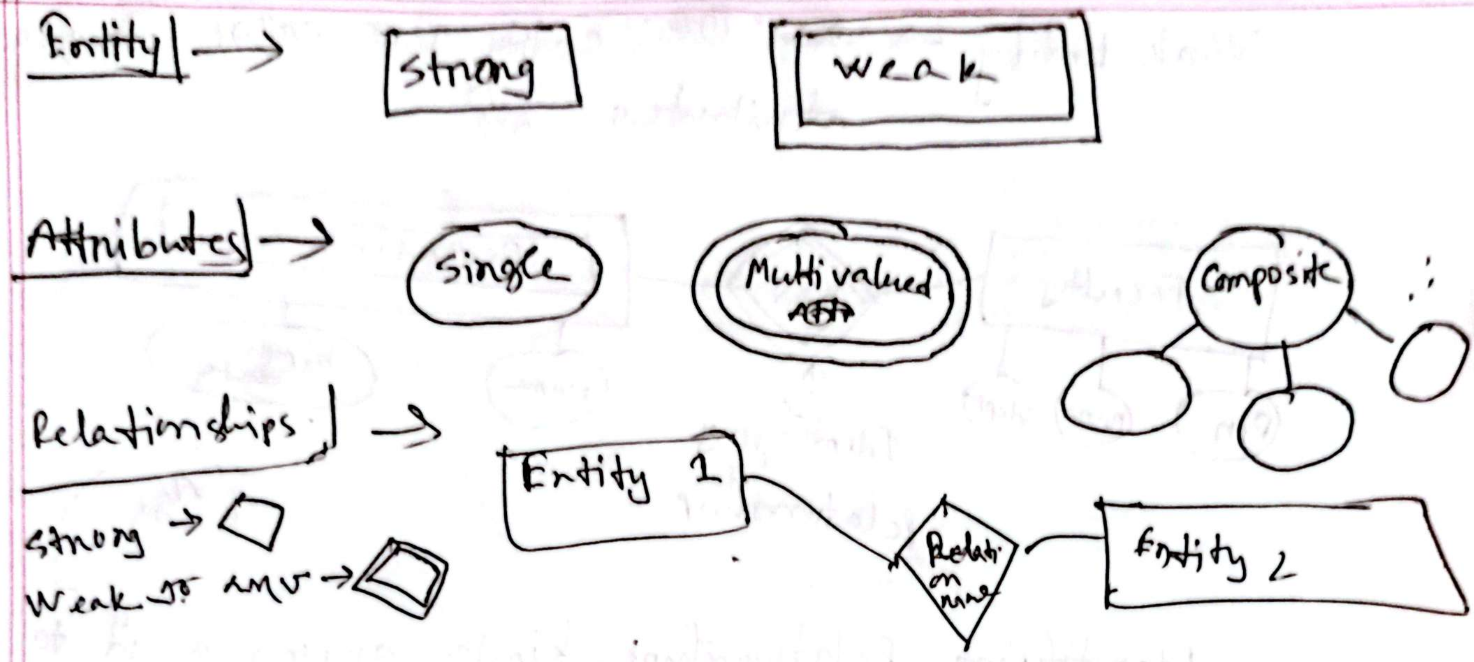
(i) Stand-Alone Interfaces
(ii) Programmer "
(iii) User-Friendly "
(iv) Mobile "
(v) Parametric "
(vi) DBA-specific "

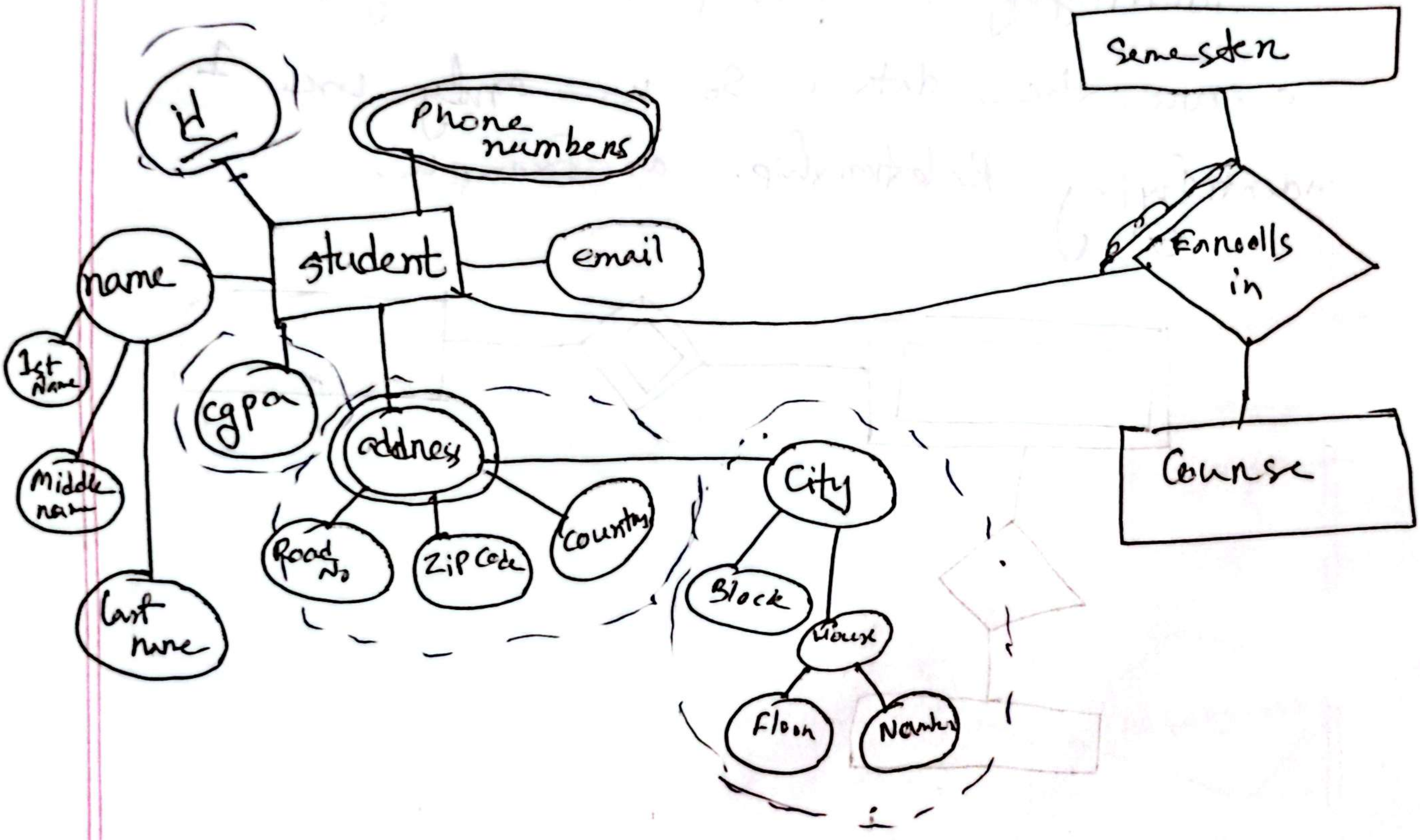


ER Diagram - Entity Relationship Diagram

Main important things →

1. Entity → যেকোনো obj যেটা আমরা db তে store করতে চাই. Regular/strong, weak.
2. ~~Attributes~~
2. Relationships → Between 2 entity
Binary, ternary, n-ary | strong, weak
3. Attributes. → Properties of Entity.
simple, Composite, multivalued





