Unit-1

Database Management Systems

A database is a collection of related data.

Data :→ that have some implicit meaning.

So, a database is a logically coherent collection of data, with some inherent meaning.
A random assortment of data cannot correctly be referred as a database.

A database management system (DBMS) is a computerized system that enables users to create & maintain a database.

Defining a database involves specifying the data types, structures and constraints of the data to be stored in the database.

The database definition or descriptive information is also stored by the DBMS called as Meta-data.

How to store Data in a table?

We must specify a data type for each data element within a record.

A data record is a collection of related data items or a row in a table.

A data element specifies the type of data a column contains.

Collection of record is a file.

Ex. Let say a file name is Cricket

Cricket

data element

Player's Name	Type	Team	Avg.
M.S. DHONI	Batsmen	India	38.09
A.B.D.	Batsmen	S.A.	53.81

Record

In that file we have data element like name of the player (Player's Name), Type of player either Batsmen, Bowler or All rounder (Type), Team in which he/she plays (Team), Batting/Bowling Average (Avg.)

Characteristics of Data Base

older approach is File System (still used in some place)

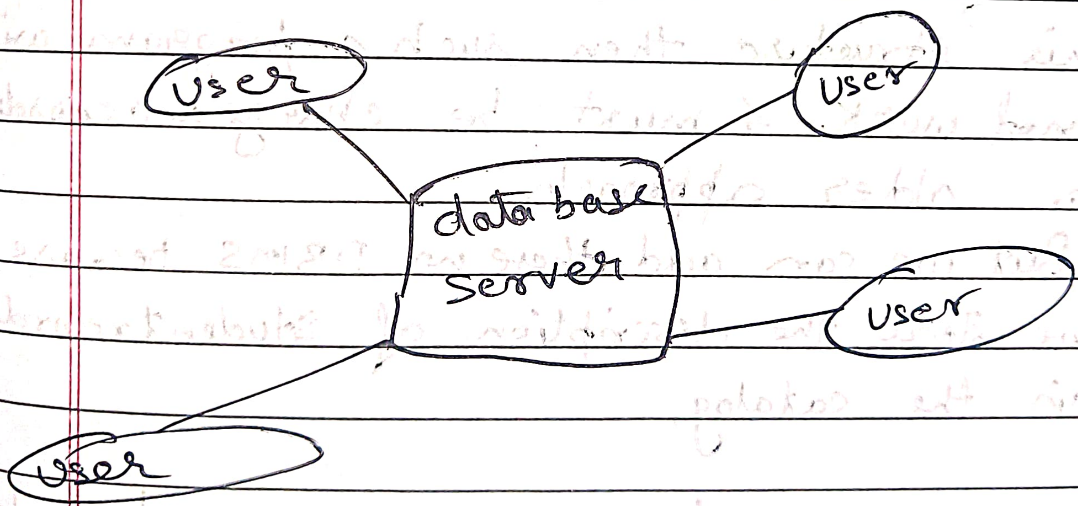
In this approach, data is stored in files, to access that data we have to write customize programs

for ex → Let say, we have SRE storing data of your marks (i.e students 4 marks) & the finance dept. is storing the data of students and their fees and their payments.

Although both are storing the data about students, each maintains ~~a~~ separate files.

This redundancy in defining and storing data results in wasted storage & more efforts to maintain common up-to-date data.

In database approach, a single repository maintains data that is defined once & then accessed by various users repeatedly through queries, transactions and application programs.



Characteristics

① → Self-Describing Nature of a Database System
↳ contains data, metadata, structure of database

like

Column-Name	Data Type	Belongs to relation
Name	character(30)	Student

2 Insulation between programs and data and data abstraction.

In traditional approach, any change to the structure of a file may require changing all the programs that access that file.
But this is not the case in most of the cases in DBMS

ex
Let say their is a table STUDENT

Student

Name	Id	specialization

If we want to add Date_Of_Birth in this structure then such a program will not work & must be changed according to older approach.
But we can add these in DBMS because we store the description of student records in the catalog.

Data Item Name	Starting Position in Record	Length of Character
Name	1	30
Id	31	4
Specialization	36	30

Just add a new record here of Date_of_Birth
no need to change the programs.

The structure of data files is stored in the DBMS catalog separately from the access programs this property is called as program-data independence.

Other property is program-operation independence → In this property, the invoking of operation (create, delete, update), their name & arguments, regardless of how the operations are implemented is operated by the user application programs.

The characterstic that allows both these properties is called data abstraction.

meaning → A DBMS provides user with a conceptual representation of data that does not include many of the details of how the data is stored or how the operations are implemented

A Data model is a type of data abstraction that is used to provide this conceptual repres.

classmate
Date
Page

(3) Support of Multiple Views of the Data

A database has many users, each may require a special perspective or view of the database.

A View could be a subset of database or it may contain Virtual data from the database but not stored anywhere.

→ A multi User DBMS must have multiple views

ex

A user may be interested in printing the transcript of a student.
Other user may be interested in accessing the course.

(4) Sharing of Data & MultiUser Transaction Processing.

A DBMS must allow multiple users to access the database at the same time.
So, DBMS must include/maintain Concurrency to ensure that several users trying to update the same data, the results of the updates is correct.

A fundamental role of multiuser DBMS Software is to ensure that concurrent transactions operate correctly & efficiently

The DBMS must enforce several transaction properties

ACID

- Atomicity
- Consistency
- Isolation
- Durability

Users of Data base

1. Data base Administrators
2. Database Designers
3. End Users
   - Casual end users
   - Naive or parametric end users
   - Sophisticated end users
   - Standalone users
4. System Analysts & Appⁿ programmers

Advantages of using DBMS Approach

1. Controlling Redundancy
2. Restricting Unauthorized Access
3. Providing Persistent Storage for Program Objects

(4) Providing Storage Structures & Search Techniques for Efficient Query Processing

(5) Backup & Recovery

(6) Multiple User Interfaces

(7) Representing Complex Relationship among Data

(8) Enforcing Integrity Constraints

↳ A DBMS should provide capabilities for defining & enforcing these constraints

Some Constraints

(1) Specifying a data type for each data item

(2) Specifies uniqueness on data item values
ex every course record must have a unique value for Course-number

(3) Referential integrity Constraints
↳ when a record in one file is related to another file there are some constraints that are needed to be followed

Database Systems

A data model- is a collection of concepts that can be used to describe the structure of a database - provides the necessary means to achieve data abstraction

Structure means→ data types, relationships & constraints that apply to the data

Types of Data Models

Data Model

- Conceptual Data Model
- Representational Data Model
- Physical Data Model

Conceptual Data Model :→ It describes the database at a very high level & is useful to understand the needs or requirements of the database.

It provides concepts that are close to the way many users perceive data

Physical Data Model :→ provide concepts that describe the details of how data is stored on the computer storage media.

Between these two model there is a class of representation data models which provide concepts that may be easily understood by end users but that are not too far removed from the way data is organized in a computer storage.

Conceptual Data models use concepts like entities, attributes & relationships

An entity represents a real-world object or concept, such as a employee or a project

An attribute represents some property of interest that further describes an entity.
ex employee's name, salary.

A relationship among two or more entities represents an association among the entities
ex

a works on relationship b/w an employee and a project.

Schema

The description of a database is called the data base schema.

ex Student

Name	Id	class	Major

} – Schema.

Instances

The data in the database at a particular moment in time is called a database state or database snapshot.

Three-Schema Architecture.

Integrity Constraints

These are rules defined in a database to maintain data accuracy, consistency & reliability

Integrity Constraints

Domain Constraints	Entity Integrity Constraints	Referential I C	Key Constraints
↓ Attribute Like Age can't be negative, mobile no. must be of 10 digits We also define the type of data Check (Age > 0) Check (length (Mobile No) = 10)	Table have one primary key which is NOT NULL	foreign key	↓ Table should have a unique attribute (Candidate key)

Candidate Key (Key is an attribute which uniquely identify the table)

The set of ~~of~~ all the keys in a table which can uniquely identify the table is called candidate key set.

ex For a student key can be
1. Registration Number
2. Adhar Card 3. Voter ID 4. Phone No.

We can pick any appropriate Key from candidate key set and make it a Primary Key. & all other are called as alternative Key

Primary Key
A key which uniquely identity & have no Null value.
Unique
(Phone No, Adhar Card, PAN, Reg. No.) → Candidate keys
PK (Unique, Not Null)

So here PK can be (Reg. No) ~~A database have only 1 PK.~~

Foreign key → It is an attribute or set of attributes that references to primary key of same table or another table.
→ Maintains Referential Integrity

Student

PK

Referenced table

Roll no.	name	Address
1	A	Delhi
2	B	Agra
3	A	Mum.

Course

PK FK

Course-id	Course-name	Roll_no
C_1	DBMS	1
C_2	CN	2

Refering table

Here

Unit-2

Entity Relationship Diagram

E-R diagram can express the overall logical structure of a database graphically

It's major components

* Rectangles → Entity
* Ellipse → Attributes
* Diamonds → Relationship
* Lines → links b/w att & Entity
* Double Ellipse → Multivalued attributes
* Dashed Ellipse → Derived attributes
* Double Lines → Total participation
* Double Rectangles → Weak Entity set.

Types of Attributes

1. Single vs Multivalued Attributes
2. Simple v/s Composite Attributes
3. Stored v/s Derived Attributes.
4. Key v/s Non-Key Attributes
5. Required v/s Optional Attributes
(mandatory field) (not necessarily field)

have only one value

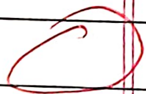
① Single v/s Multivalued → more than one values

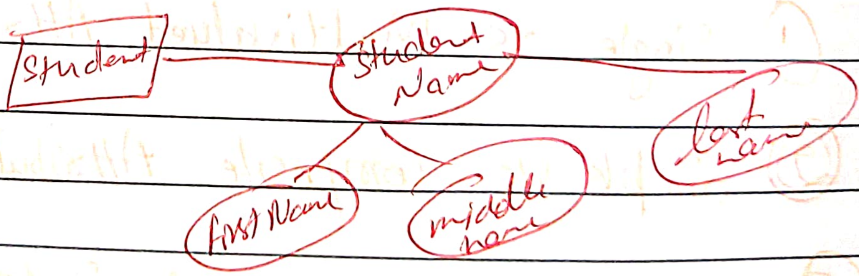
Single	Multivalued
Registration NO, Age, Salary	Phone No

Single Valued → 

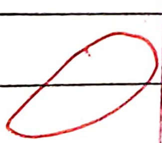
Multi valued → 

② Simple v/s Composite Attributes → can be divided (made up of more than one value)

Simple	Composite
Can't be broken further like age.	Name → firstname, middlename, last Name
	Address



③ Stored v/s Derived Att → can be derived from other attribute

Stored	Derived
DOB	Age (derived from dob)
	 dotted/dashed ellipse

If a ... has some attributes these are called descriptive attributes.

4) Key vs Non-key Att.

