Sadman
DDMS (D) [redacted]
01122159 2

C.L. 23.1.24

Class-1

DBMS

* Data
* Information
* Web-based information system.
* two-tier, three-tier
* Database, DBMS
* DBMS → key features
* User → web server → SQL → DBMS
  High level → standard level → machine Language → Database
  Low level

* Database system structure
  → Query procesor
  → storage manager
* DDL, DML, Query evaluation engine

C.W
27.1.24

Class - 2

DBMS

* DDL → Data Definition Language
(Data type)
(int, string)

* DML → Data manipulation Language
(values)

C.W
30.1.24

DBMS

→ All keys

→ All attributes.

SK → { A, B, AB, CDE, ACDE, BD, ABCD, EABC }
↓
super key

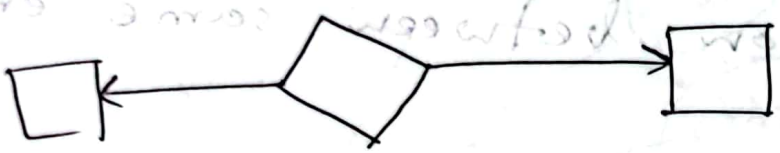
CK → { A, B, CDE }
↓
Candidate key

CW
3.2.21

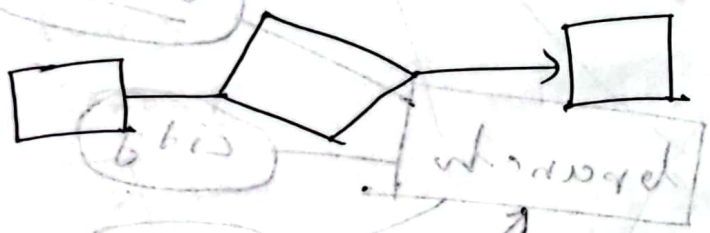
DBMS

Saturday
C.T

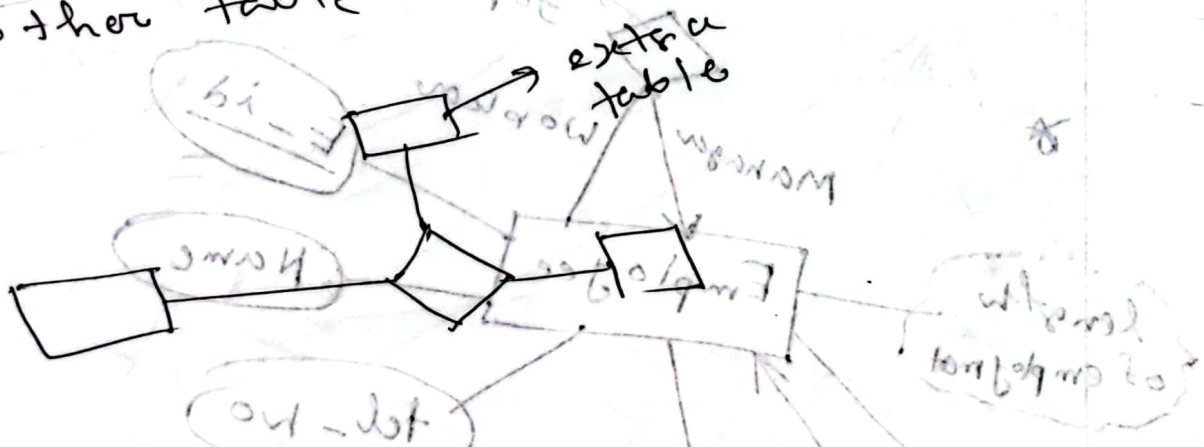
One to one



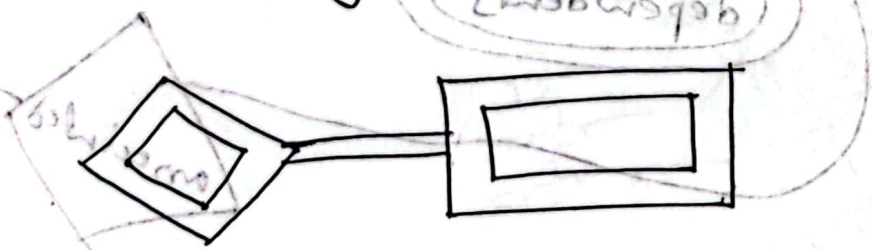
many to one



If we can't use table with another table



* Weak entity set

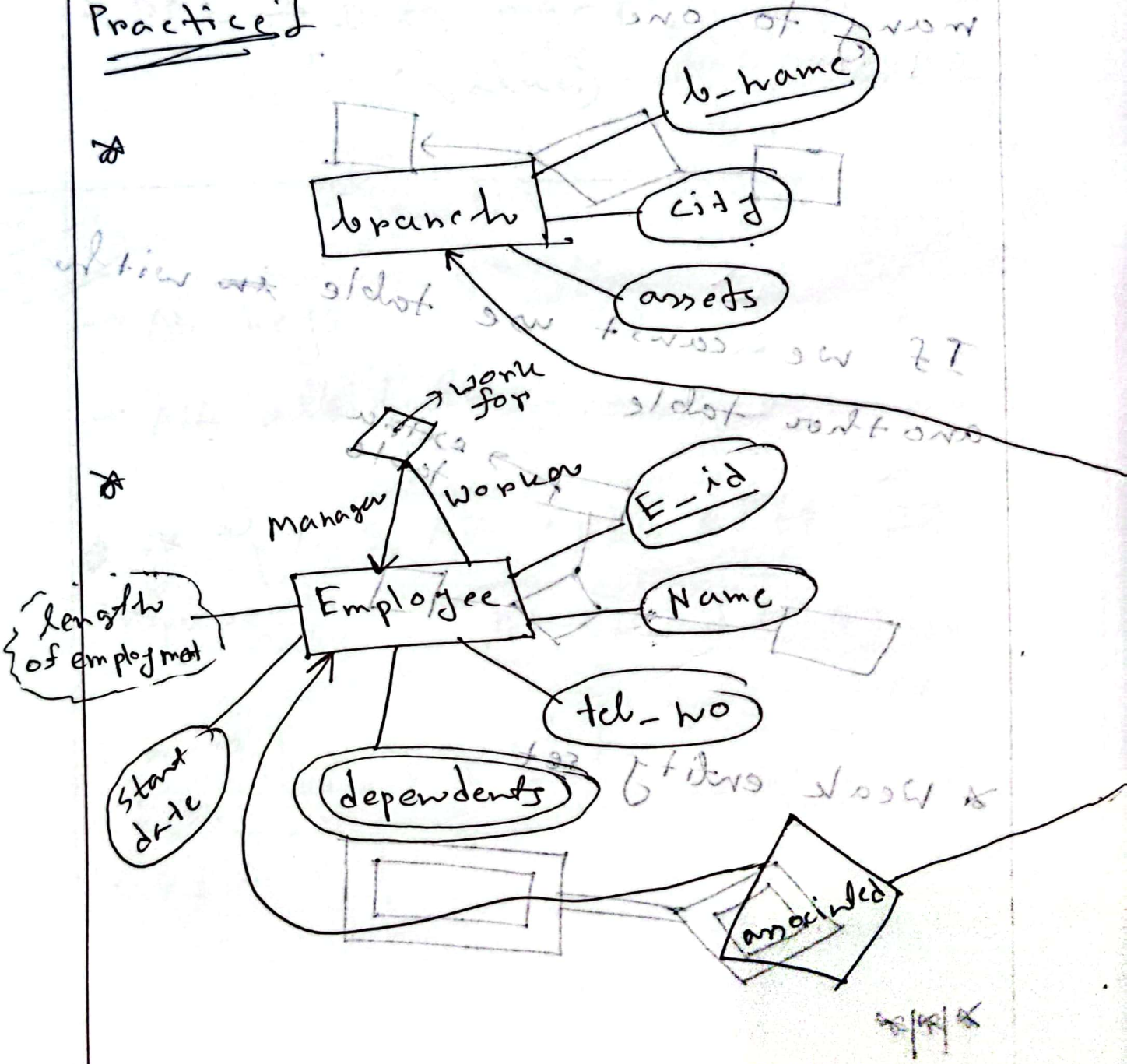


*

Recursive Relationship set

→ Relation between same entity set.

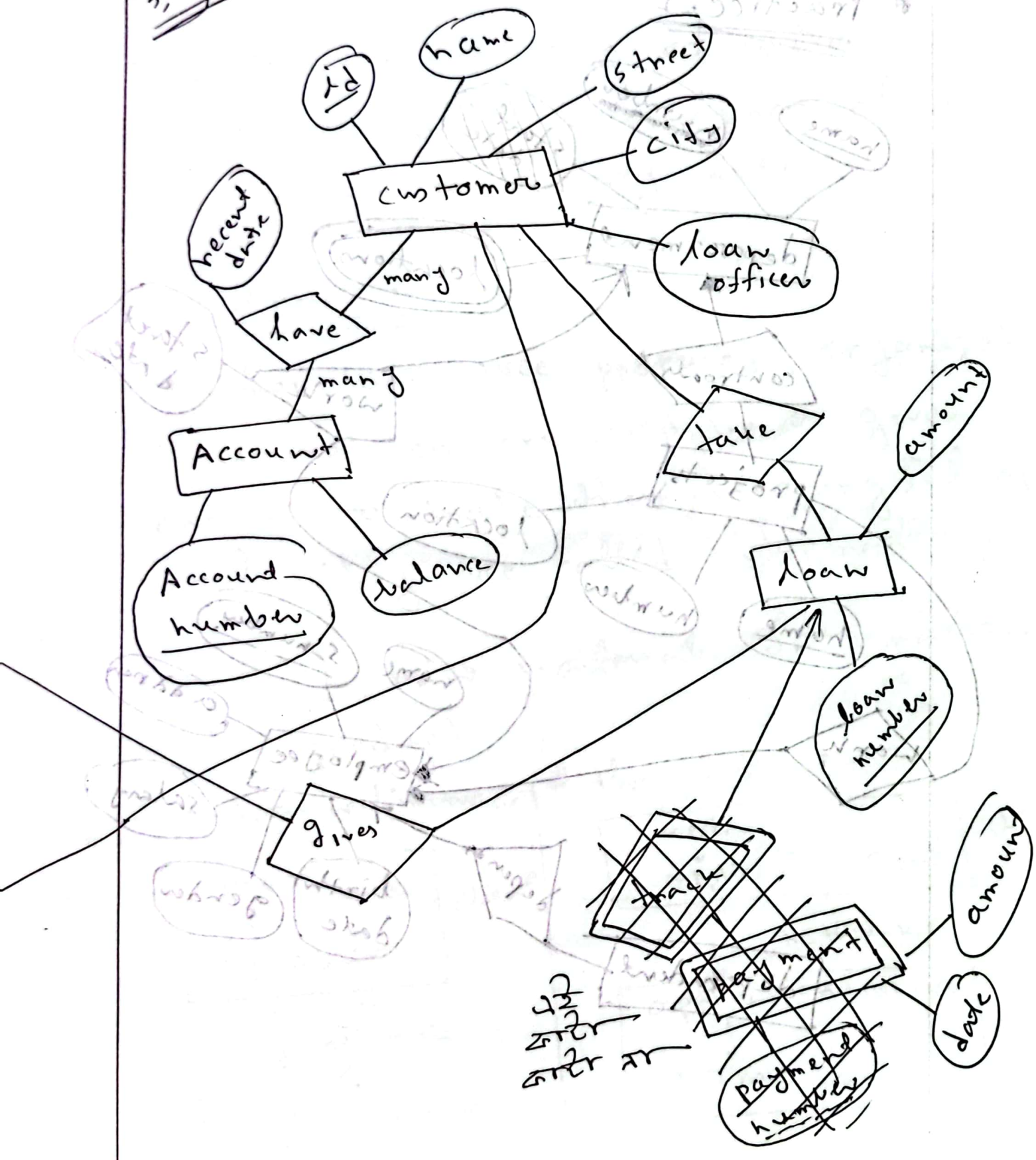
Practice]