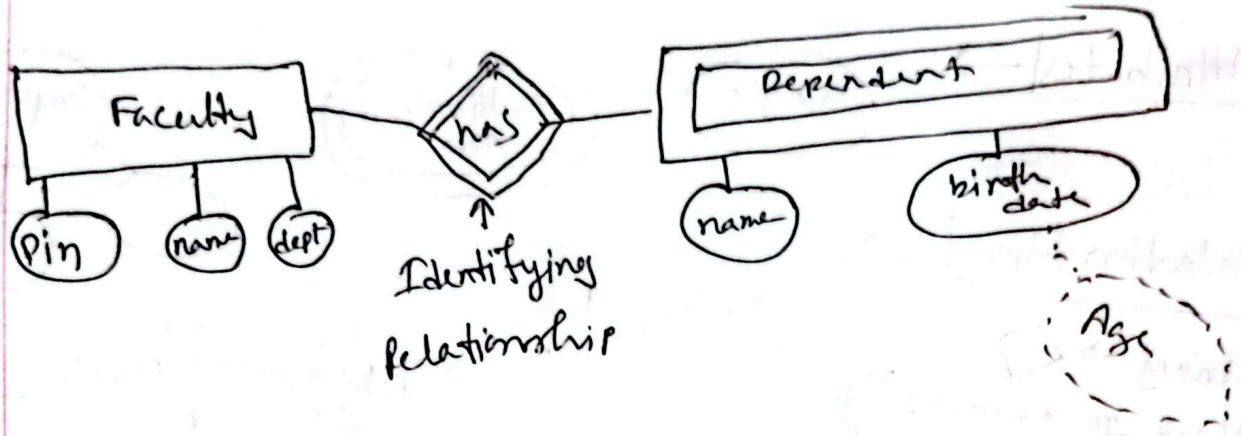
যেখানে যদি ১টি address দিতে হবে (add)

—— অনেকগুলো —— ((add))

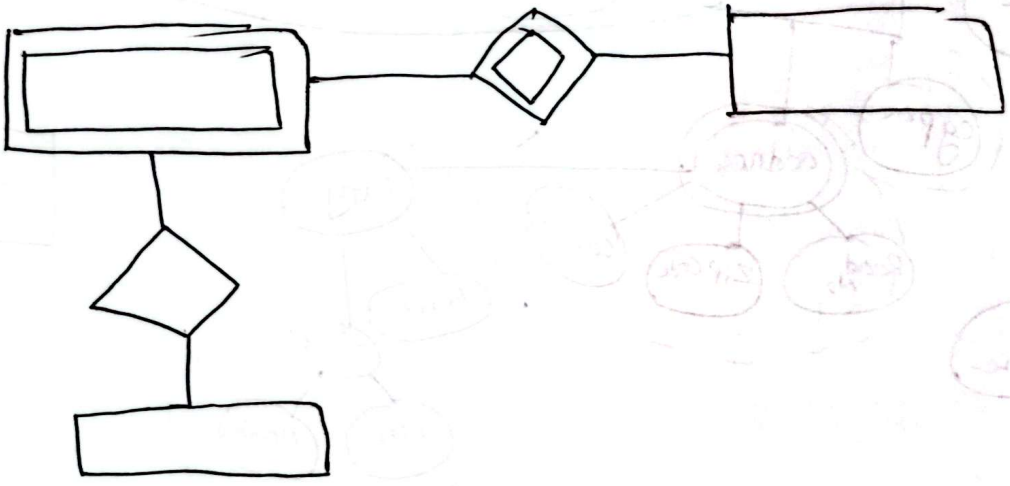
Key attribute

যেটা unique.
স্টুডেন্টদের ক্ষেত্রে যেমন id.
Underline হবে

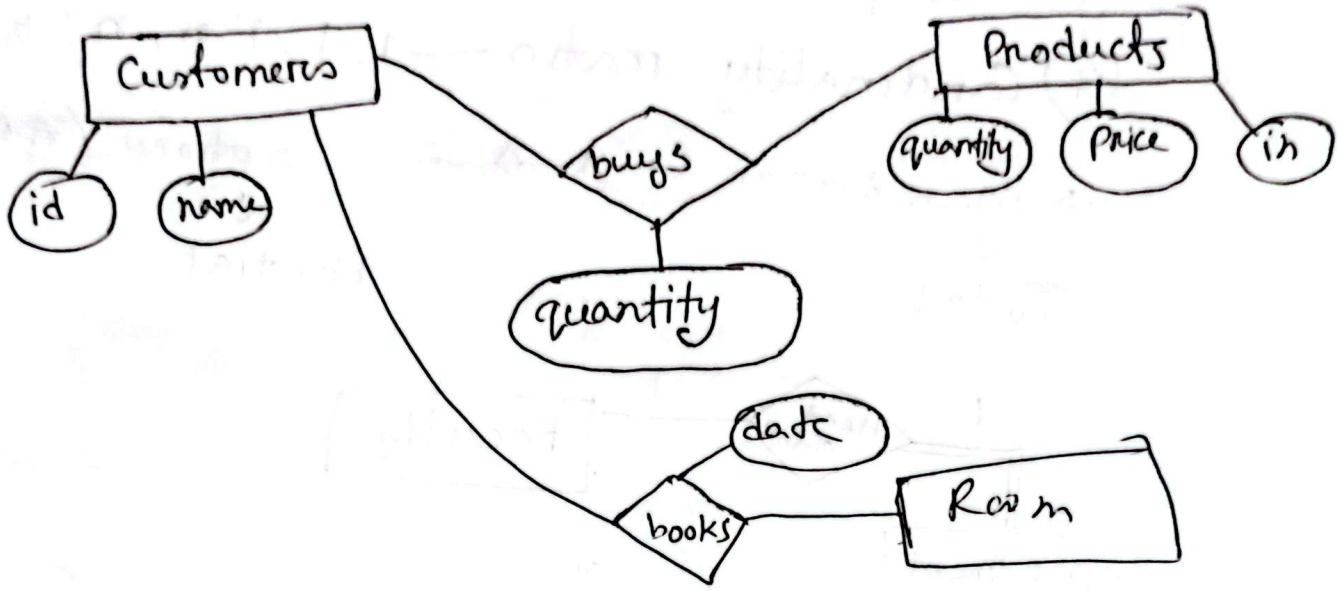
Weak Entity → যেটার নিজের entity তে কোনো unique attributes নাই



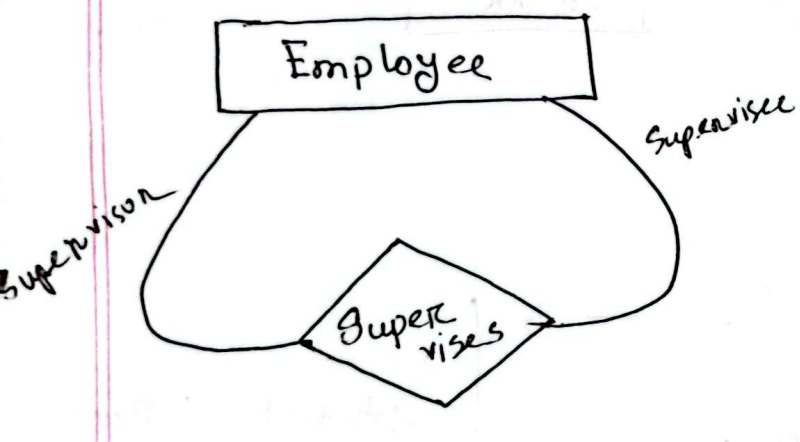
Identifying Relationships kinda assigns a id to secure the data. So we only use 1 identifying Relationship. as Example