Key	Non-key
Unique Value	not necessary unique value
underlined ellipse	ellipse

5) Complex Attribute
(composite + Multivalued)

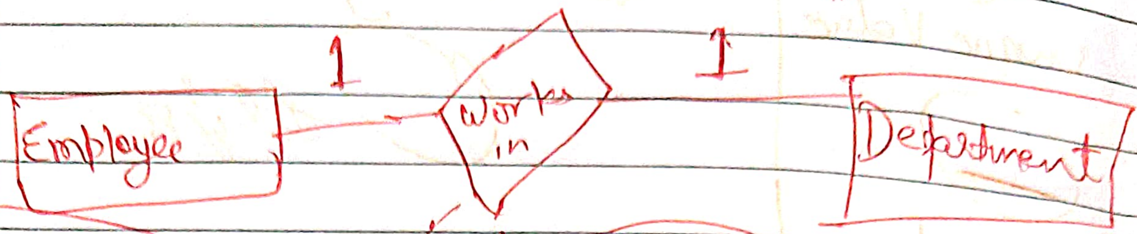
ex → (If someone has two residential address)

~~Types of~~ Degree of Relationships (Cardinality)

1) One - One Relationships (1-1)
2) One - Many " (1-M)
3) Many - One " (M-1)
4) Many - Many " (M-N)

1. One-One

Relationship also have a table in which the key of both entities is necessary



Eid	Ename	Age
E1	V	25
E2	W	21
E3	W	21
E4	X	29
E5	Z	31

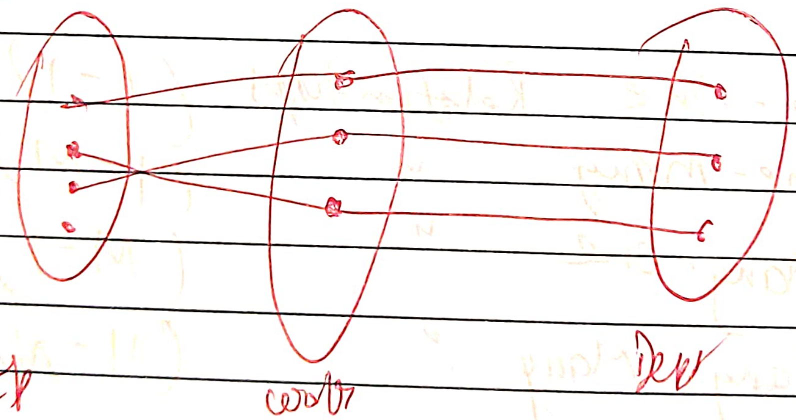
Works in

FK Eid	Did FK
E1	D1
E3	D2
E2	D3

Did	Dname	Loc
D1	IT	Bang
D2	HR	Delhi
D3	Production	Delhi

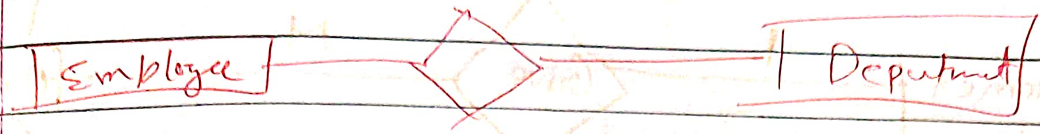
Primary Key in this table can be Either Eid or Did.

let say Eid is the key of works in table



We can merge the relation and entity table because to save space. We can merge it with any of entity (~~let say~~ here Eid or Did both can become key let is say Eid is the key in works in table) with key in relationship table

So the works in table will merge with employee table



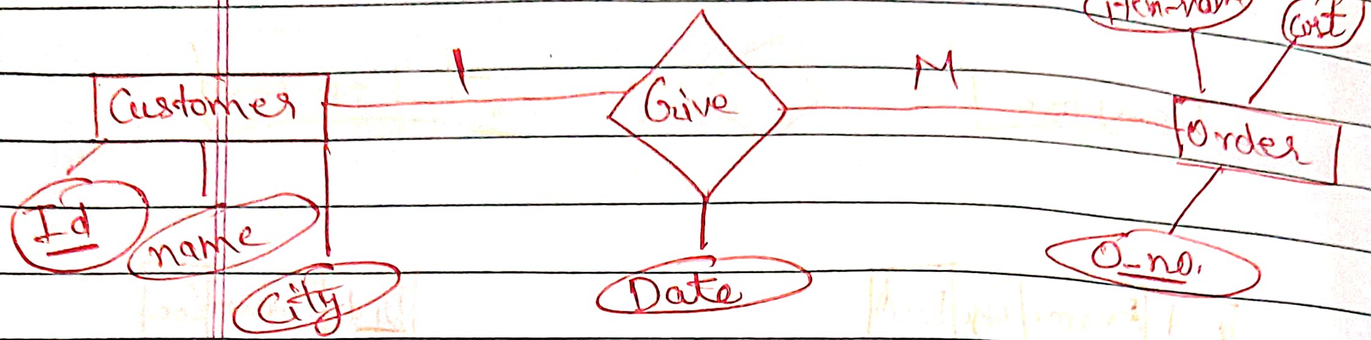
Eid	Ename	Age	Did
E1	V	25	D1
E2	W	21	D3
E3	W	21	D2
E4	X	29	Null
E5	Z	31	Null

Did	Dname	Loc
D1	IT	Banglore
D2	HR	Delhi
D3	Prod	Delhi

Total Number of tables ⇒ 2

(2) One - Many Relationship (1-M)

A customer Give many Order.



FK - FK

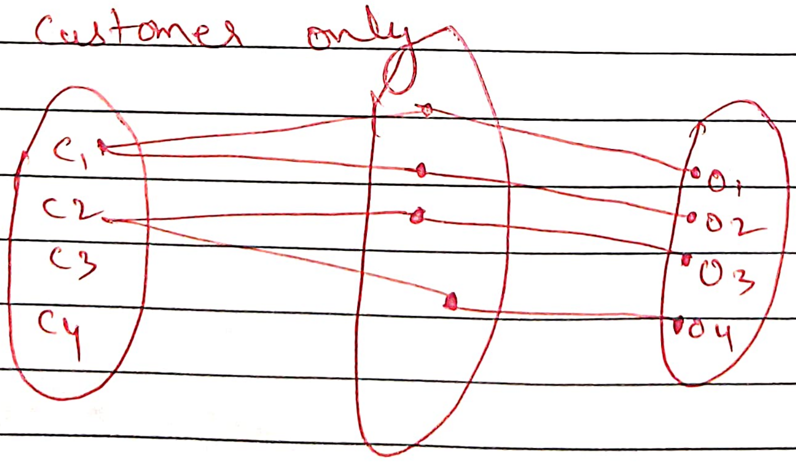
Id	Name	City
C_1	A	Jal
C_2	B	Delhi
C_3	C	Mum.
C_4	A	Mum

Id	O-no	Date
C_1	O_1	5 Aug
C_1	O_2	5 Aug
C_2	O_3	6 Aug
C_2	O_4	7 Aug

O-no	Cost	Item-name
O_1	100	Bucket
O_2	2K	Shoes
O_3	1.5K	Shirt
O_4	2K	Jeans

data repeat   No repeat   So Primary key (O-No)

A customer can give many order but an order will be mapped with one customer only



o-no is the PK so we can merge Give table with order table (bcz PK is in order table)

Id	Name	City

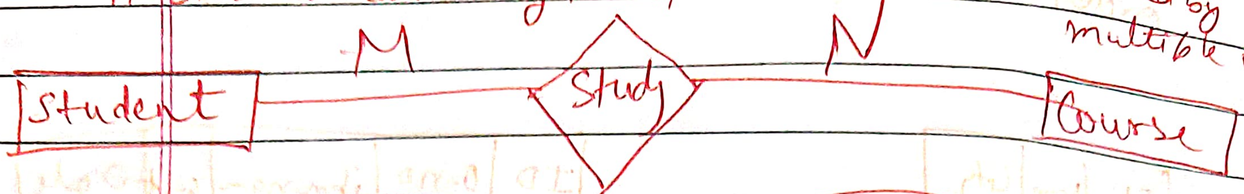
ID	O-no	item-name	cost	Date
C_1	O_1			
C_1	O_2			
C_2	O_3			
C_2	O_4			

The table will be merged on Many side

Total table → 2

3 Many-Many (M-N)

A student can study Multiple course & A course can be studied by multiple user



pk | fk | fk | pk

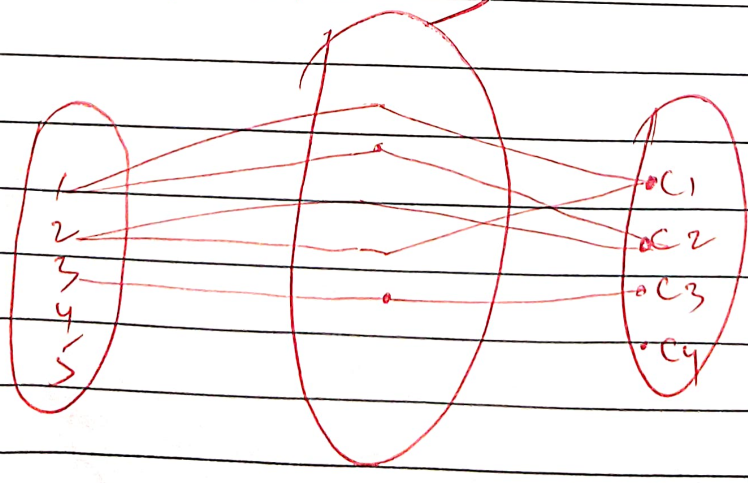
Roll_No	Name	Age
1	A	16
2	B	17
3	A	18
4	C	19
5	D	20

Roll-No	Cid
1	C1
2	C2
1	C2
2	C1
3	C3

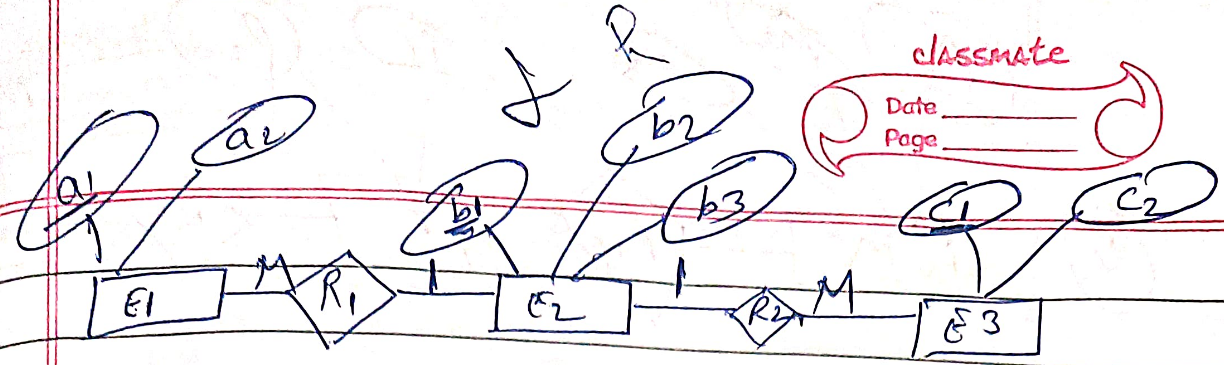
Cid	name	Credit
C1	Maths	4
C2	Phy	4
C3	Chem.	4
C4	Hindi	4