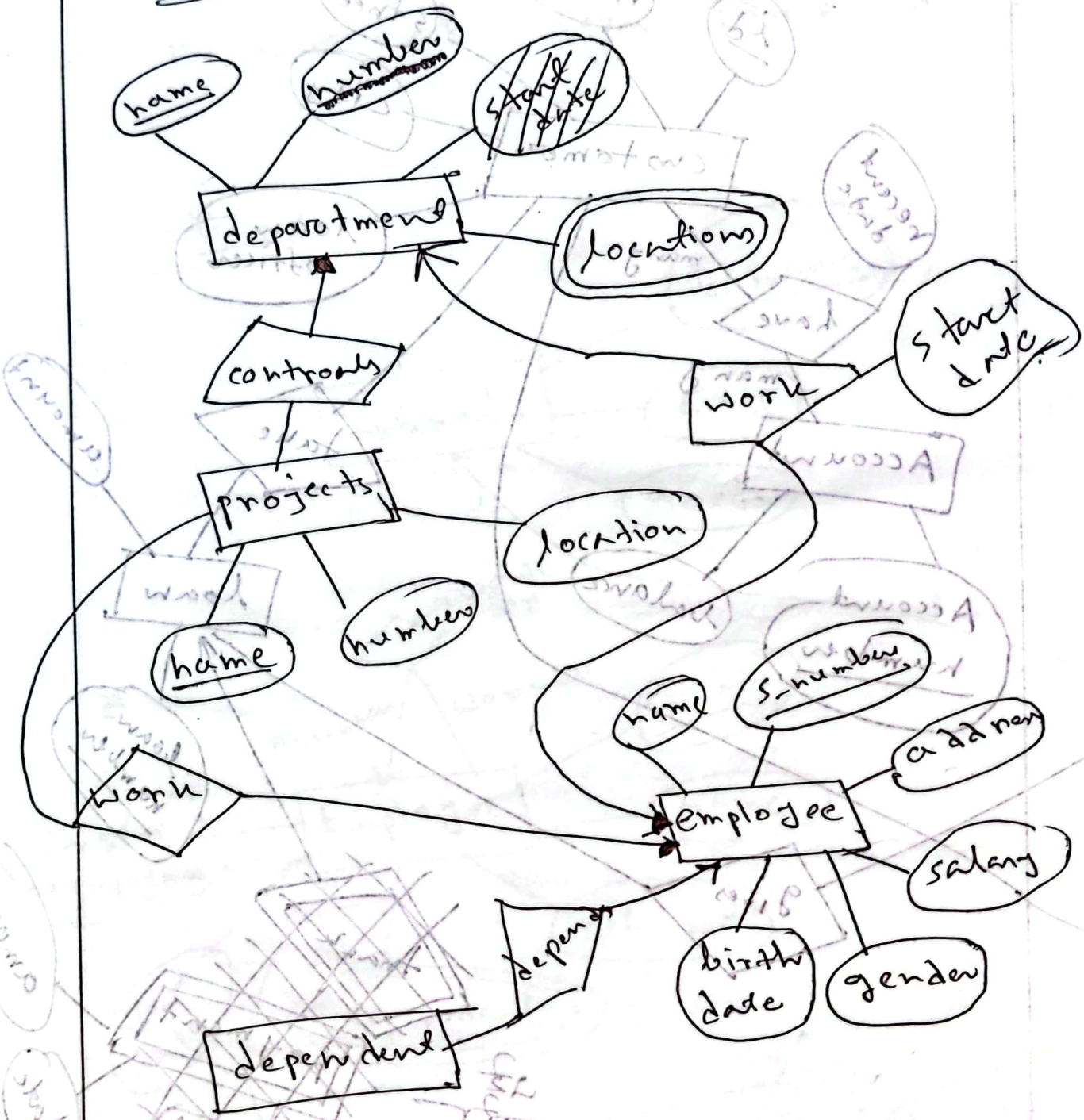
C.W
6.2.24

DBMS

3, 4, 5



* Practice-2


name
number
start date
department
locations
controls
work
start date
projects
location
name
number
work
name
s_number
address
employee
salary
depen
birth date
gender
dependent

Theory

Data

→ Data is a collection of values.

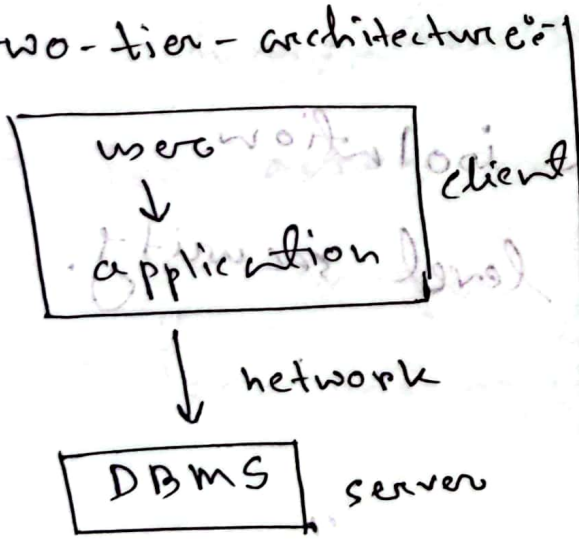
→ Computer data is a bunch of 1's & 0's, known as binary data.