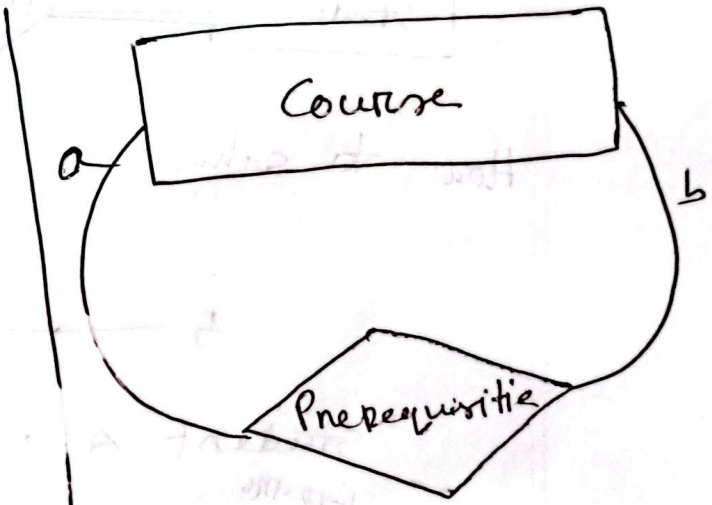


A relationship can have attributes too.



#Recursive Relationships





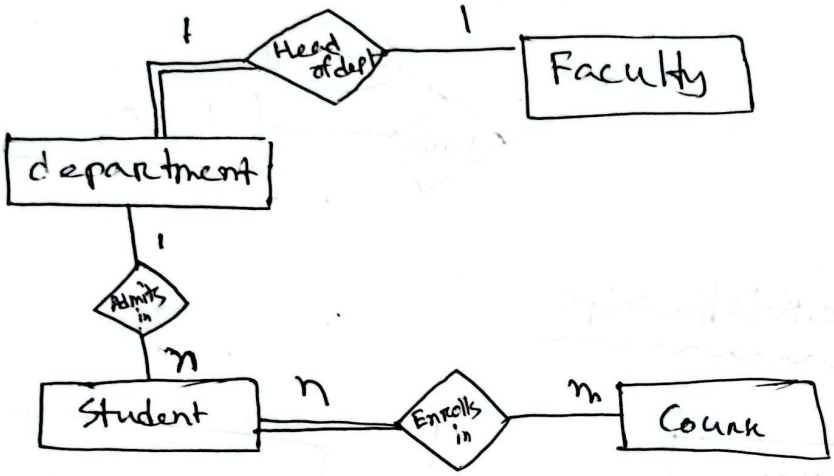
a → course to take before

b → course to take after

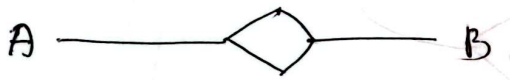
Constraints

(i) Participation constraint → Relationship টি কি mandatory নাকি Optional
(ii) Cardinality ratio — 1:1, 1:n, n:n

→ mandatory (A ═◇═ n) ; optional (A ─◇─ B)
↓ ↓
Total Partial



How to solve,



Student A : B course.

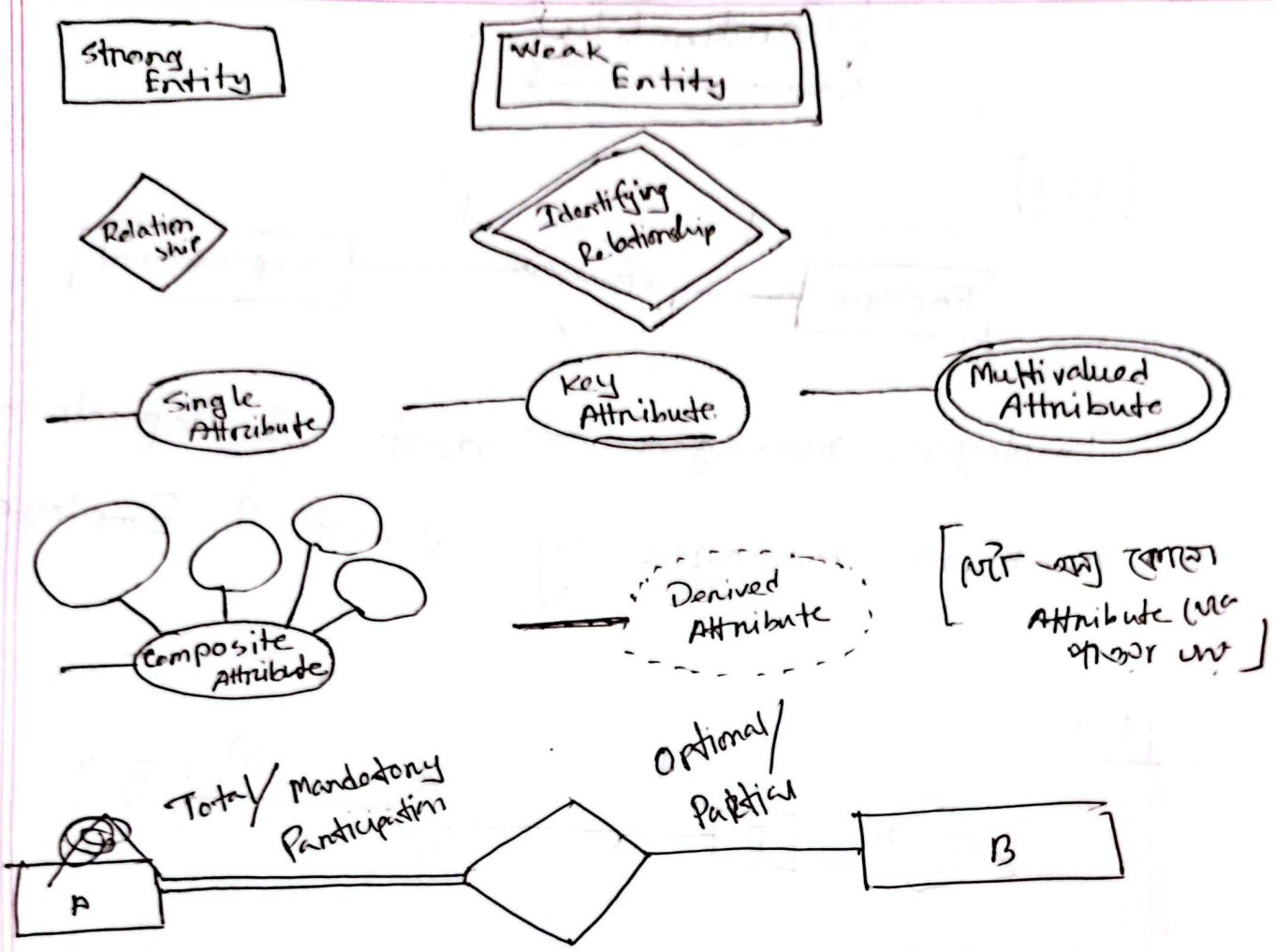
1 student course নিতে পারে
↳ 1 ⟶ n
n ⟵ 1
n : n