Primary key in Relationship table is the combined of both keys

Can't Reduce tables here
Total table - 3

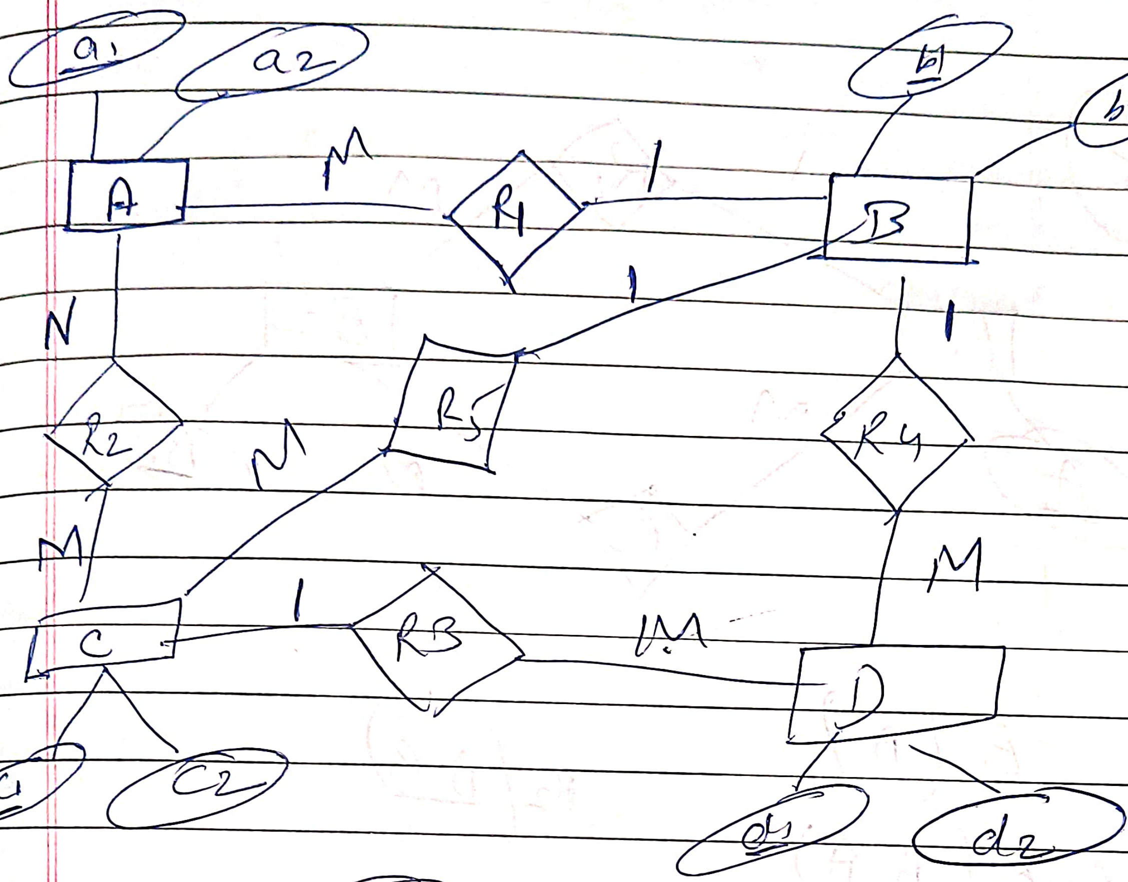


Q1



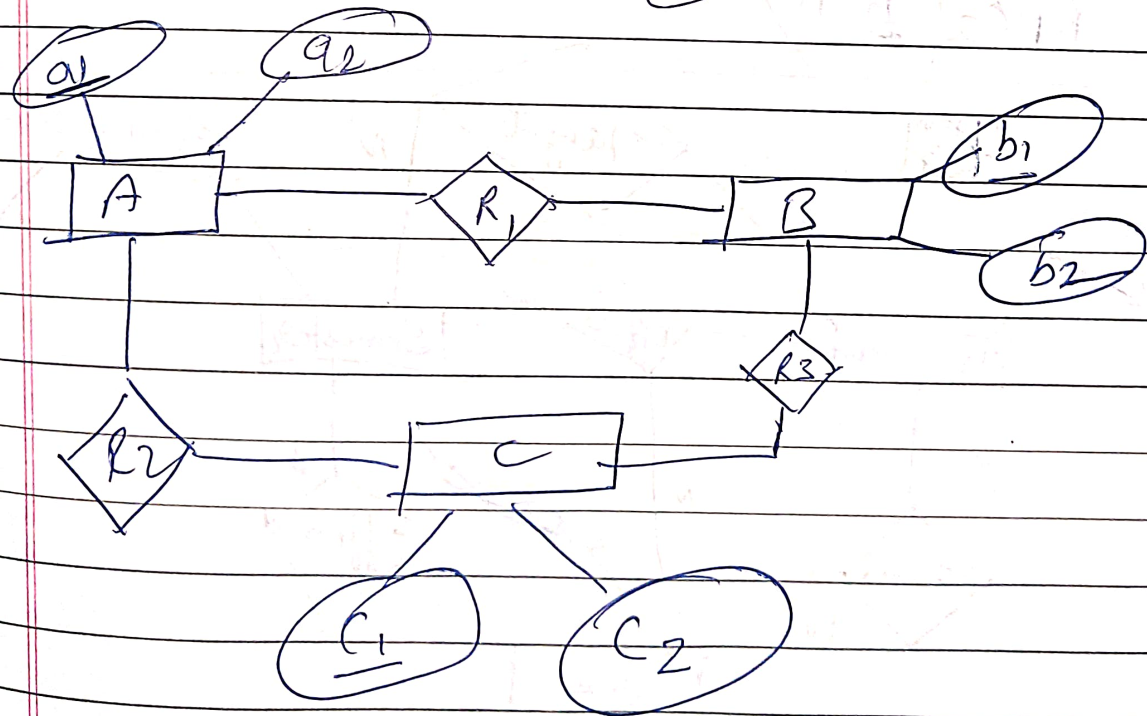
a1
a2
E1
M
R1
1
b1
b2
b3
E2
1
R2
M
c1
c2
E3

2



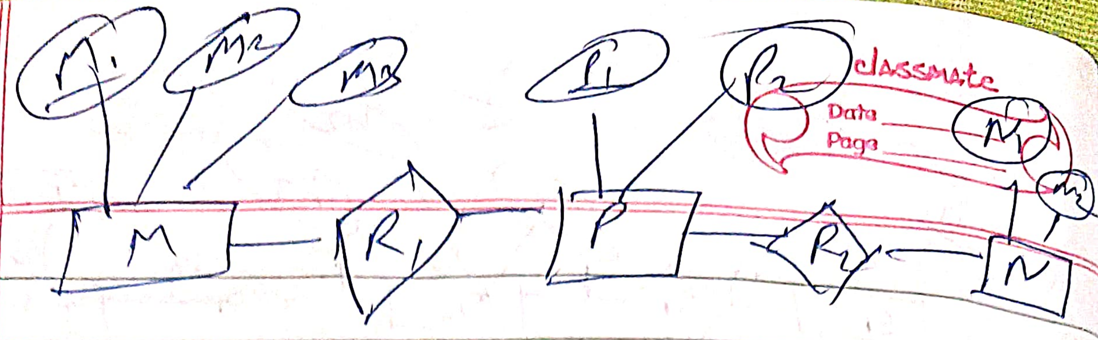
a1
a2
A
M
R1
1
b1
b2
B
N
1
R5
R2
M
1
R4
M
M
C
1
R3
M
D
c1
c2
d1
d2

3

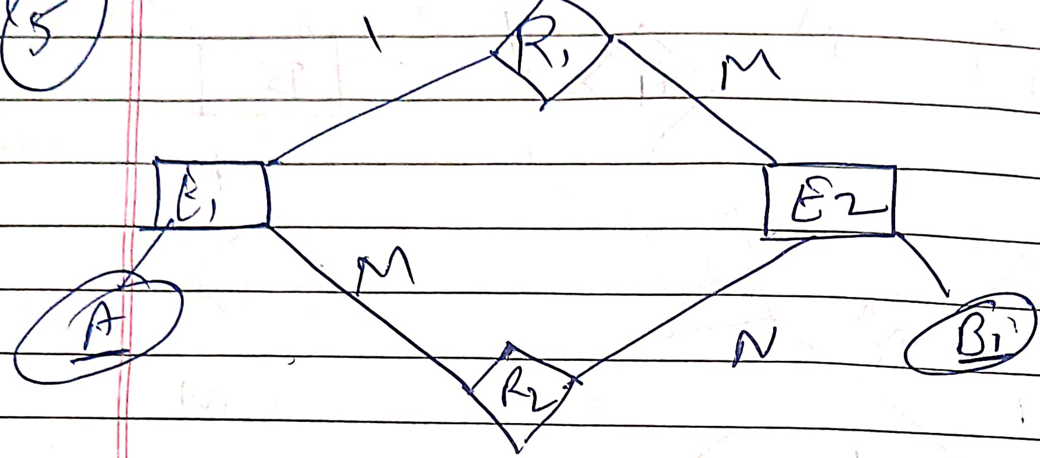


a1
a2
A
R1
B
b1
b2
R3
R2
C
c1
c2

4



5

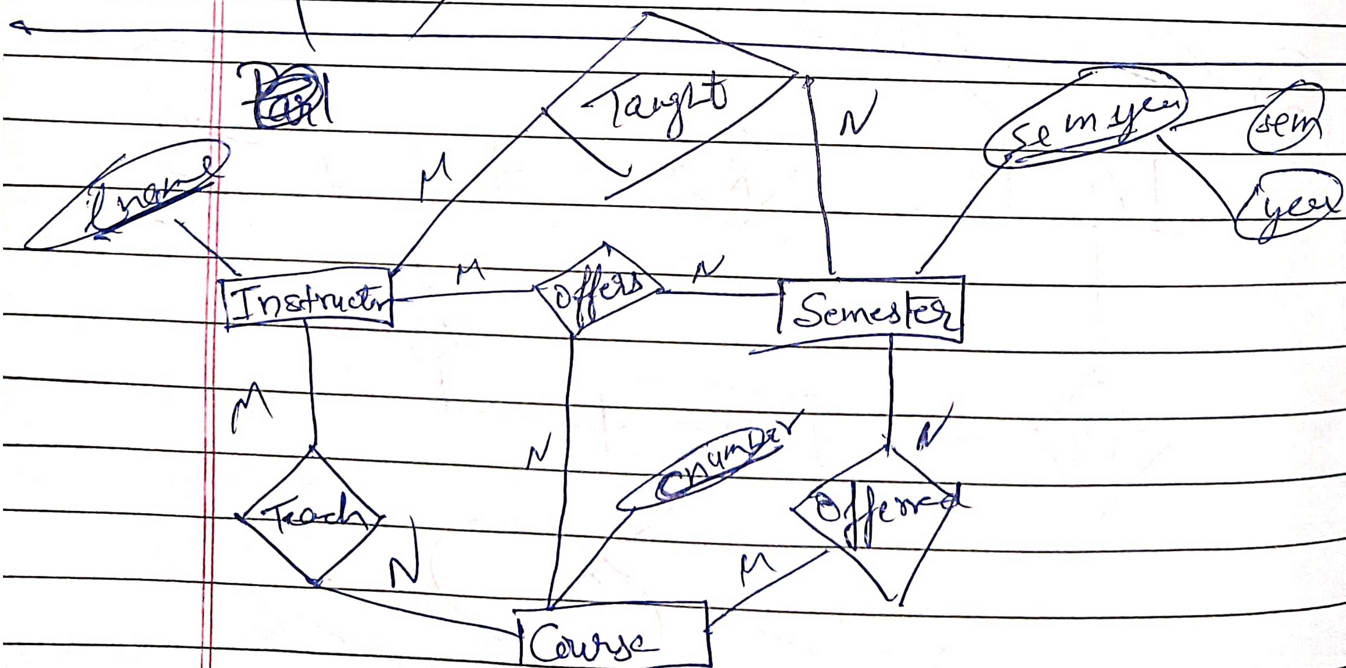


$E_1(A)$

$R_2(\underline{A\ B})$

$R_1 E_2(\underline{B}\ A)$

~~Pearl~~



Cardinality ratio: → max. no. of relationship an entity can participate

Participation Constraints (min relationship an entity can participate)

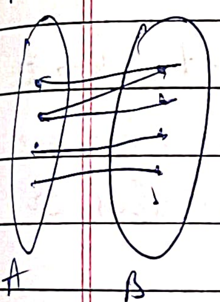
Total

↳ Requires every entity in one set to take part in a relationship in another set.