Information

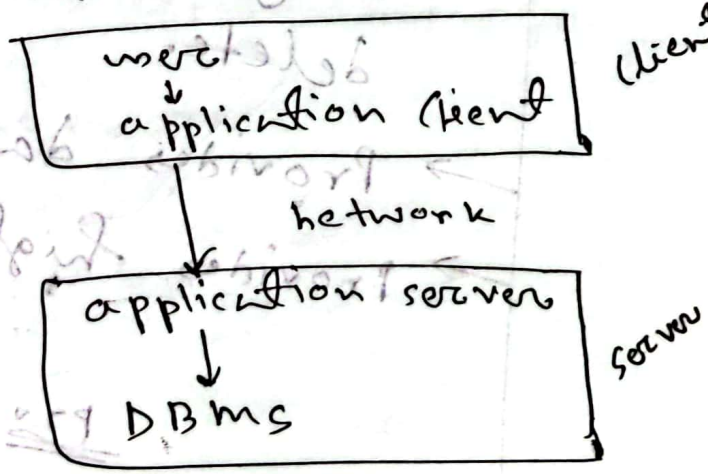
→ When data are processed, organized, structured or presented in a given context to make them meaningful or useful, they are called information.

* Data is raw material, Information is the product.

two-tier-architecture:-



three-tier:-



Database

→ organized collection of interrelated data.

DBMS

→ software that interact with end users.

→ control the storage.

→ retrieval of data.

DBMS key features:

→ Minimise data redundancy (repetation)

→ Provides data consistency.

→ Easy to write, update, search, delete.

→ Provides data isolation

→ Provides high level security.

P.T.O

Query processor

DDL (Data Definition Language)

↓

data type, data dictionary.

(int, varchar(5))

DML (Data Manipulation Language)

→ query evaluation

→ query optimization

Query Evaluation engine

→ Execute low-level instruction.

Storage manager

→ Authorization & Integrity manager

→ Transaction manager

→ File manager

→ Buffer manager.

* Entity : Real world object.

* Attributes : descriptive properties.

→ simple : values can't be divided

→ composite : " can " "

→ Derived : values can be calculated from other attribute

→ Multivalued : more than one value.

→ Descriptive : can't be associated with any entity set.

* keys

superkey : set of one or more attributes (unique)

Candidate key : minimal superkey, no proper subset.

discriminator

Primary key: chosen candidate key.

Partial key: provides identification of weak entity.

composite key: key with more than one attribute.

Foreign key: the one table key which refer the primary key of another table.

* **Relationship sets**

Recursive: the same entity set participate more than once in relationship.

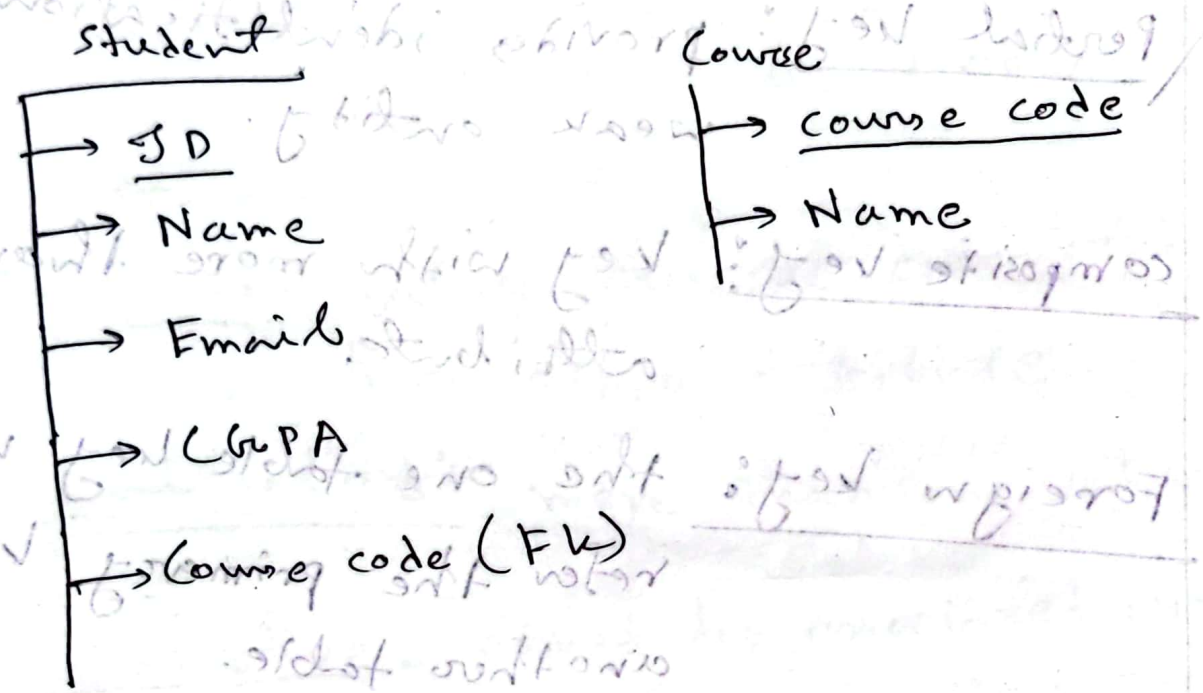
Parallel: more than one relationship between same entity sets.

CW
12.2.24

DBMS

Schema + theory → C.T

(foreign key)



* one to one
(যেকোনো এক পাশে foreign key)

* one to many
(many তে foreign key যাবে)

* [one এর side এ Arrow হবে]

*  → আলাদা table বানাবে।
main table এর Primary key
multivalued attribute table এর
foreign key হবে। (আর যার সঙ্গে
হবে Primary key. foreign key এর
সাথে মিল হবে, আর যদি root এর
আর value না হবে তাহলে main
value table এ বসবে।

* Derived attribute table এ আসবে
না।

* many to many table এর extra
আর একটা table হবে। table গুলোর
primary key গুলো foreign key
এর সাথে মিল হবে।

G.W
17.2.24

DDL

CREATE:

* Database : Create Database UIV

~ " IF NOT EXIST

* Delete:

DROP Database UIV;

Data Type

→ INT, INT(30), INT(UNSIGNED),

→ DOUBLE, DOUBLE(10,3),
DOUBLE(UNSIGNED)

→ DATE
↓
"YYYY-MM-DD"

→ TIME ("HH:MM:SS")