1 course নিতে পারে অনেক student

Student : Dept

1 ⟶ 1
n ⟵ n
n : 1

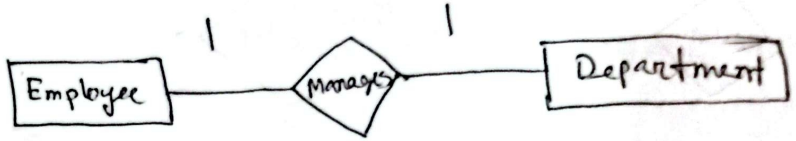
Summary of notation for ER diagram



ER Problems Solved

Cardinality

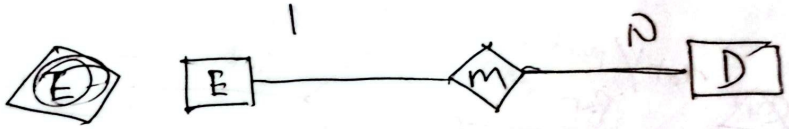
1 : 1



Employee manages at most 1 Department

Dept is managed by at most 1 Employee

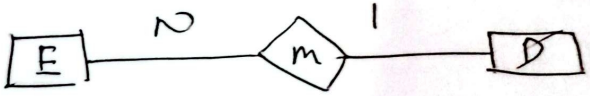
1 : N



E manage many ~~properties~~ dept.

dept is managed by only 1 E.

N : 1



~~Em~~ E manages only 1 Dept

Dept managed by many E

M : N

E

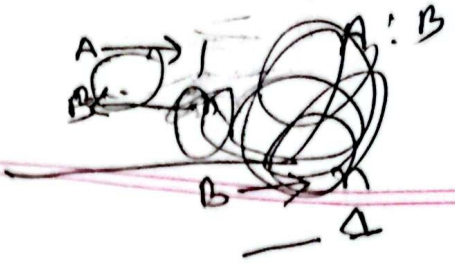
E manag

Cours

~~+ Student~~

+ Student

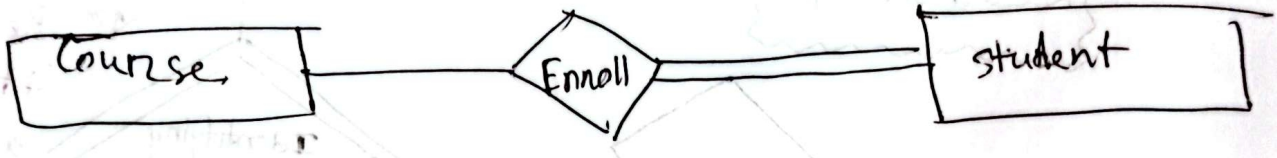
+ Course



M : N

E manages Many ; Dept managed by many

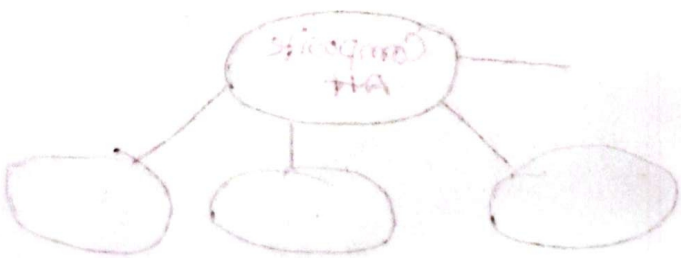
Constraints



~~* Student dependent on cou.~~

* Student কে must select করতে হবে।

* Course student কয়টি নিতে পারে তার [5 টা থেকে 3 টা নিতে 2টা course এর মত]

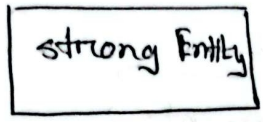


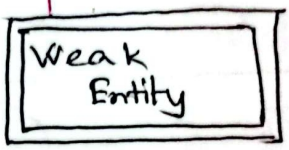
1. Ans

ER Model

~~Reality Sho~~

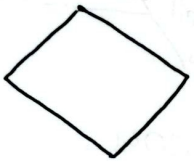
Entity

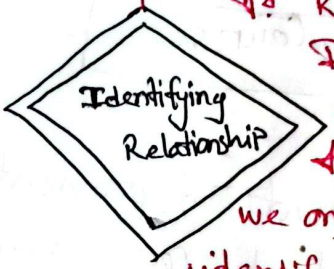




→ No ~~key~~ Unique / key attribute

Relationships

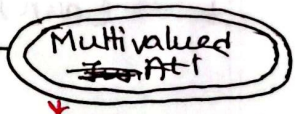




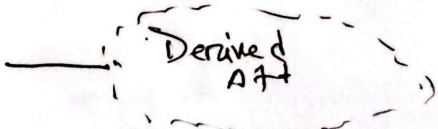
→ Weak ~~Attribute~~ Entity ↔ Relationship. It kinda assigns a id to secure the data. But we only use 1 identifying relationship