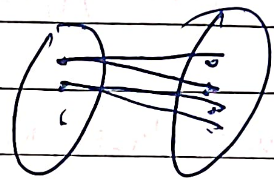
Partial

↳ Not necessarily every entity in one set to take part in relationship with another set.

eg each student is required to be registered in a minimum of one course.

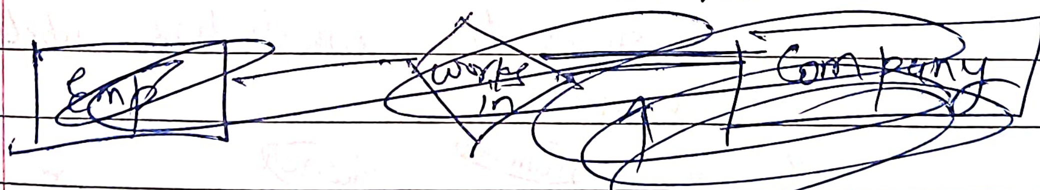


A to B Total participation



Partial

participation of weak entity in an identifying relationship is alway total



Total participation

Participation Total

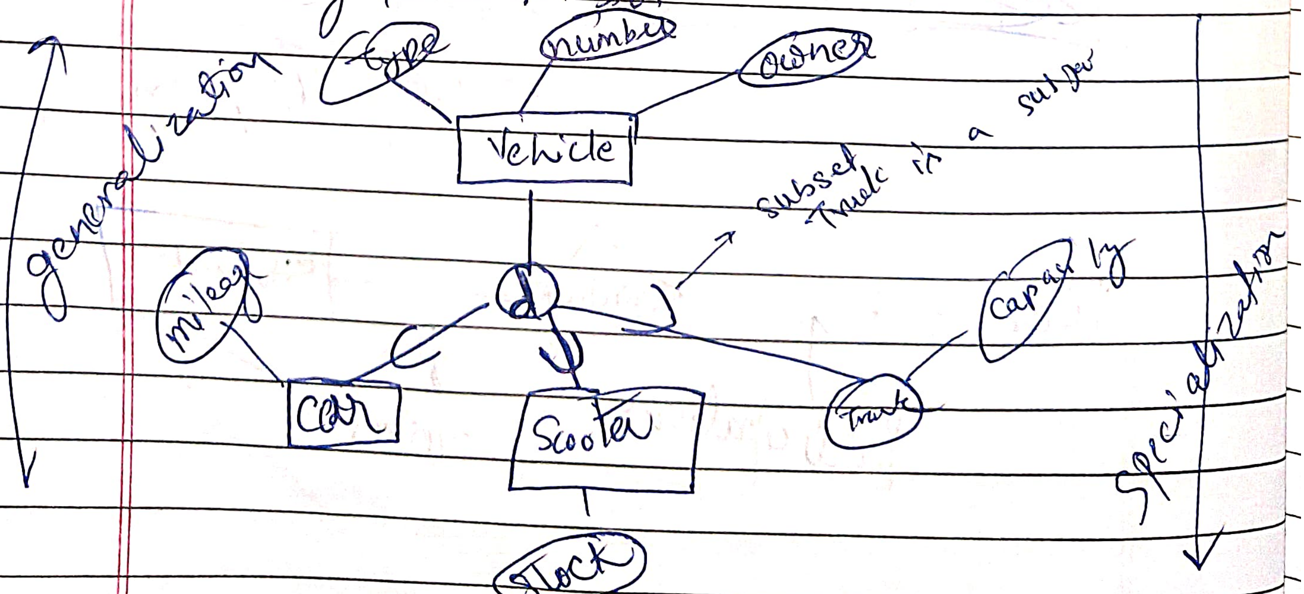
Enhanced-ER Diagram

EER model includes all modeling concepts of the ER model in addition it has foll. concepts.

1. Subclasses & Superclass
2. Generalization & Specialization
3. Category Or Union type
4. Attribute & Relationship Inheritance

(1) Superclass is a higher-level entity set that has common attributes.
Subclass is a lower-level entity set that inherits attributes & relationships from its superclass but also has its own specific attributes or relationships.

ex Science is a superclass which has subclasses like Phy, Chem, Bio.

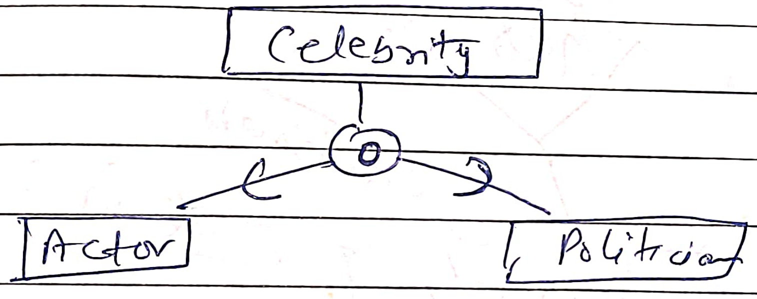


d → disjoint / disjunction

Date
Page

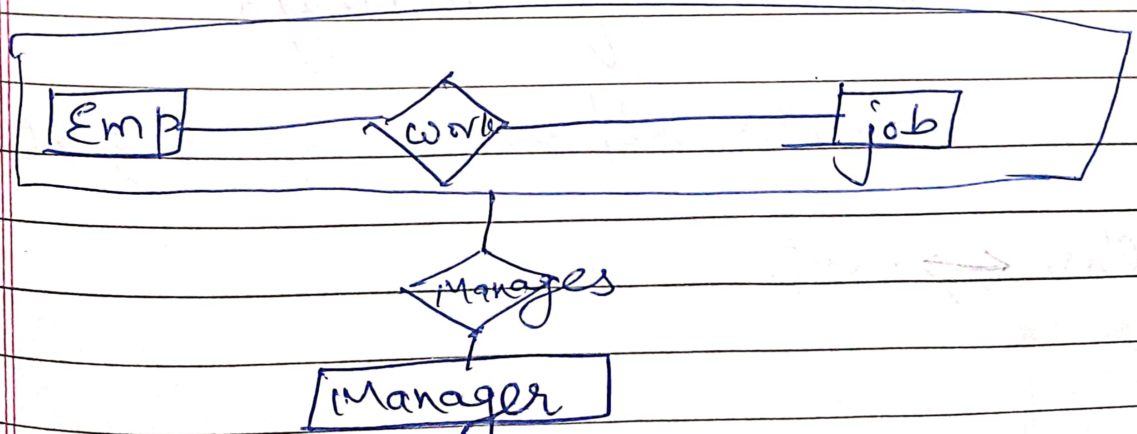
If we are designing the ER diagram using Top-down approach meaning firstly we make the superclass entity and then make the ~~supercl~~ subclasses is called specialization & vice-versa is called generalization (Bottom to top.

d → disjoint
O → whole

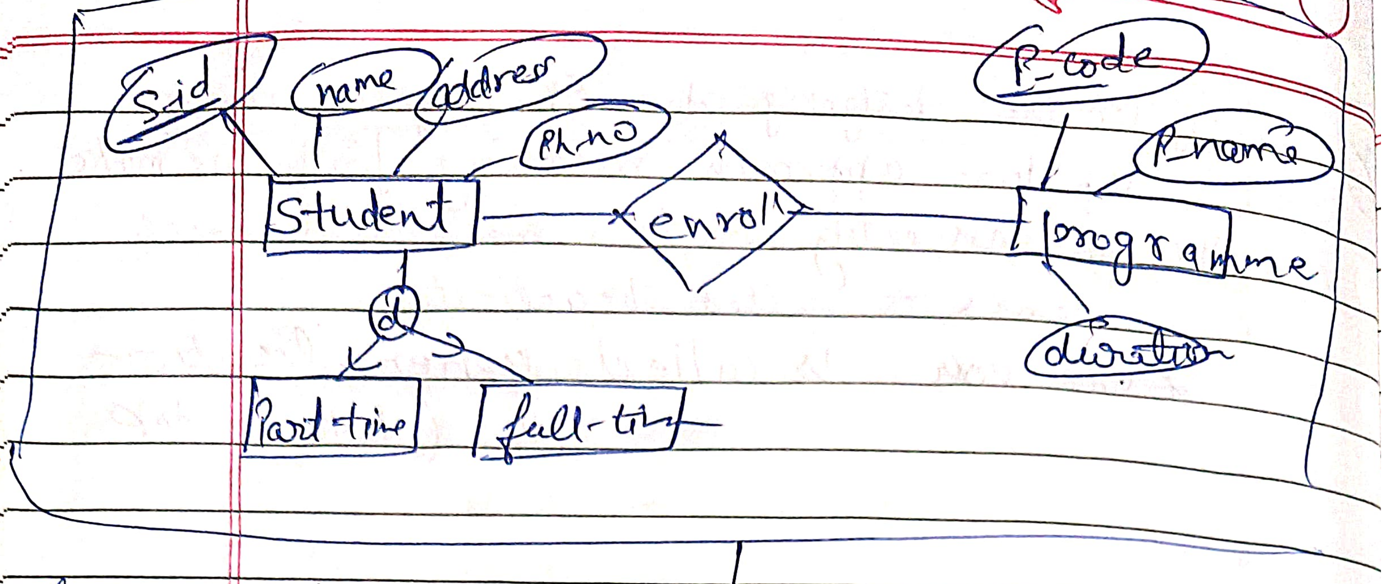


an actor can be a politician & a politician can be an actor

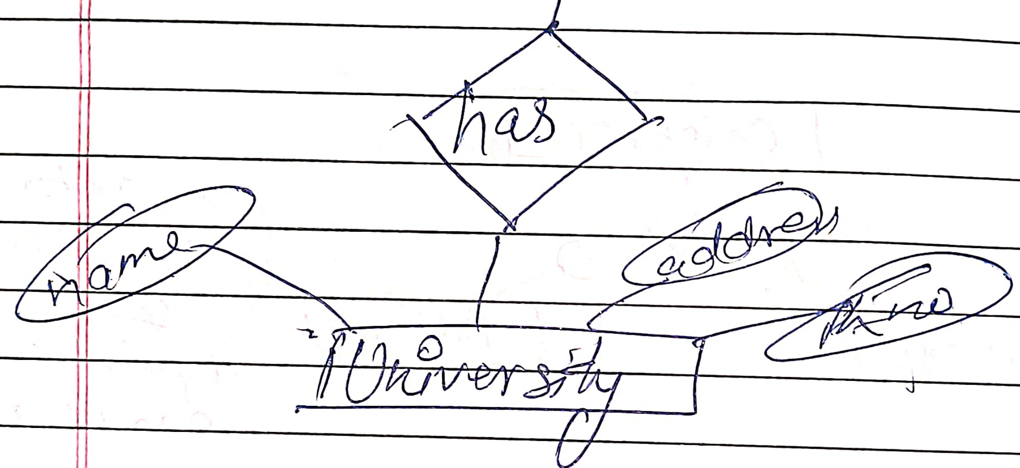
Aggregation



Manager will manage emp & also job so we aggregate both



Part-time & full-time entities have all the attributes of student & can have their own attributes also



UML Class Diagram

Unified Modeling Language

Class → Blueprint / template of an Object.