→ DATETIME
"YYYY-MM-DD hh:MM:ss"

→ CHAR, CHAR(11)

→ VARCHAR(11)

•

Constraints

(1) Primary key

(2) foreign key

C.W
25.2.24

* Create table student(

id varchar(11),
name varchar(30),
mobile char(11),
email varchar(30),
cgpa double(3,2)

);

* primary key (id).
* constraint fk1 (constraint name) foreign key (course_code) references course (course_code)

course → tablename

Alter table

* Alter table student Add column course_code varchar(30);
* Alter table student Drop column course_code;
* Alter table student add constraint pk1 primary key (id);

* Alter table student drop primary key;

* Alter table student add constraint fk1 foreign key (col_name) references ref_table (ref_col_name)

* Alter table drop foreign key fk1

That one foreign key

Unique

* Alter table student add unique (id);

* Alter table student unique removes

Drop index id;

* Alter table student_info
rename student_info;

* Alter table student_info
Change column id s_id varchar(11)
old name
new name

Insert

* Insert into student_info
values ('01', 'Ibrahim', '01879222610',
'Ibrahim@gmail.com', 3.24);

* update student_info set
name = 'Ibrahim'
where s_id = '01';

* Delete from student_info where sid = 6;

Select

* select * s_id, name from student_info;

* select * from student_info;

* select * from student_info where sid %2 != 0;

* select * from student_info order by cgpa desc;

colname

* select * from student_info order by cgpa desc Limit 2;

last theke

* Limit 3, 1 → Exact oita

* select distinct name
from student_info
where
order by
Limit

* &&, AND

* select * from student_info

where | s_id | between | 02 | and | 05

* where s_id in (02, 05);

* %. operator

* where
  like

* where name like '%am%';
  (substring)

* where name like 'x%'
  ↳ starts with x

* %x → ends with x

* '___' → 3 characters

* 'A_B%m' → 3 ... & ends with m

* where ABS (salary)

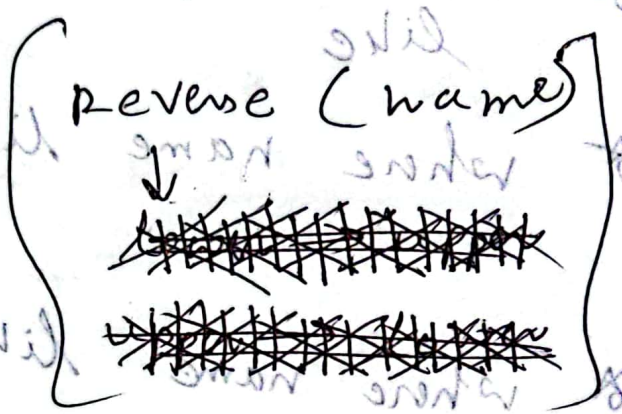
* select Round (1.35, 1)

  Truncat (1.34, 2)

* select name, lower(name)

upper(name)

reverse(name)



* where name = reverse(name)

palindrome

* concat(s_id, ' ', cgpa)

↳ একত্র করে,

* substr(name, 3, 5)

* select name, LPAD(name, 10, '*')
from student_info

*****abc

* RPAD(name, 10, '*')

abc*******

* Left(name, 3)
↳ left থেকে 3 ঘর
নেওয়াবে

Amman → Amm

* Right(name, 3)
↳ Amman → man

C.W
27.2.24

DBMS

Join

Student

ID

Name

CGPA

course_code (FK)

Course

course code

course name

```
Select name, course_code.
from student
where id-1. 2 = 0
```

Inner Join

```
select s.name, s.course_code,
c.course_name From student AS s
Inner Join
course AS c
ON s.course_code = c.course_code
```