Attributes

→ Only 1 for (fixed) person



একাধিক এর att থাকতে পারে।

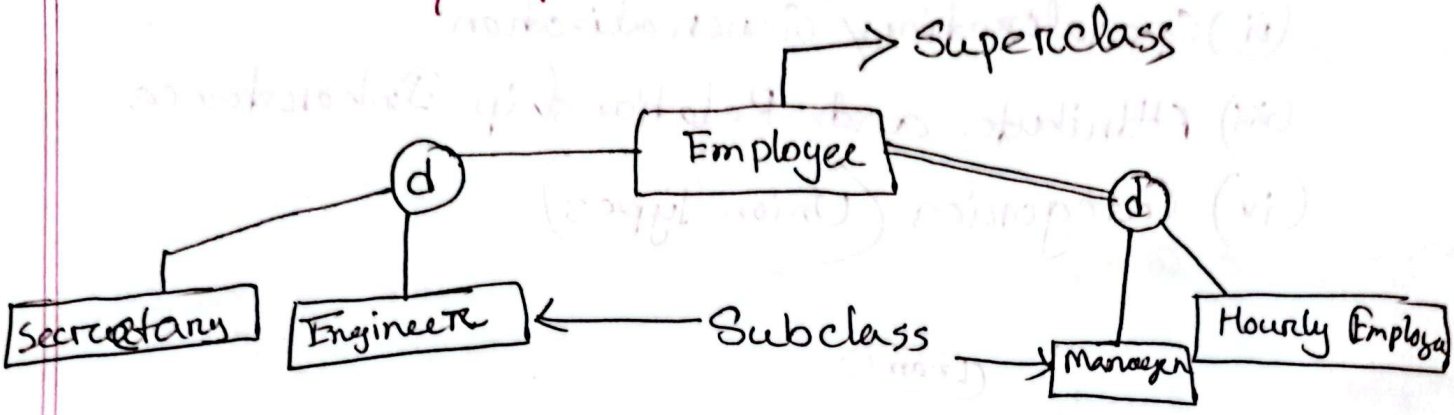


~~Att~~ যেকে এটা আমরা হিসাব করে নিতে পারি।

Chapter-4

EER → Extended ER

(i) Subclass / Superclass



Chapter - 4

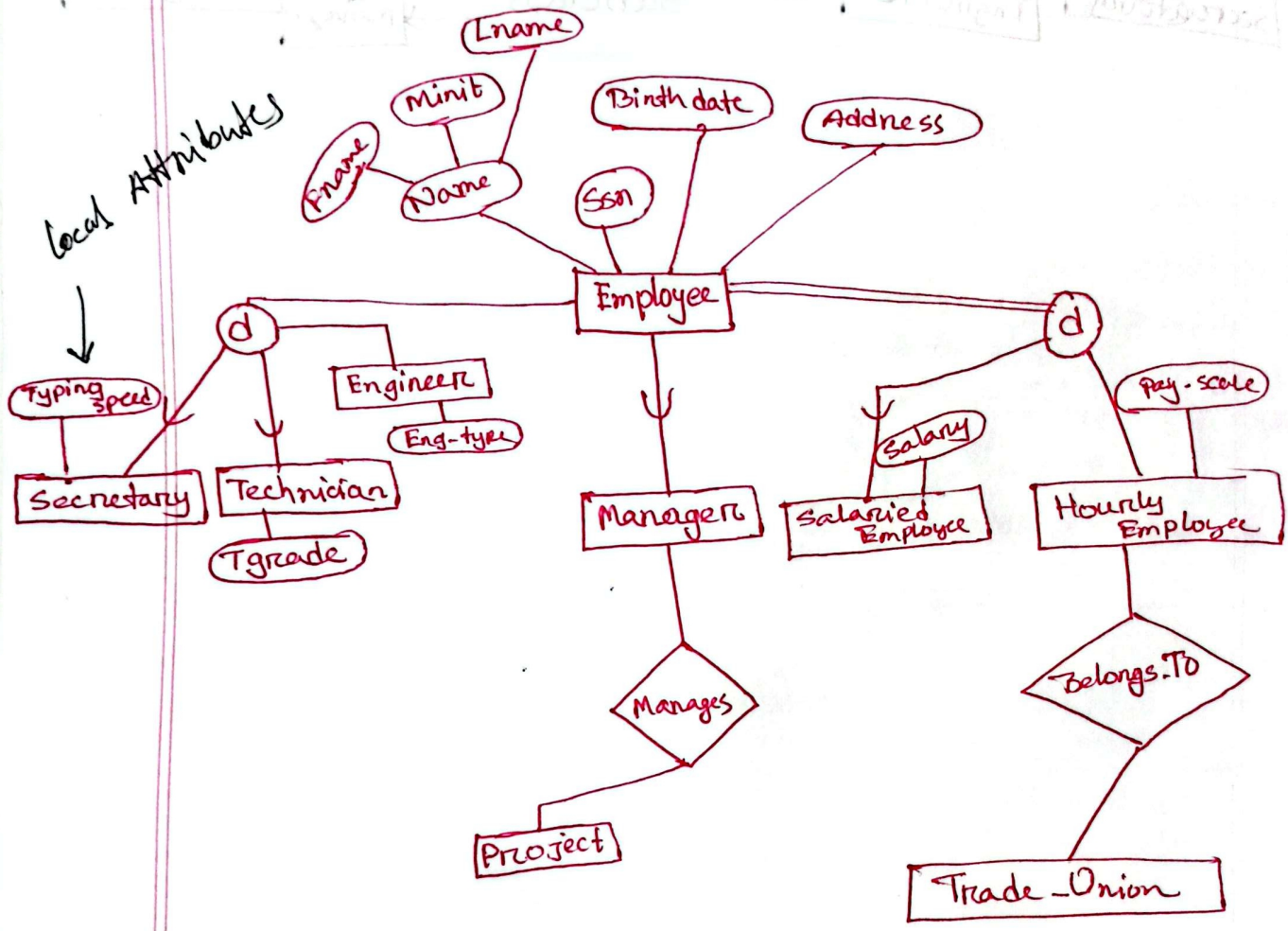
EER → Extended ER

(i) Subclasses / Superclasses

(ii) Specialization / Generalization

(iii) Attribute and Relationship Inheritance

(iv) Categories (Union Types)



Here, Employee is the Superclass

Manager, secretary etc are Subclass.

Relationship :-> ① Employee / Manager } Superclass/Subclass
Employee / Secretary

(ii) IS-A Relationship. Secretary IS-A Employee.
[Inherits]

(iii) Superclass এর entity থাকলে subclass এ থাকতে.
[But not mandatory]

(iv) ~~An entity in the superclass~~.

(iv) For any entity to exist in the db, superclass এর member হতে হবে

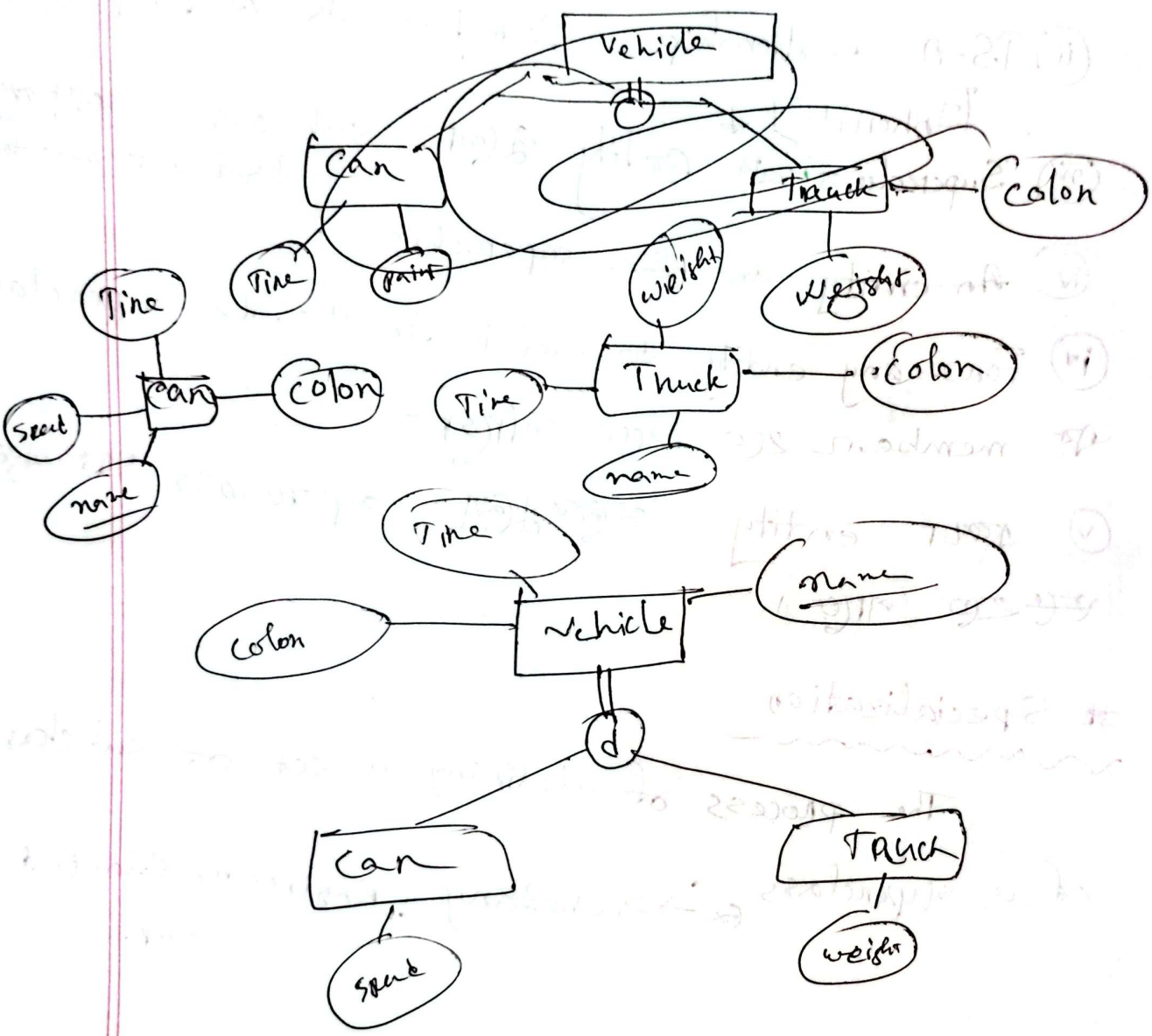
(v) একটা entity একাধিক superclass এর অন্তর্ভুক্ত হতে পারে।

* Specialization

The process of defining a set of subclasses of a superclass Ex -> Secretary, Engineer, Salaried emp...