Class ⟷ entity

Class Name
Attributes
Operations/ functions

Public accessifier

- + → Public → other classes can also access
- − → Private → other classes can't access
- # → Protected → can only be accessed within the same class & the classes which are inherited by this class.

Student
- name : string - address : string - DOB : string - Phno : number
+ Register() : void + Login() : void + fill_details() : void

return type

Generally attributes are private & functions/methods are public

Can also add parameters

~~Relations~~

ex + fill_details(in fathers name : string) : void

parameter can be →
- input (In)
- output (Out)
- or both (InOut)

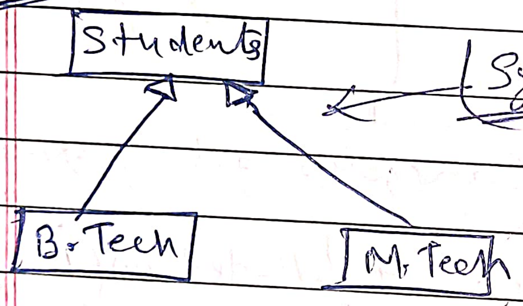
Input → An input parameter is a parameter passed from calling object to the called object during a method invocation

Relationships b/w classes -

↳ It means how classes are connected.

1. Composition
2. Association
3. Directed Association (⟶)
4. Aggregation
5. Generalization (Inheritance)
6. Realization (Interface Implementation)
7. Dependency Relationship (- - ->)
8. Usage Relationship (— use →)

Generalization



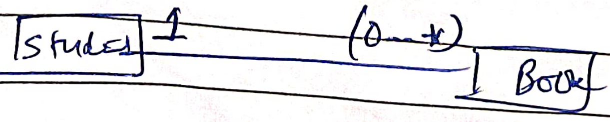
(Symbol ——▷)

B.Tech & M.Tech are also students

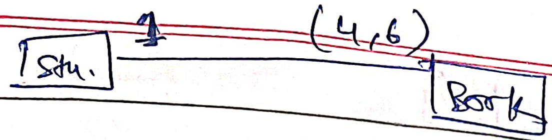
Association

(———) Symbol

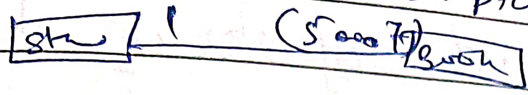
1	exactly one
(0...*)	zero or more
(1...*)	one or more
(0...1)	zero or one
(5...7)	5 to 7
(4, 6)	4 or 6



means one student can pick zero or more books



a student can pick 4 or 6 book



a student can pick 5,6,7 books

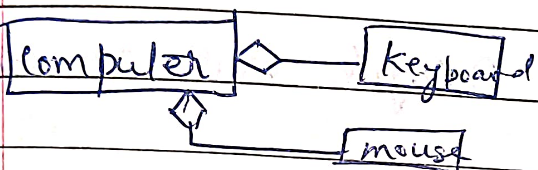
Association

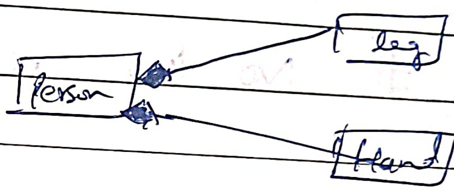
Aggregation → (part of)

symbol

Composition

symbol





~~Keyboard & mouse are part of computer.~~

Here Keyboard & mouse ~~can be~~ have individual identity of their own without computer, keyboard & mouse can exist

without person leg & hand can't hold

Adv. of UML class diagrams

1. Class diagms represent the system's classes, attributes, methods & relationships, providing a clear view of its architecture.
2. Shows various relationships (association, generalization)
3. It serve as a visual ~~representation~~ tool for communication among team members & stakeholders.

Date
Page

Unit 3

Relational Models

Tabluler format of storing database

RDBMS (Relational DBMS) → proposed by "Codd"

Codd Rule

Data in DB file must be in tabular format

No two rows of Datafile must be same.

Stud

Sid	Sname	DOB
S_1	A	1990
S_2	B	1991
S_3	B	1990

Relational Instance (snapshot) record set of DB table

Relational Schema :→ Definition/structure of the DB table

Arity :→ no. of fields/attributes of the DB table.

Cardinality :→ No. of records of DB table

Candidate Key :- Minimal set of attributes which can differiante the records uniquely

stud(Sid, Sname, DOB)

Candidate key :→ Sid

Sid Sname → also diff. record uniquely but it is not minimal

classmate
Date
Page

Enroll (Sid Cid fee)

Sid	Cid	fee
S_1	C_1	-
S_1	C_2	-
S_2	C_2	-

Candidate key → (Sid Cid)

ex R

A	B	C
5	4	8
5	4	9
5	6	8
5	6	9
6	4	8

C.K :→ A ✗ B ✗ C ✗
AB ✗ BC ✗ AC ✗
ABC ✓

ex :→ Emp(eid, ename, DOB, pass-no, acc-no, ifsc, panid)

Primary key :→ is one of candidate key whose attr / field (column) value is not null.
ex eid

Almost one PK allowed for any RDBMS table
(0 or 1)

Alternative key :→ All the C.K of relational schema other than P.K.
A.K. fields don't allowed null values.

Simple Candidate Key :- Candidate Key with only 1 attr.
ex eid, panid - etc

Compound C.K. → C.K. with atleast 2 attributes
ex acc-no, IFSC

Prime Attribute → Attribute which belongs to some C.K. schema

Prime att. set of Emp {eid, pan-no, Acc No, IFSC, banid}

Non-Prime att ⇒ Att which does not belong to any CK of schema.

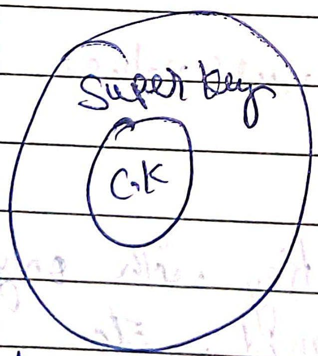
ex {ename, DOB}

ex R(A, B, C, D, E, F, G)
C.K are {A, B, CD, DE}

(CD, DE) (overlapped C.K some common attr.)

Super Key

Set of attr. which can differentiate record uniquely (may not minimal attr. set)



minimal superkey → C.K.

Q How many Superkeys are possible if R(A, B, C, D) & its C.K {A}.

Sol^n

S.K = C.K [Any subset of other attr. (B, C, D)]

= A, AB, AC, AD
ABC, ACD, ABD, ABCD } 8

No. of SKs in $R(A_1, A_2 \ldots A_n)$ with CK $\rightarrow A_1$

Soln $\boxed{\text{No. of S.K} = 2^{n-1}}$

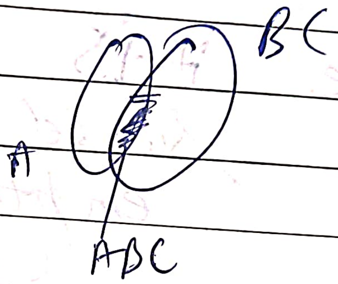
Q. $R(A, B, C, D, E)$ & CK {A, BC}

$$\underset{A}{2^{n-1}} + \underset{BC}{2^{n-2}} - \underset{ABC}{2^{n-3}}$$

$n = 5$

$$2^4 + 2^3 - 2^2$$

$= \underline{20}$



$R(A_1, A_2 \ldots A_n)$

Q. $C.K = \{A_1A_2, A_2A_3\}$

$$\underset{A_1A_2}{2^{n-2}} + \underset{A_2A_3}{2^{n-2}} - \underset{A_1A_2A_3}{2^{n-3}}$$

Q. $R(ABCDEF)$ C.K {A, BC, CD}

$$n(X \cup Y \cup Z) = n(X) + n(Y) + n(Z) - n(X \cap Y) - n(X \cap Z) - n(Y \cap Z) + n(X \cap Y \cap Z)$$

− sign (intersection of even sets)

+ sign (intersection of odd sets)

Q Max S.Ks if $(A_1, A_2, \ldots A_n)$

if every Key is a Candidate key

Sol^n $R(ABC)$

A, AB, AC
B, BC
C, ABC

$$\left.\right\} 7 = 2^3 - 1 = 7$$

i.e. it is the ~~no of~~ total no. of ~~subset~~ in a set except the empty ~~set~~

Referenced Integrity Constraints (foreign key)

Due to

(I) Referenced Relation

(A) Insertion :→ No violation

(B) Deletion :→ can cause violation

(i) On Delete :→ No action
deletion is restricted (default)

(ii) On Delete Cascade :→
On deletion of referenced relation record forced to delete related referencing record

(II) Referencing Relation

(A) Insertion: May cause violation

Sol^n Insertion restricted if violation occurs

(B) Deletion :→ No violation

classmate
Date
Page

(iii) On delete : Set Null
If FK allows to set 'Null' then set null is related FK values.

(c) Updation : May cause violation
(i) On update No action (default)
(ii) On update Cascade
(iii) On update Set Null

Normalizations

Used to eliminate/reduce redundancy in DB tables.

Sid	Sname	DOB	Cid	Cname	Instr	fee
S_1	A	1990	C_1	DB	X	400
S_2	A	1990	C_1	DB	X	400
S_3	B	1998	C_1	DB	X	800
S_3	B	1998	C_2	Algo	Y	700
S_3	B	1998	C_3	OS	Y	800
			C_4	CO	Z	40

Redundancy   Redundancy.

Problems because of Redundancy (DB anomalies)

1) Insertion Anomalies
2) Deletion Anomaly
3) ~~De~~ Updation Anomaly

Insertion Anomaly :→ In order to store some data we have to store data for all relation

here for (C4) we have to add student info.

Deletion Anomaly :→ If we have to delete stud. info we are forced to delete its subject info

eg Delete stud info (S3)

with S3 → Info regarding course is also lost.

Updation Anomaly :→

In updation with some redundant copies we have to update in all places.

0% redundancy in DB table ⟺ No DB anomalies

To reduce redundancy we decompose the table into subrelation ~~to~~, it is called Decomposition

R_1

Sid	Sname	DOB
S_1		
S2		
S3		

Sid (FK)	Cid (FK)	fee
S_1	C1	
S2	C1	
S3	C1	
S3	C2	
S3	C3	

Cid	Cname	Instr
C1		
C2		
C3		

this is now a normalised DB design
No redundancy, No anomalies

Functional Dependency (FD)

X, Y some sets of attributes of relation R t_1, t_2 any tuples of R

$\boxed{X \rightarrow Y}$ FD implied relation R

if $\boxed{t_1.X = t_2.X \quad \text{then } t_1.Y = t_2.Y}$

meaning if left hand value is same in 2 or more tuples then right hand value should be same.