left join

* select * from student as s

LEFT JOIN course as c

ON s.course_code = c.course_code

* Same for right join

* select
  from join
  where
  order
  like

* Cross join

  select * from Student

  cross join Course

* Natural join

select * from student

natural join course

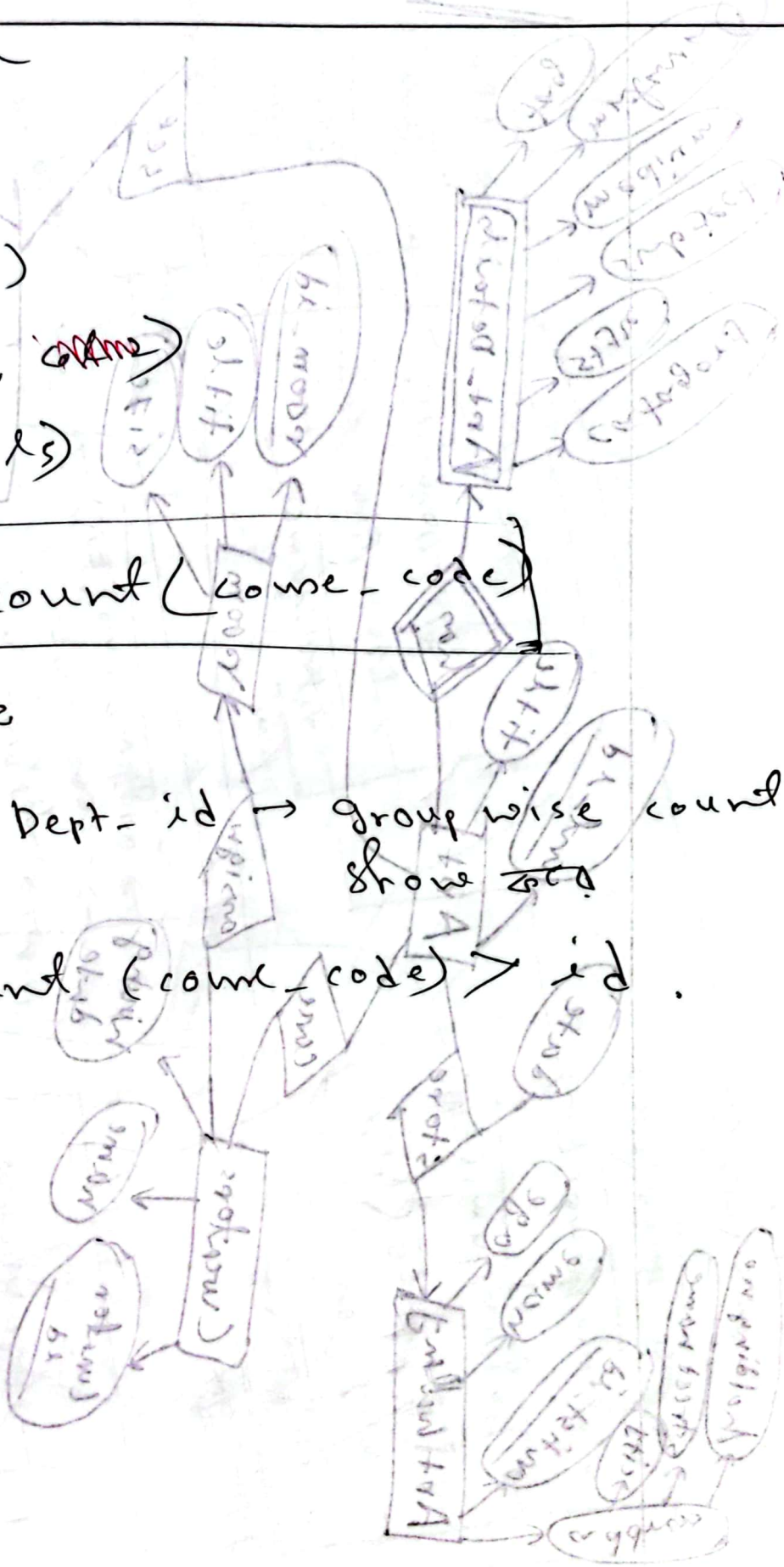
* self Join

select * from table name

Join table_name
ON (~~ = ~~)
where condition
order by
like
limit

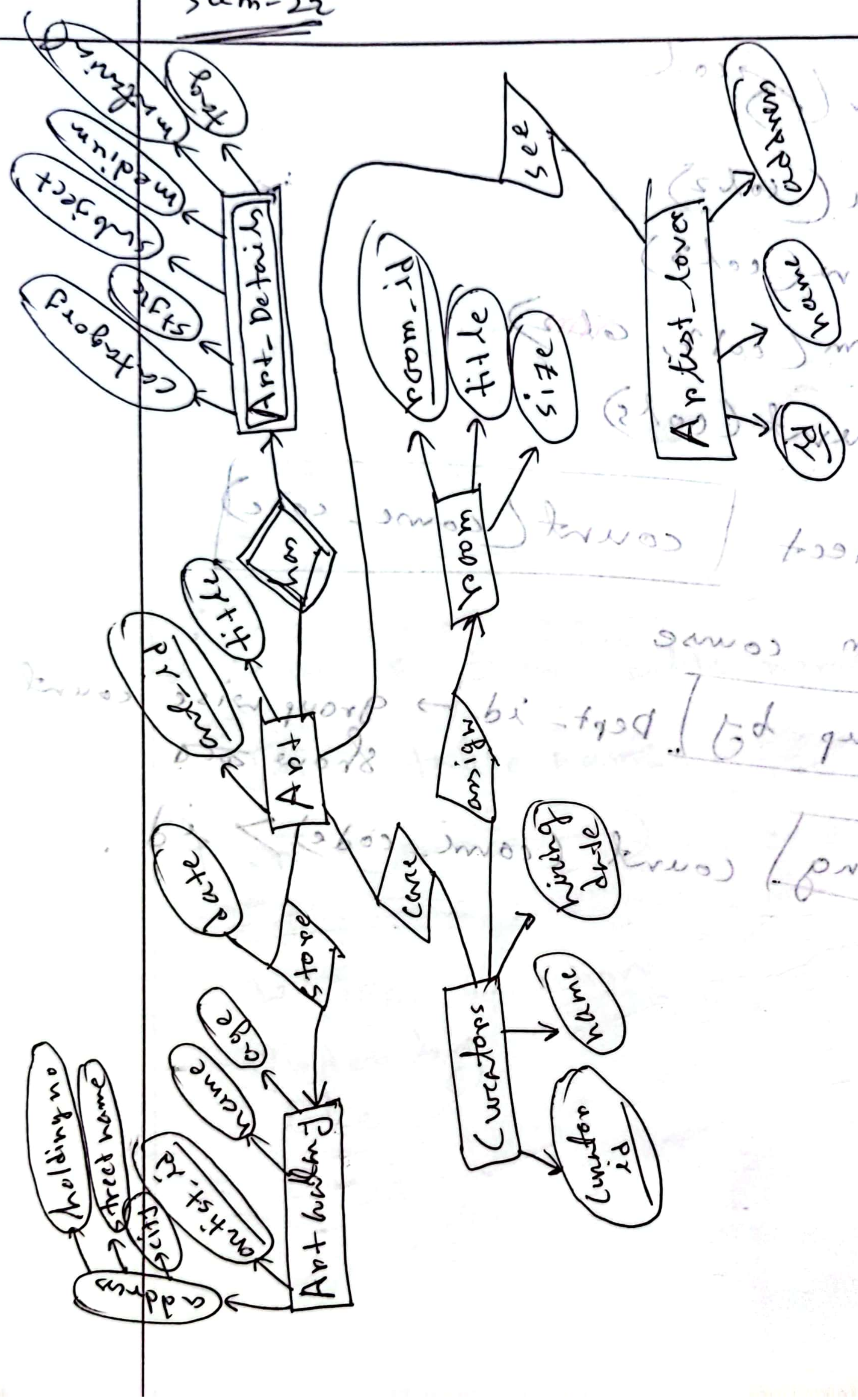
* Avg(col)
* Max(col2)
* Min(col3)
* sum(col1, col2)
* count(col5)

* select count(course_code)
from course
group by Dept_id → group wise count show
Having count(course_code) > id .