Generalization

যখন ২টি superclass নিয়ে আলোচনা superclass এর under এ নিয়ে যাই, তখন তারা subclass হয়ে যায়।



Types of Specialization

Predicate-Defined

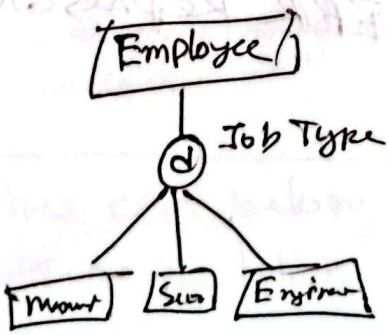
Membership is determined by a condition or rule. (Ex) → EMPLOYEE Belongs to ENGINEER if Jobtype = "Engineer".

Attribute defined

Membership is based on a specific attribute in the superclass. (Ex) → EMPLOYEE specialization based on JobType.

User-Defined

Membership is manually determined by the db user. (Ex) → PROJECT assigned to CRITICAL_PROJECT based on user decision.



#2 basic Constraints can apply to a specialization / generalization

(i) Disjointness Constraint

(ii) Completeness Constraint

(i) Disjointness Constraint

1. Disjoint Specialization (d)

- An entity can belong to only one subclass.

Ex → • A student can Either be GRAD or Undergrad

- EER Representation → (d)

2. Overlapping Specialization (o)

- An entity can belong to multiple subclasses.

Ex → • An Employee can be both Manager and Tech.

- EER Representation → (o)

(i)

(ii)

(iii)

(iv)

(ii) Completeness Constraint

1. Total Specialization (=)

- Every entity must belong to atleast one subclass.

Ex → • Vehicle must be a car or truck

- EER Representation : Double line (=)

2. Partial Specialization (—)

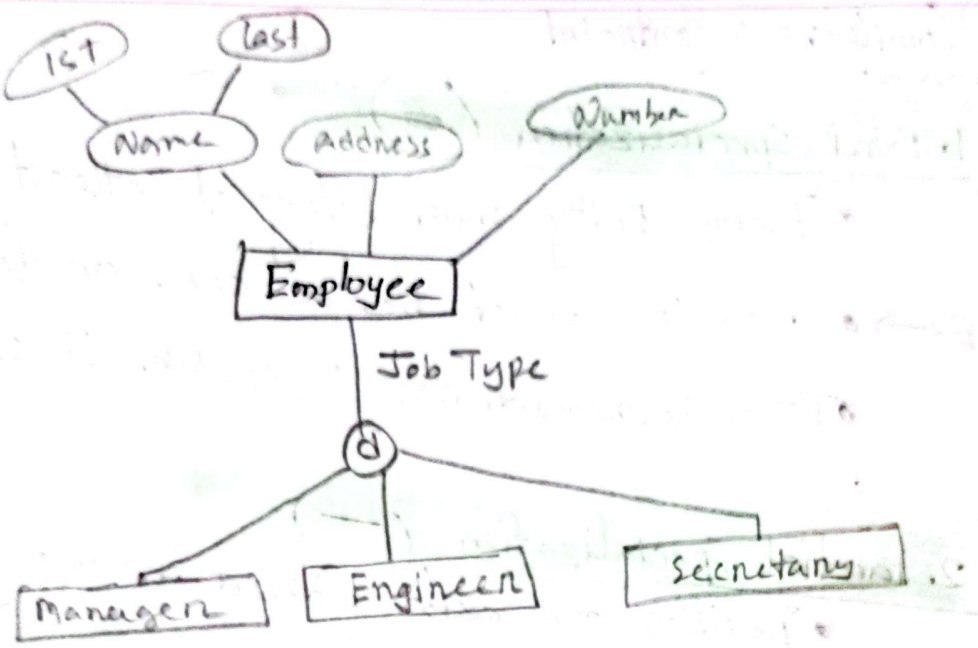
- Entities can exist without belonging to any subclass

Ex → • Some Employees may be neither Eng nor Tech.

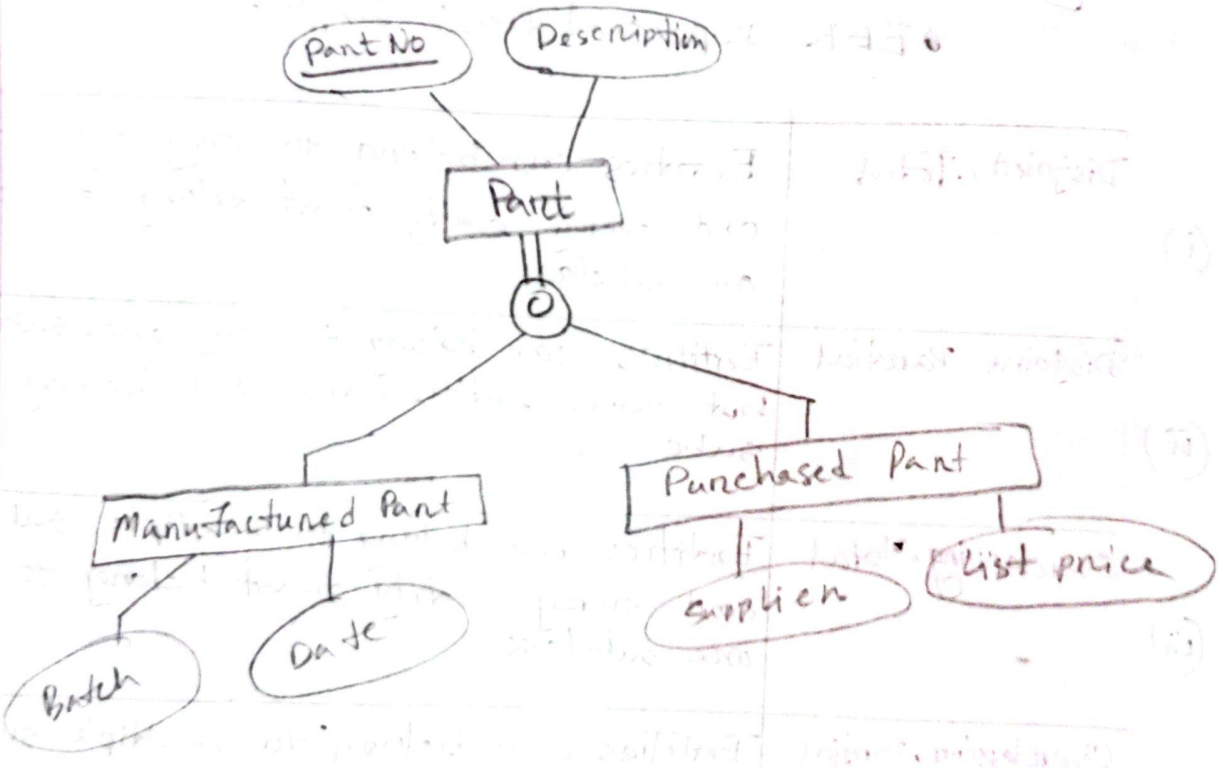
- EER Representation : Single line (—)