R

	X	Y
t_1	x_1	y_5
t_2	x_1	y_5
	x_2	y_2
	x_3	y_2
	x_4	y_4
	x_2	y_2
	x_1	y_5

$X \rightarrow Y$ ✓

$Y \rightarrow X$ ✗

If left hand value is different then right hand value may or may not be same.

eg

Stud

Sid	Sname	Cid
S_1	A	C_1
S_1	A	C_2
S_1	A	C_3
S_2	B	C_3
S_2	B	C_4
S_3	C	C_1
S_4	C	C_2
S_5	D	C_3

Sid → Sname ✓

Sname → Sid ✗

Cid → Sname ✗

Sid → Cid ✗

Cid → Sid ✗

Types of FD's

Trivial FD (Self FD)	Non-Trivial FD	Semi-Non trivial FD
Sid → Sid Sname → Sname Sid Sname → Sname Sid Sname → Sid Sname X → Y is trivial FD of R if $X \supseteq Y$	X → Y are non trivial if $\boxed{X \cap Y = \Phi}$ Sid → Sname Sid Cid → Sname	Combination of trivial & non-trivial Sid → Sid Sname Sid Cid → Cid Sname

Armstrong rules over FD

X, Y, Z some att. sets over R

1. Reflexivity (Trivial FD)
   X → X
2. Transitivity
   if X → Y & Y → Z then X → Z
3. Augmentation
   if X → Y then XZ → YZ
4. Split Rule if X → YZ then X → Y, X → Z
5. Merge Rule (Union Rule) if X → Y, X → Z then X → YZ

classmate
Date
Page

Attribute Closure (X^+)

X is some attr. set of rel. R

X^+ = {set of all attributes which can be determined by X}

R(ABCDEFG)

ex given FD = {A→B, C→D, AB→E, BE→C, EF→G}

A^+ = {ABECD} × BE^+ = {BECD} ×

AF^+ = {ABECDFG} ✓

Super Key :→

X is superkey of R iff X^+ must determine all attributes of R.

Candidate Key :→

X is C.K. of R

iff (i) X must be superkey of R
X^+ = {all attr. of R}

(ii) No proper subset of X is superkey of R
$\forall Y \subset X$ such that Y^+ not determine all att. of R, then X is C.K. of R

ex R(ABCDE)

{AB→C, C→D, B→E} find CK of R?

sol AB^+ = {ABCDE} ✓

& no proper subset of AB is S.K. so AB is C.K.

A^+ → {A}

B^+ → {BE}

ex $R(ABCDE)$

$\{AB \to C, BC \to D\}$

ex $R(ABCDEF)$

$\{AB \to C, C \to D, CD \to BE, DE \to F, EF \to A\}$

CK $\{AB, BEF, BDE, C\}$

ex $R(ABCDEF)$

$\{AB \to C, C \to DE, E \to F, F \to B\}$

C.K $\{AB, AF, AE, AC\}$

ex $R(ABCDE)$

$\{A \to B, B \to C, C \to D, D \to A\}$

$\{AE, DE, CE, BE\}$

ex $R(ABCDEF)$

$\{AB \to C, C \to A, BE \to C, EC \to FA,$
$BC \to D, ACD \to B, CF \to BD, D \to E\}$

CK $\{AB, BC, CD, BE, BD, CF, EC\}$

Q $R(ABCDEH)$

$\{A \to B, BC \to D, E \to C, D \to A\}$

CK = $\{AEH, BEH, DEH\}$

R(ABCDE)

Q $\{A \to B,\ A \to C,\ CD \to E,\ B \to D,\ E \to A\}$
which is not implied by above set.

(a) $CD \to AC$ (b) $BD \to CD$ (c) $BC \to CD$
(d) $AC \to BC$

Membership Test

$X \to Y$ FD is member of FD set (F)
iff X^+ must determines Y in FD set (F)

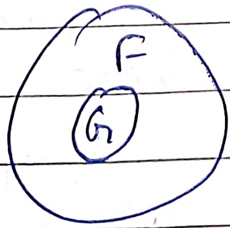
Equality of FD sets

F & G sets logically are equal iff

(i) F covers G :- Every FD of G set must be member of F set

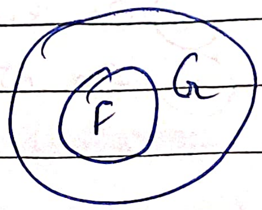
(ii) G covers F :- Every FD of F set must be member of G set.

F covers G ⇒



$G \subseteq F$

G covers F



$F \subseteq G$

Q $F = \{A \to B, B \to C, C \to A\}$

$G = \{A \to BC, B \to AC, BC \to A, AB \to C\}$

which is True

(a) $F \subset G$ (b) $F \supset G$ (c) $F = G$ (d) None

Q R(A, B, C, D, E, F)

$A \to B$, $A \to C$, $CD \to E$, $CD \to F$, $B \to E$

(i) Prove $CD \to EF$ is True
(ii) Prove $AD \to EF$ is True

Use on Reflexive, Transitivity & Augmentation

Canonical form / minimal cover

↳ irreducible set of FD.

eg R(WXYZ)

$X \to W$

$WZ \to XY$

$Y \to WXZ$

There can be three types of redundancy here.

$\alpha \to \beta$

(i) On left side
(ii) On right side
(iii) On whole FD

How to solve it?

classmate

Date

Page

Sol^n $X \to W$ (a)

Step 1 $WZ \to X$ (b)

$WZ \to Y$ (c)

$Y \to W$ (d)

$Y \to X$ (e)

$Y \to Z$ (f)

Step 2

Find closure of each FD two times such that one with that FD included one time excluded if both closures are same means that FD is redundant

$X^+ = \{XW\}$ → with (a) included

$X^+ = \{X\}$ without (a)

didn't get the same closure so (a) is essential

$(WZ)^+ = \{WZXY\}$ → with (b)

$(WZ)^+ = \{WZYX\}$ → without (b)

so $WZ \to X$ is redundant remove it

$(WZ)^+ = \{WZY\}$ with (c)

$(WZ)^+ = \{WZ\}$ without (c) essential

$Y^+ = \{WXZY\}$ with (d)

$Y^+ = \{XZWY\}$ without (d)

remove $Y \to W$

$Y^+ = \{YXW\}$ with (e)

$Y^+ = \{YZ\}$ without (e) essential

$Y^+ = \{YZ\}$ with f

$Y^+ = \{Y\}$ without f essential

Date
Page

$X \to W$
$WZ \to Y$
$Y \to X$ left side Redundancy is done
$Y \to Z$

<u>for Right Side</u>

$WZ^+ = WZYX$
$W^+ = W$] if here W^+ can determine
$Z^+ = Z$] (WZYX) then Z is redundant & vice-versa

So here there are no redundancy

Now combine them

$X \to W$
$WZ \to Y$ } Canonical form
$Y \to XZ$

There can be more than one solution to this depending on the order we are checking.

Q $\{A \to B,\ C \to B,\ D \to ABC,\ AC \to D\}$

Sol^n
$A \to B$ → $A^+ = \{A\}$ without this FD
$C \to B$ → $C^+ = \{C\}$ " " "
$D \to A$ → $D^+ = \{DBC\}$ " " "
~~$D \to B$~~ → $D^+ = \{ABCD\}$ → " " → Redundant
$D \to C$ → $D^+ = \{DCAB\}$ " " "
$AC \to D$ → $AC^+ = \{ACB\}$ " "

classmate
Date
Page

$A \to B$, $C \to B$, $D \to A$, $D \to C$, $\underline{AC \to D}$

for Region

$A^+ = \{AB\}$
$C^+ = \{CB\}$ } not Redundant

Q ~~R(ABCD)~~

Q R(ABCDE)

$A \to B$
$AB \to C$
$D \to ACE$

Soln
$A \to B$ ⇒ $A^+ = \{A\}$ ✓
$AB \to C$ $AB^+ = \{AB\}$ ✓
$D \to A$ $D^+ = \{DCE\}$ ✓
~~$D \to C$~~ $D^+ = \{DAEBC\}$ Not essential
$D \to E$ $D^+ = \{DABC\}$ ✓

$A \to B$, $\underline{AB \to C}$, $D \to A$, $D \to C$
$AB^+ = \{ABC\}$
$A^+ = \{ABC\}$ ~~B is redundant here~~ B is redundant

$A \to B$, $A \to C$, $D \to A$, $D \to E$

$\boxed{A \to BC,\ D \to AE}$

$\bowtie \rightarrow$ natural join

Date ____
Page ____

Properties of Decomposition

1. Lossless Join decomposition
2. Dependency Preserving decomposition.

Lossless Join decomposition

Relational Schema R with instance r decomposed into subrelation $R_1, R_2, \ldots R_n$

(a) In general

$$(R_1 \bowtie R_2 \bowtie R_3 \bowtie R_4 \ldots \bowtie R_n) \supseteq R$$

(b) if $(R_1 \bowtie R_2 \bowtie R_3 \ldots \bowtie R_n) = R$
then lossless join decomposition

(c) if $(R_1 \bowtie R_2 \bowtie R_3 \ldots \bowtie R_n) \supset R$
then lossy join decomposition.

ex

R

Sid	Sname	Cid
S_1	A	C_1
S_1	A	C_2
S_2	B	C_2
S_3	B	C_3

{ Sid → Sname }

C.K = SidCid

(i) decompose R into subrelation

R_1

Sid	Sname
S_1	A
S_2	B
S_3	B

R_2

Sid	Cid
S_1	C_1
S_1	C_2
S_2	C_2
S_3	C_3

CK = Sid Cid

Sid → Sname

CK → Sid

Join $R_1 \bowtie R_2$

Sid	Sname	Cid
S_1	A	C_1
S_1	A	C_2
S_2	B	C_2
S_3	B	C_3

$R_1 \bowtie R_2 = R$

then this is lossless join

(ii) decompose R into subrelations

R_1

Sid	Sname
S_1	A
S_2	B
S_3	B

Sid → Sname

CK = Sid

R_2

Sname	Cid
A	C_1
A	C_2
B	C_2
B	C_3

CK = Sname Cid

Join $R_1 \bowtie R_2$

Sid	Sname	Cid
S_1	A	C_1
S_1	A	C_2
S_2	B	C_2
S_2	B	C_3
S_3	B	C_2
S_3	B	C_3

$R_1 \bowtie R_2 \supset R$

Lossy join decomposition

→ not present in R

these records are called as spurious Tuples

classmate
Date
Page

if Relational schema R with FD set (F) decomposed into subrelations R_1, R_2

Given decomposition is lossless iff

(i) $R_1 \cup R_2 = R$ → means all attributes of R should be covered.

and

(ii) $(R_1 \cap R_2) \to R_1$
or
$(R_1 \cap R_2) \to R_2$

means common attribute of both R_1 and R_2 should be a super key for any one subrelation.

ex $R(ABCDE)$ $\{AB \to C, C \to D, B \to E\}$

(i) decompose into $\{ABC, CD\}$
$R_1 \cup R_2 \neq R$ so lossy

(ii) decompose $\{ABC, DE\}$
$R_1 \cup R_2 = R$
$R_1 \cap R_2 = \phi$ so lossy

(iii) decompose $\{ABC, CDE\}$
$R_1 \cup R_2 = R$
$R_1 \cap R_2 = C$ $C^+ = \{CD\}$ not a superkey
so lossy.

R_1 ~~R_2~~ R_2 R_3

(iv) $\{ABC, CD, DE\}$

$R_1 \cup R_2 \cup R_3 = R$

$R_1 \cap R_2 = C$ $C^+ = \{CD\}$ super key for R_2

$R_1 R_2$

$R_1 R_2$ R_3

$\{ABCD, DE\}$

$R_{12} \cap R_3 = D$ $D^+ = \{D\}$ not a S.K.