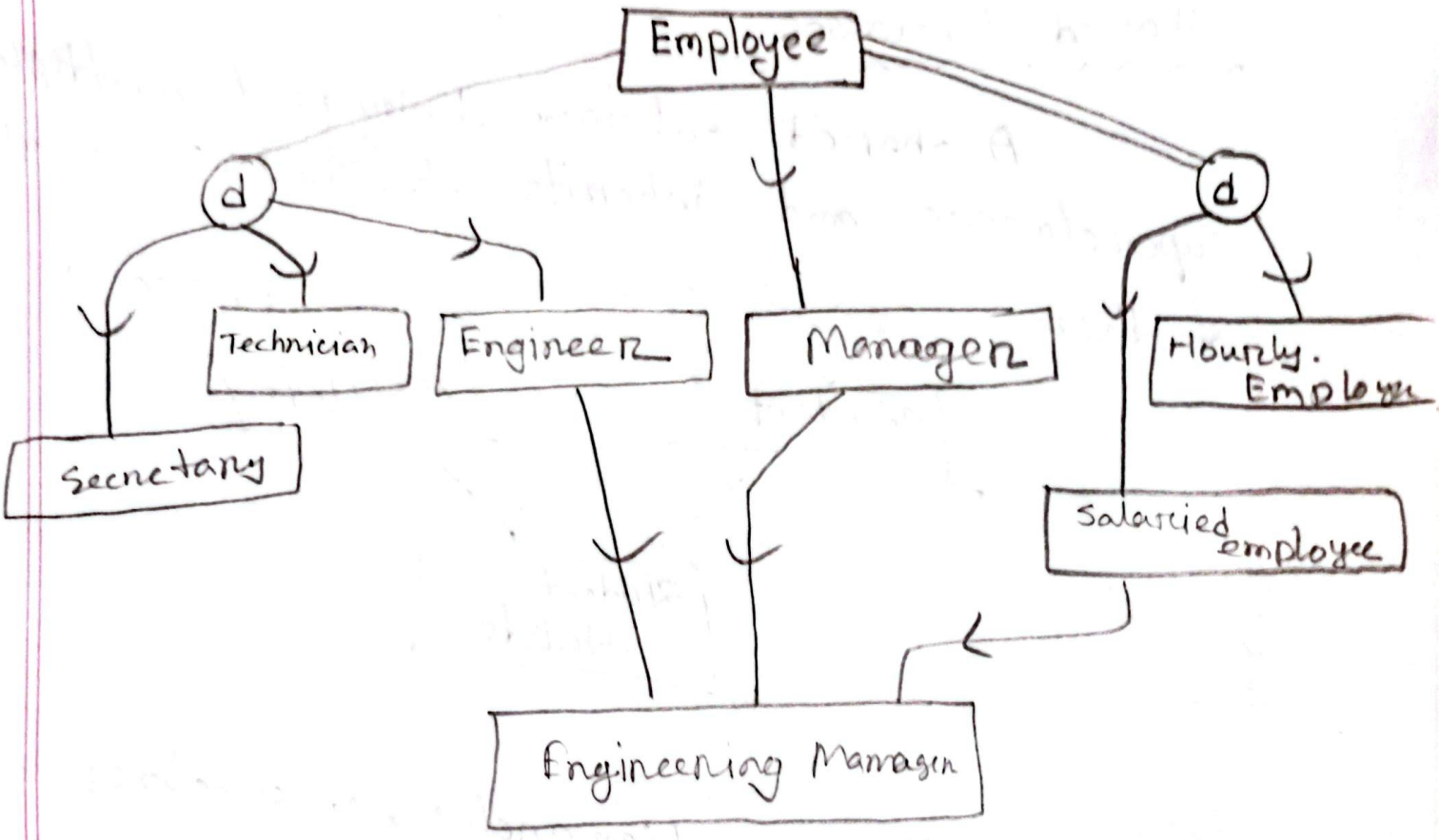
(i)	Disjoint, Total	Entities can belong to only one subclass, and every entity must belong to at least one subclass
(ii)	Disjoint, Partial	Entities can belong to only one subclass, but some entities may not belong to any subclass
(iii)	Overlapping, Total	Entities can belong to multiple subclasses, and every entity must belong to at least one subclass
(iv)	Overlapping, Partial	Entities can belong to multiple subclasses and some entities may not belong to any subclass.

(i)


1st
last
Name
Address
Number
Employee
Job Type
d
Managern
Engineern
Secretary

(ii)


Part NO
Description
Part
O
Manufactured Part
Purchased Part
Batch
Date
Supplier
List price



Hierarchies, Lattice & Shared Subclass

Hierarchy

Subclasses have only one superclass, forming a tree structure: (Single Inheritance)

EMPLOYEE → MANAGER → ENGINEERING MANAGER

Lattice

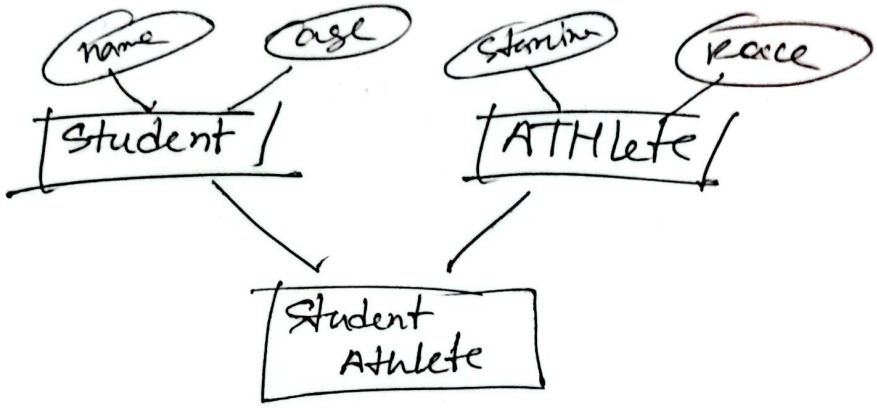
Subclasses can have multiple superclasses. A subclass can have more than one parent superclass,

Person Employee

Engineering Manager

Shared Subclasses

A shared subclass belongs to multiple superclasses and inherits attributes from all of them.