So lossy.

R_1 R_2 R_3

(v) ~~R(AB)~~ $\{AB\}$ $\{CD\}$ $\{BC\}$

$R_1 \cup R_2 \cup R_3 \neq R$ lossy

(vi) $\{ABC, CD, BE\}$

C^+ {C is a SK}

ABCD BE

$\exists$ B^+ {BE}

ABCDE lossless join decomposition

Dependency Preserving Decomposition

Relational Schema R with FD set F decomposed into subrelation $R_1, R_2 -- R_n$ with FD set $F_1, F_2 -- F_n$

In general

$$\{F_1 \cup F_2 \cup F_3 — F_n\} \subseteq F$$

(i) if $\{F_1 \cup F_2 -- F_n\} = F$ then dependency preserving

(ii) if $\{F_1 \cup F_2 \cup — F_n\} \subset F$ then not dependency preserving.

mean some FDs are lost.

Q R(ABCDE)

$\{A \to B, B \to C, C \to D, D \to BE\}$

decompose {AB, BC, CD, DE}

R_1(AB)	BC	CD	DE
✓ $\boxed{A \to B}$ $\{A^+ = ABCDE\}$	$\boxed{B \to C}$ ✓ $B^+ = \{BCDE\}$	$\boxed{C \to D}$ ✓ $C^+ = \{CDBE\}$	$\boxed{D \to E}$ ✓ $D^+ = \{DBC\}$
✗ $B \to A$ $\{B^+ = BC\}$	$\boxed{C \to B}$ ✓ $C^+ = \{CDBE\}$	$\boxed{D \to C}$ ✓ $D^+ = \{DBEC\}$	$E \to D$ ✗ $E^+ = \{E\}$
F_1	F_2	F_3	F_4

$F_1 \cup F_2 \cup F_3 \cup F_4$

$\Rightarrow \{A \to B, B \to C, C \to B, C \to D, D \to C, D \to E\}$

$D^+ = \{DCEB\}$ → so B can also be determined by D.

So $F_1 \cup F_2 \cup F_3 \cup F_4 = F$ So

Dependency Preserving

Q $R(ABCD)$

$\{AB \to CD,\ D \to A\}$

Soln →

$R_1(ABC)$	$R_2(BCD)$	$R_3(AD)$
$\boxed{AB \to C}$ ✓	$\boxed{BD \to C}$ ✓	$\boxed{D \to A}$ ✓
$AB^+ = \{ABCD\}$	$B^+ = B$	$A^+ = A$
$A^+ = (A)$	$C^+ = C$	$D^+ = DA$
$B^+ = (B)$	$D^+ = DA$	
$C^+ = (C)$	$BC^+ = BC$	
$AB^+ = (ABCD)$	$BD^+ = BDAC$	
$BC = (BC)$		
$AC \to (AC)$		

If AB^+ contains CD by using these 3 FDs then we can say it is D.P.

$AB^+ = \{ABC\}$ ✗ So Not D.P.

Q $R(ABCD)$

$\{AB \to C,\ C \to D,\ D \to A\}$

$R_1(ABC)$	$R_2(CD)$
$A^+ = \{A\}$	$C^+ = \{CD\}$ $\boxed{C \to D}$
$B^+ = \{B\}$	$D^+ = \{DA\}$
$C^+ = \{CDA\}$ $\boxed{C \to A}$	
$AB^+ = \{ABCD\}$ $\boxed{AB \to C}$	
$BC^+ = \{BCDA\}$ $\boxed{BC \to A}$	
$AC^+ = \{ACD\}$ ϕ	

$\{C \to A, AB \to C, BC \to A, C \to D\}$.

$[F_1 \cup F_2 \subset F]$ $(D \to A)$ not preserved

Normal Forms

Used to identify degree of redundancy.

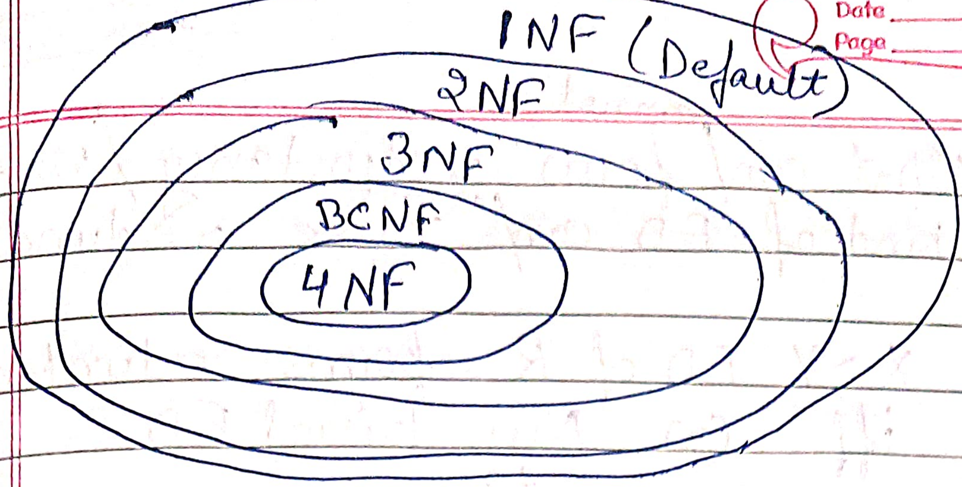
Redundancy in DB relation

Non-Trivial FDs (single valued dependency) $(X \to Y)$

Non-Trivial MVD's (multivalued dependency) $(X \to\to Y)$

over FD set (single valued)
- 1 NF
- 2 NF
- 3 NF
- BCNF → 0% Redundancy but for MVD redundancy may exist.

4 NF → 0% Redundancy over FD's & MVD



1NF

Relation is in 1NF iff
→ No Multivalued attributes in R.

→ Default NF of RDBMS relation.

Sid	Sname	Cid
S_1	A	C_1/C_2
S_2	B	C_2/C_3
S_3	B	C_3

Not in NF
Not in RDBMS.

⇓

Sid	Sname	Cid
S_1	A	C_1
S_1	A	C_2
S_2	B	C_2
S_2	B	C_3
S_3	B	C_3

C.K → Sid Cid

1NF

1NF 2NF 3NF BCNF 4NF

Default NF

Used to reduce/eliminate Redundancy.

What can/cannot form redundancy / What kind of FD can/cannot cause redundancy?

1. $X \to Y$ FD of R forms redundancy iff (a) Non-Trivial FD
&&
(b) 'X' is not superkey of R.

2. $X \to Y$ FD of R not form redundancy iff (a) Trivial FD ($X \supseteq Y$)
or
(b) X is a superkey of R

$X \to Y$
↑
S.K.

ex R(Sid Sname Cid)
C.K. = Sid Cid

S_1	A	C_1
S_1	A	C_2
S_1	A	C_3
S_2	B	C_2
S_2	B	C_3

X → Y
Sid → Sname
↑
not S.K.
So can have redundancy.

classmate

Date

Page

ex $R(ABCDE)$

$\{AB \to C,\ B \to D,\ D \to E,\ AE \to F,\ C \to A\}$

$AB^+ = \{ABCDEF\}$

$BC^+ = \{BCDEAF\}$

$C.K = \{AB, BC\}$

$\underbrace{AB}_{S.K} \to C,\ \underbrace{B}_{} \to D,\ \underbrace{D}_{} \to E,\ \underbrace{AE}_{} \to F,\ \underbrace{C}_{} \to A$

Not S.K

form Redundancy.

Note :-

(i) Proper Subset of CK ⟶ Non-Prime $(B \to D)$

(ii) Non-Prime ⟶ Non-Prime $(D \to E)$

(iii) Proper Subset of CK and non-prime combination ⟶ Non-Prime $(AE \to F)$

(iv) Proper Subset of CK → Proper Subset of CK $(C \to A)$

These form redundancy.

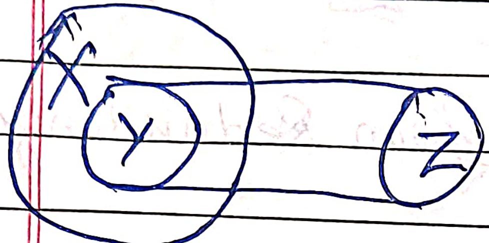
2NF: R is in 2NF iff

(i) It is in 1NF already

(ii) No partial dependencies in R.

Partial Dependency :-

X is any c.k. of R.
$Y \subset X$, Z is non-prime attr. of R,
then $\boxed{Y \to Z}$ is partial dependency.
Prime → Non-Prime



3NF :

R is in 3NF iff
(i) It is already 2NF
&&
(ii) Every non-trivial FD $X \to Y$ in R with
(a) X must be superkey of R.
or
(b) Y must be prime attr. of R.

$X \to Y$ → 3NF
X: SK, Y: Prime

$X \to Y$ → 3NF
X: Prime, Y: Prime

$X \to Y$ → 3NF
X: Non Prime, Y: Prime

$X \to Y$ → 3NF
X: SK, Y: Non-prime

Proper Subset of CK → Proper Subset of CK

BCNF (Boyce Codd NF)

R is in BCNF iff
(i) ~~It~~ is already in 3NF &&
(ii) Every non-trivial FD $X \rightarrow Y$ in R with X must be S.K.

Find highest NF of relation R

Q R(ABCDEF) $\{ABD \rightarrow C,\ BC \rightarrow D,\ CD \rightarrow E\}$

Sol^n

$ABDF^+ = \{ABCDEF\}$
$ABCF^+ = \{ABCDEF\}$ CK {ABDF, ABCF}

$\underbrace{ABD}_{\text{Prime}} \rightarrow \underbrace{C}_{\text{Prime}}$ 3NF

$BC \rightarrow \underbrace{D}_{\text{Prime}}$ 3NF

$\underbrace{CD}_{\text{Prime}} \rightarrow \underbrace{E}_{\text{Non-Prime}}$ Not in 2NF

Not in 2NF
So highest NF is 1NF

Q R(ABCDE) $\{ABD \rightarrow C,\ BC \rightarrow D,\ CD \rightarrow E\}$

C.K. = {ABD, ABC}

$\underbrace{ABD}_{\text{S.K}} \rightarrow C$ BCNF

$BC \rightarrow \underbrace{D}_{\text{Prime}}$ 3NF

$CD \rightarrow E$ non Prime

CD is not a proper subset of any candidate key
So we have to check 2NF

ABD → $A^+ = A$
$B^+ = B$
$D^+ = D$
$AB^+ = AB$
$BD^+ = BD$
$AD^+ = AD$

ABC ⇒ $C^+ = C$
$AC^+ = AC$
$BC^+ = BCDE$

$\underbrace{BC}_{\text{Prime}} \rightarrow \underbrace{E}_{\text{Non-Prime}}$

Not in 2NF

Highest NF is 1NF

Q R(ABCD) {AB → C, BC → D}

Sol^n

C.K. = {AB}

$\underbrace{AB}_{SK} \rightarrow C$ → BCNF

$BC \rightarrow \underbrace{D}_{\text{Non-Prime}}$

Not in 3NF

2NF Test

AB ⇒ $A^+ = A$
$B^+ = B$

no partial dependencies
so 2NF

Highest 2NF

Q R(ABCD) {AB → C, C → A, AC → D}

Sol^n

$AB^+ = \{ABCD\}$ C.K = {AB, BC}

$BC^+ = \{BCAD\}$

$\underbrace{AB}_{\text{S.K.}} \to C$ BCNF

$C \to \underbrace{A}_{\text{Prime}}$ 3NF

$\underbrace{AC}_{} \to \underbrace{D}_{\text{Non-Prime}}$

2NF Testing

AB → $A^+ = A$, $B^+ = B$

BC → $B^+ = B$, $C^+ = CAD$ (D: Non-Prime)

C → D

Partial Dependencies

So Highest NF is 1NF.

Q R(ABCDEF) {AB → C, C → DE, E → F, F → B}

Sol^n :- C.K = {AB, AF, AE, AC}

$\underbrace{AB}_{\text{S.K.}} \to C$

$\underbrace{C}_{\text{Prime}} \to DE$

$E \to \underbrace{F}_{\text{Prime}}$ 3NF

$F \to \underbrace{B}_{\text{Prime}}$ 3NF

$C \to \underbrace{D}_{\text{Non-prime}}$

$C \to \underbrace{E}_{\text{Prime}}$ → 3NF

2NF Testing

AB → $A^+ = A$, $B^+ = B$

AF → $F^+ = FB$

$AE \rightarrow E^+ = EFB$

$AC \rightarrow C^+ = CDE$ → Non-prime → P.D.

Not in 2NF

Highest NF is 1NF.

Q R(ABCDEF)

{AB → C, C → D, CD → AE, DE → F, EF → B}

Soln Ck. (AB, AEF, C, ADE)
(if all attr. are prime attr. then relation R is 3NF atleast)

$\underbrace{AB}_{SK} \rightarrow C$, $\underbrace{C}_{SK} \rightarrow \underbrace{D}_{Prime}$, CD → AE

BCNF, BCNF

CD → A Prime

CD → E

3NF

$DE \rightarrow \underbrace{F}_{Prime}$ $EF \rightarrow \underbrace{B}_{Prime}$

3NF 3NF

classmate

Date

Page

Q R(ABCDE)

{A→B, B→C, C→D, D→A}

Sol^n

C.K. (AE, BE, CE, DE)

all attributes are prime attr so atleast 3NF

$\underset{\text{Prime}}{A \to B}$ 3NF

$\underset{\text{Prime}}{B \to C}$ 3NF

$\underset{\text{Prime}}{C \to D}$ 3NF

$\underset{\text{Prime}}{D \to A}$

3NF

Q R(ABCD)

{A→B, B→AC, C→D}

Sol^n C.K. {A, B}

$\underbrace{A}_{SK} \to B$ BCNF

$\underbrace{B}_{SK} \to AC$ BCNF

C → D
Non-Prime non-Prime
not in 3NF

2NF Testing

Here CK is single attribute so proper subset of CK is empty set & empty set can't derive non-prime att.

So no Partial Dependencies.

2NF

Q Relational Schema R with no non-trivial FD's
What is the highest NF of R.

(i) 2NF (ii) 3NF (iii) BCNF (iv) 4NF

Sol^n

(No non-trivial FD) ⟹ (No Redundancy over FD's in R)

⇕

R in BCNF
may not in 4NF

Q A relational Schema R with only two attributes is always in BCNF (also in 4NF). True or false.

Sol^n

Let R(AB)
what are the options for FD here

(i) $A \to B$ $\quad C.K = \{A\}$ $\quad \underbrace{A}_{SK} \to B$ so BCNF

(ii) $B \to A$ $\quad CK = \{B\}$ $\quad \underbrace{B}_{SK} \to A$ BCNF

(iii) $A \to B,\ B \to A$ $\quad CK\{A, B\}$ $\quad \underbrace{A}_{SK} \to B$ $\quad \underbrace{B}_{SK} \to A$ BCNF

(iv) No non-trivial FD ⟹ BCNF

True.

LLJ → Lossless Join
PD → Partial Dependency
DP → Dependency Preserving

Decomposition of Rel. into higher NF

Rel. decompose into 2NF, 3NF, BCNF with lossless join & Dependency Preserving.

2NF decomposition

R(- - - - XY - - -) FD → { - - - , (X → Y), - - }

↓ Partial Dependency

$X^+ = \{XY\}$

Take closure of X & make another relation

R_1(- - - X - -)
Remaining all attr. with determinant of P.D.

R_2(XY)
{X → Y}
CK = {X}

Lossless ✓
DP ✓
PD ✗

ex. R(ABCDE)
{A → B, B → E, C → D}

Sol^n CK = {AC}

A → B : Prime → Non-Prime, P.D
B → E : Non Prime → Non Prime, NO P.D.
C → D : Prime → Non-Prime, P.D.

R(ABCDE)

$A^+ = \{ABE\}$ C^+

R_1(ABE)
A → B BCNF
SK
B → E
no P.D.

R_2(CD)
C → D
SK

R_3(AC)

So 2NF

P → Prime attr.
NP → Non-Prime attr.

ABD

classmate
Date
Page

Q $R(ABCDEFGHIJ)$

$F = \{AB \to C,\ AD \to GH,\ BD \to EF,\ A \to I,\ H \to J\}$

Soln $CK = \{ABD\}$

$\underbrace{\underset{P}{AB} \to \underset{NP}{C}}_{PD}$, $\underbrace{\underset{P}{AD} \to \underset{N.P}{GH}}_{PD}$, $\underbrace{\underset{P}{BD} \to \underset{N.P.}{EF}}_{PD}$, $\underbrace{\underset{P}{A} \to \underset{NP}{I}}_{PD}$, $\underset{NP}{H} \to \underset{NP}{J}$

I is also in the closure of AB & is of A→I is But P.D. so we will not take it.

$R(ABCDEFGHIJ)$

$AB^+ = (ABC\cancel{I})$ $AD^+ = (ADGHJ)$ $BD^+ = (BDEF)$ $A^+ = AI$

$R_1(ABC)$ $R_2(ADGHJ)$ $R_3(BDEF)$ $R_4(AI)$ $R_5(ABD)$

Q $R(ABCDEF)$ $\{AB \to C,\ C \to D,\ B \to E,\ E \to F\}$
BCNF (under $AB \to C$), PD. (under $B \to E$)

Soln $CK = (AB)$

$R(ABCDEF)$

$B^+ = \{BEF\}$

$R_1(ABCD)$ $R_2(BEF)$ → LJ, DP, No PD, 2NF

$AB \to C$, $C \to D$ Not in 3NF

$B \to E$, $E \to F$ Not in 3NF

$C^+ = \{CD\}$ $E^+ = (EF)$

$R_{11}(ABC)$ $R_{12}(CD)$ $R_{21}(BE)$ $R_{22}(EF)$

$AB \to C$ BCNF; $C \to D$ BCNF; $B \to E$ BCNF; $E \to F$ BCNF

4 tables & 3 Foreign Keys.

Q1 R(ABCDE)
{AB → C, BC → A, AC → B}

Soln CK {ABDE, ACDE, BCDE}

Q2 R(ABC) {AB → C, C → A}
CK (AB, BC)

Q3 R(ABCD) {ABC → D, D → B}

Q4 R(ABCD) {AB → C, C → D, A → B}

Q5 R(ABCD) {A → B, B → C, C → D, D → A}

Q6 R(ABCDE) {A → B, BC → D, D → BC,
DE → ∅}

Q7 R(ABCD) {A → E, B → E, A → F}

Q8 R(ABCD) {AB → CD, C → A, D → B}

Sol: 1 R(ABCDE)

$\{AB \to C,\ BC \to A,\ AC \to B\}$

CK $\{ABDE,\ ACDE,\ BCDE\}$

all attr. are prime attr. so atleast 3NF
But not in BCNF because left side of all ~~E~~ FDs is not a S.K.

R(ABCDE)

AB^+

$R_1(ABC)$ — $R_2(ABDE)$

$AB \to C$ BCNF

$BC \to A$ BCNF

$AC \to B$ ~~Not in~~ BCNF

No relevant FD. So BCNF

Lossless Join ✓
D.P. ✓
BCNF ✓

2 Tables & 1 Foreign key

Sol 2. R(ABC) $\{\underbrace{AB}_{SK} \to C,\ \underbrace{C}_{P} \to \underbrace{A}_{P}\}$

BCNF 3NF

Ck $\{AB, BC\}$

R(ABC)

$C^+ = CA$

$R_1(BC)$ & $R_2(AC)$

no non-trivial FD so BCNF

$C \to A$ BCNF

Lossless Join ✓
D.P. ✗
BCNF ✓

2 Table
1 Foreign key

BCNF does not guarantee Dependency Preserving.

Sol^n 3 R(ABCD) { $\underbrace{ABC}_{SK} \to D$, $D \to B$ }

CK(ABC, ACD) BCNF 3NF

R(ABCD)

$D^+ = \{DB\}$

$R_1(ACD)$ $R_2(BD)$

No non-trivial FD $D \to B$

BCNF BCNF

2 Table &
1 F.K.

LLJ ✓
DP ✗
BCNF ✓

Sol^n 4 R(ABCD) { $\underbrace{AB}_{SK} \to C$, $C \to D$, $A \to B$ }

CK {A} BCNF NP NP SK
Not in 3NF BCNF

R(ABCD)

$C^+ = (CD)$

$R_1(\underline{A}BC)$ $R_2(\underline{C}D)$

$AB \to C$
$A \to B$
SK
BCNF

$C \to D$
SK
BCNF

LLJ ✓
DP ✓
BCNF ✓

2 Table &
1 f.K.

Sol^n 5 $R(ABCD)$ $\{A \to B,\ B \to C,\ C \to D,\ D \to A\}$

$\underset{SK}{A}$ $\underset{SK}{B}$ $\underset{SK}{C}$ $\underset{S.K}{D}$

C.K. = $\{A, B, C, D\}$

BCNF

Sol^n 6 $R(ABCDE)$

$\{A \to B,\ BC \to D,\ D \to BC,\ DE \to D\}$

P N.P. P P

C.K $\{ACE, ADE\}$ ↑ Partial Dependency 3NF irrelevant X (trivial)

P NP

$D \to B$ → Partial Dependency

$D \to C$ → 3NF

↑ Prime

$R(AB\cancel{C}DE)$

$A^+ = \{AB\}$ $D^+ = \{DBC\}$

$R_1(AB)$ $R_2(DBC)$ $R_3(ADE)$

$A \to B$ $D \to C$ BCNF

SK SK

BCNF BCNF

[LJ] ✓

~~AD~~ DPX

BCNF ✓

3 Table &

2 F.K.

Sol^n 7 R(ABCD) {A → E, B → E, A → F}

irrelevant F.D.

So C.K = {ABCD} ← all attr. are in C.K.

So

BCNF

SK BCNF 3NF 3NF

P P P P

Sol^n 5 R(ABCD) {AB → CD, C → A, D → B}

C.K. = {AB, AD, BC, CD}

R(ABCD)

$C^+ = \{CA\}$ $D^+ = \{BD\}$

$R_1(AC)$ $R_2(BD)$ $R_3(CD)$ LLJ ✓

C → A D → B DP ✗

BCNF BCNF BCNF BCNF ✓

3 Table & 2 F.K.

	1NF	2NF	3NF	BCNF	4NF
Lossless Join	✓	✓	✓	✓	may not always
Dependency Preserving	✓	✓	✓	may not always	may not always
0% Redundancy	No	No	No	Yes over single valued FD	Yes over both FDs & MVD's (multi valued dependency)

For DB design Top priority is lossless Join then DP & then 0% redundancy

Most accurate NF → 3NF
because Lossless Join & D.P. both are followed.