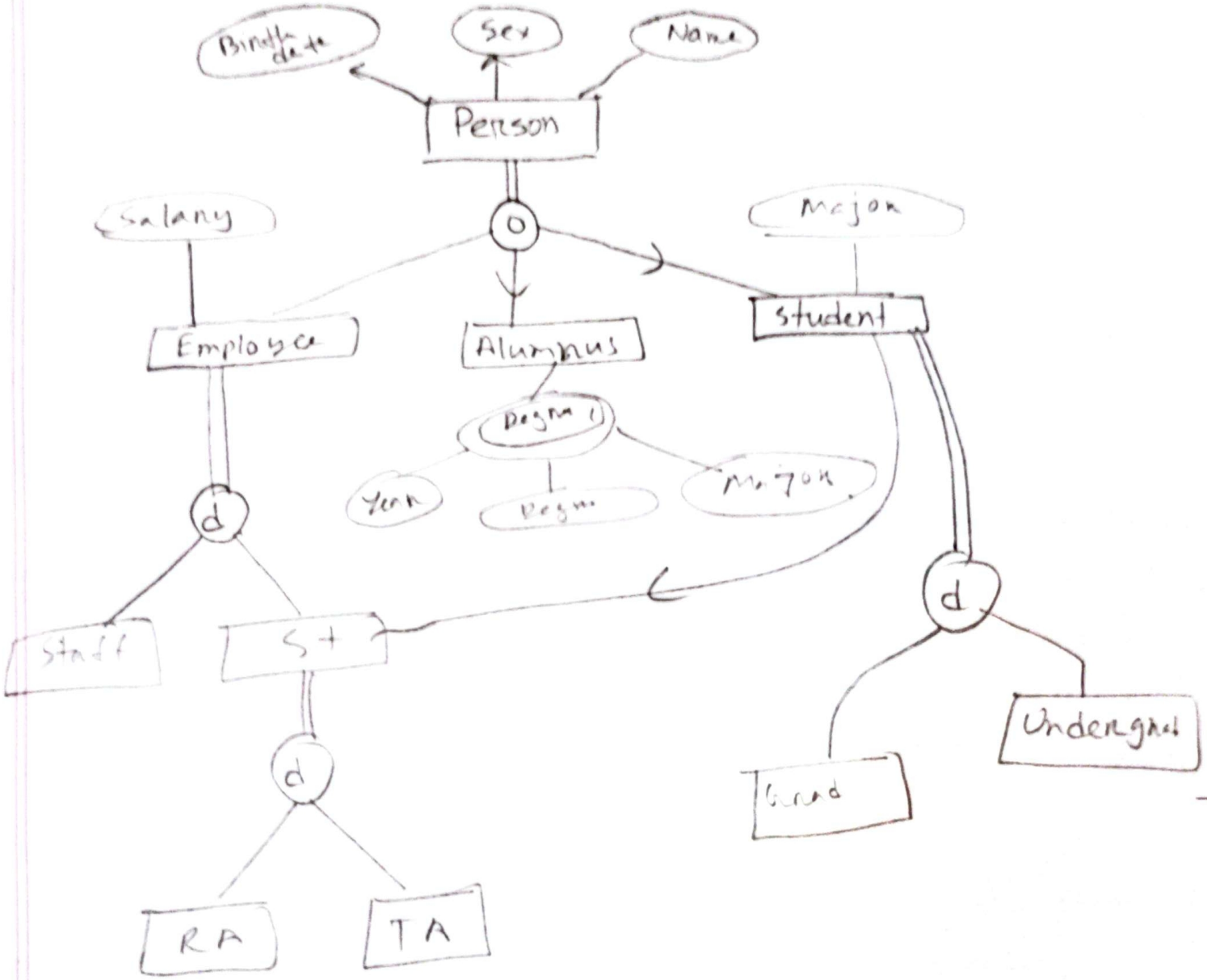


- In a lattice or hierarchy, a subclass inherits attributes not only of its direct superclass, but also of all its predecessor superclasses.

Specialization	Generalization
start with one entity and break it into smaller specific subclasses. This is a top-down conceptual refinement process	start with many entities and combine them into a common superclass. This is a bottom-up conceptual refinement process.

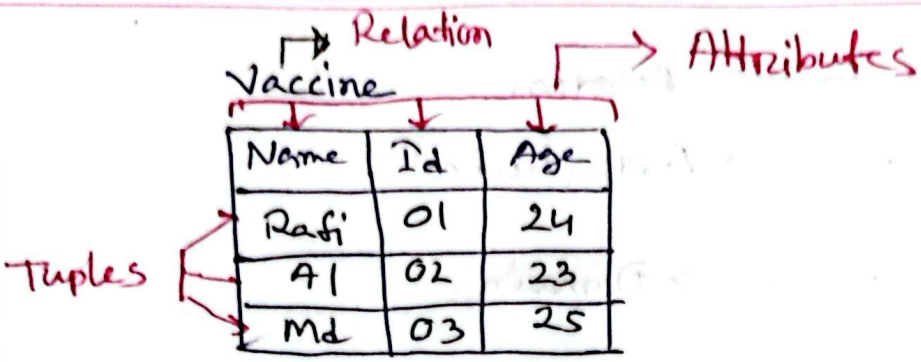
In practice, both processes are often used together.

Example -> University Database



Chapter-5

DMand RDC



Schema of a Relation ⟶ $R(A_1, A_2 \dots A_n)$

Vaccine(Name, id, age)

- Each att. has a domain or a set of valid strings (Number, Fix or Group [size]) (Fixed values)
- The domain may have a data-type and/or format.
- Tuples is an ordered set of values. (<...>)
- <"Rafi", 01, 24>
- The relation state is a subset of the Cartesian product of the domains of it's att.
- R(A1, A2) be a Relation schema.

$dom(A_1) = \{0, 1\}$ $dom(A_2) = \{a, b, c\}$

So, $dom(A_1) \times dom(A_2)$ is all possible combinations.

$\{\langle 0,a\rangle, \langle 0,b\rangle, \langle 0,c\rangle, \langle 1,a\rangle, \langle 1,b\rangle, \langle 1,c\rangle\}$

Table ⟶ Relation

Col. Header ⟶ Attribute

All possible col. values ⟶ Domain

Row ⟶ Tuple

Table Definition ⟶ Schema of a relation

Populated table ⟶ ~~State~~ State of a Relation

- Thee tuples (rows) in a table have no specific order
- The att. and their values are in a fixed order.

Characteristics of Relations

Atomic Values Each value in a tuple is indivisible

Valid Values Each value must come from the domain of its own att.

Tuple Format For a row $t = \langle v_1, v_2, \ldots, v_n \rangle$ each v_i corresponds to a specific column A_i

Null Values Special values used for unknown or inapplicable data.

#Notation

- t[Ai] or t.Ai Refers to the value of col. Ai in row t.
- t[Au, Av, ..., Aw] Refers to specific values of selected cols in row t.

#Relational Integrity Constraints

↳ 3 main types in the relational model

(i) Key Constraint

Each row in a table must be unique, identified by a key. (like primary key)

(ii) Entity Integrity Constraints:

The primary key of a table can't be null, ensuring each record is uniquely identifiable.

(iii) Referential Integrity Constraints

(FK=PK) is (FK not in PK null is also allowed) Foreign keys must match a valid primary key in another table or be null, ensuring relationships between tables remain consistent.

Domain Constraint

সেই col এ আছে ওই datatype মানতে হবে।

Key Constraints

Superkey : set of keys (→ Attributes) that are unique.
প্রত্যেক row কে আলাদা val

Key : K ⊆ SK ; minimal superkey.

যদি কোন superkey এর set থেকে কোনো এক ~~key~~ att সরানোর পর সে আর super key না থাকে তবে সেটা key.

(Ex) { course id, name } ; এখানে course id সরালে সে আর unique না superkey থাকে না, তাই এটা key।

Key Type	Purpose	Null allow.	Example
Primary key	Uniquely identifies each row in the table.	NO	Student_ID in Student table
Foreign key	References the primary key in another table. [Not unique]	Yes. if no match	Student_ID in Course table
Candidate key (candidate for PK)	Potential key that could uniquely ~~index~~ identify. [unique] rows.	Depends on use	{serial} {state, reg}

Every primary key is a candidate key.

but not vice versa.