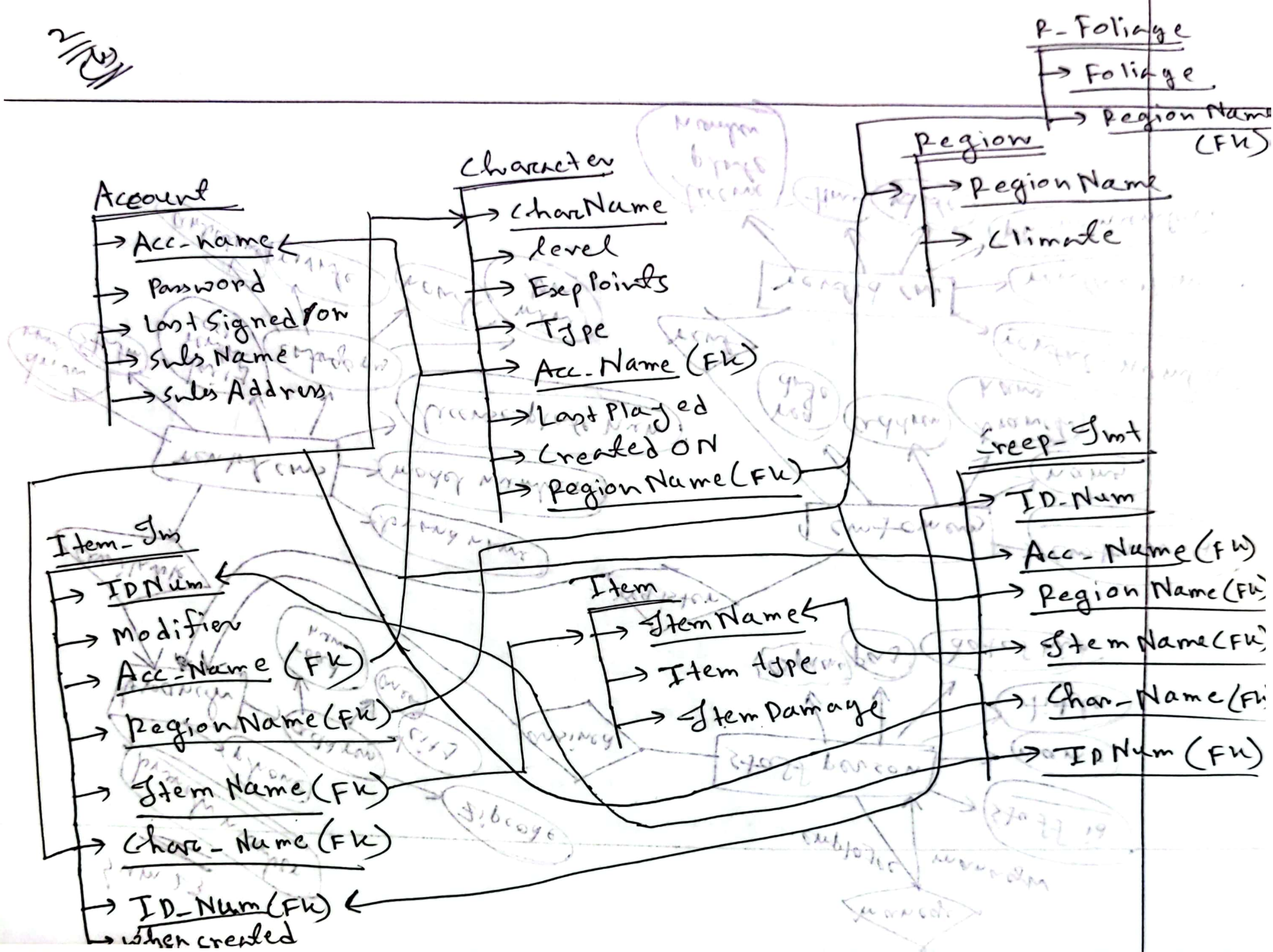


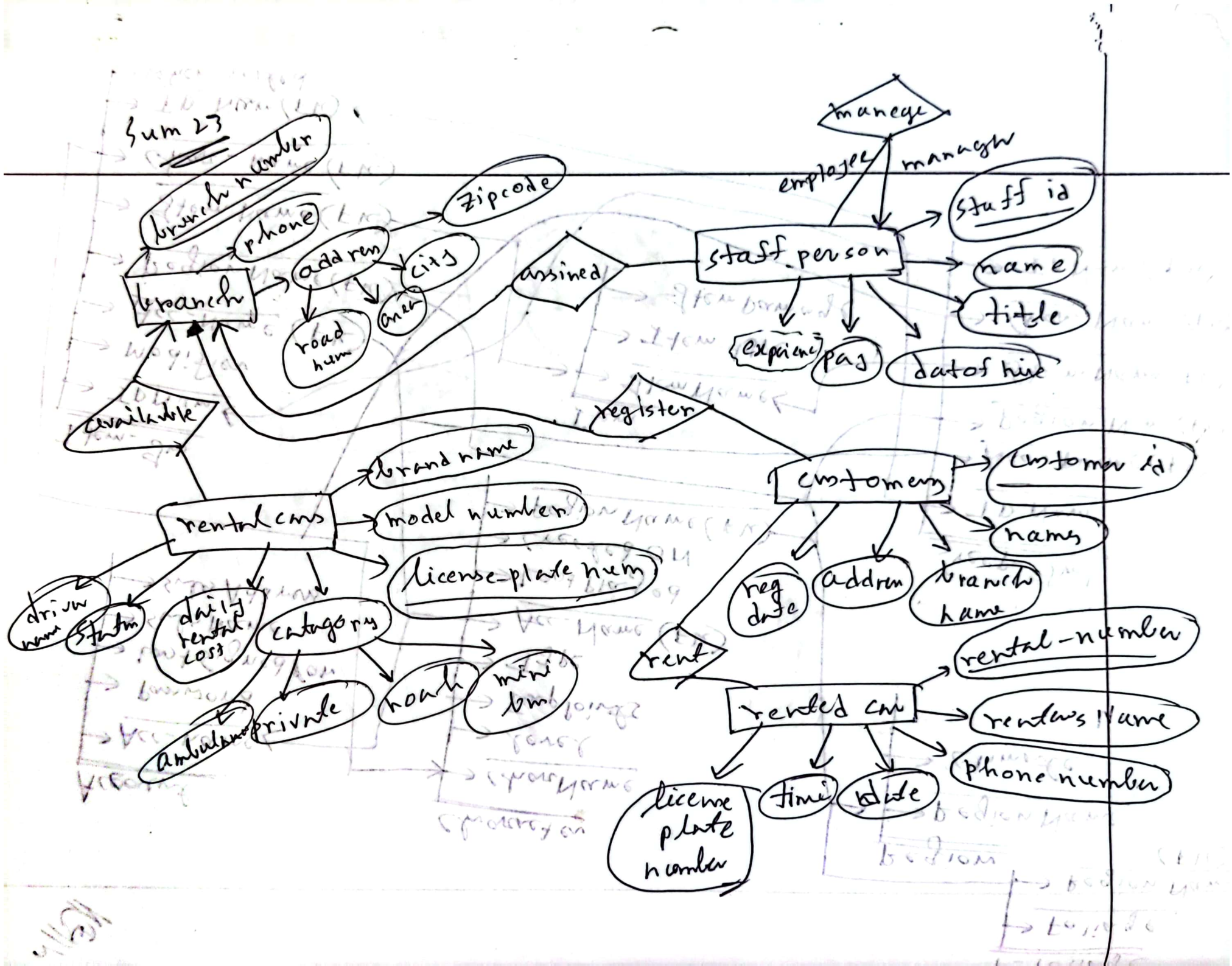
sum-22

DBMS

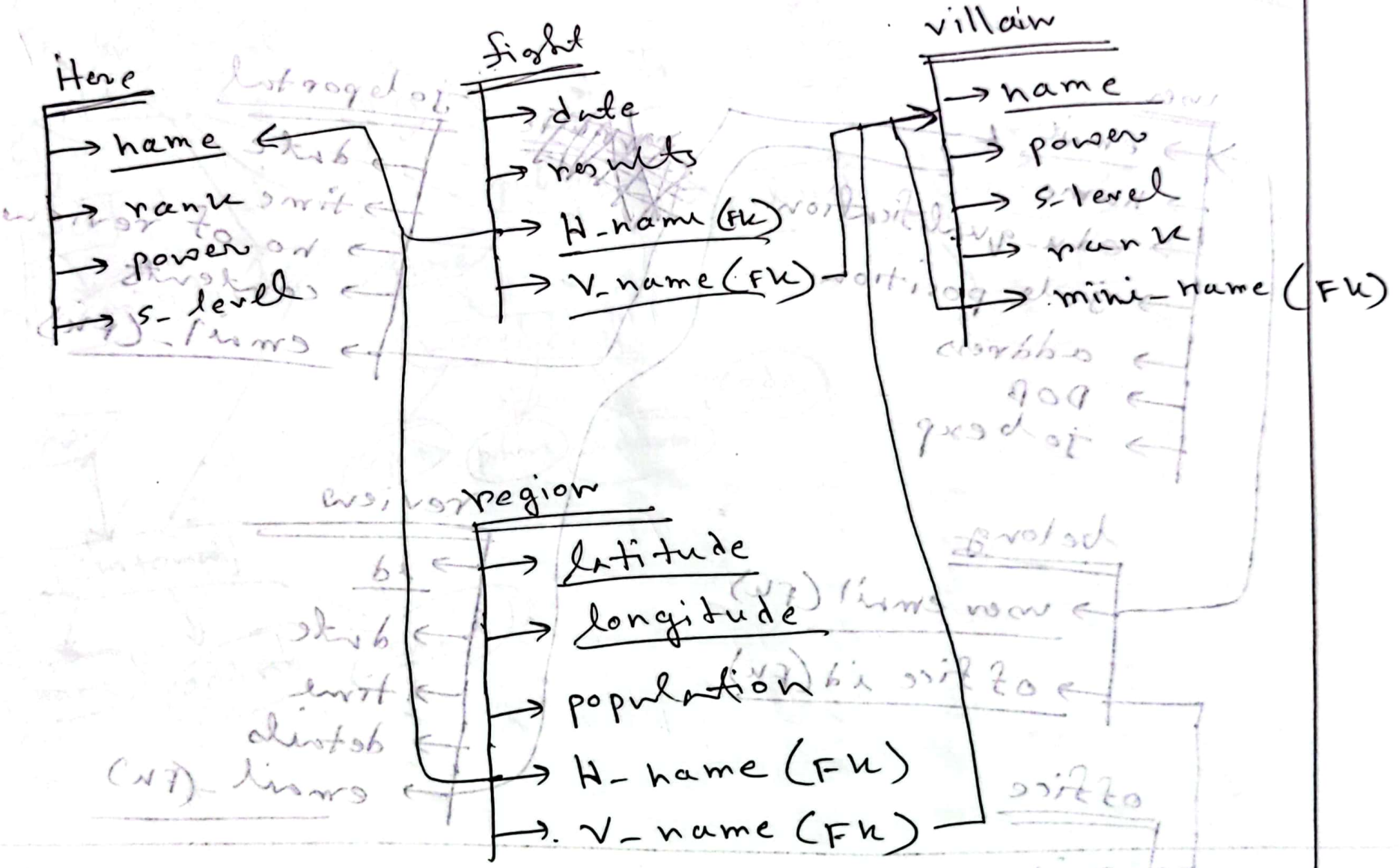


Art_Details
text
medium
subject
category
Art
title
art_id
room
room_id
title
size
see
Artist_lover
address
name
id
assign
care
Curators
hiring date
name
Curator id
store
date
Art building
age
name
artist_id
city
holding no
street name
address

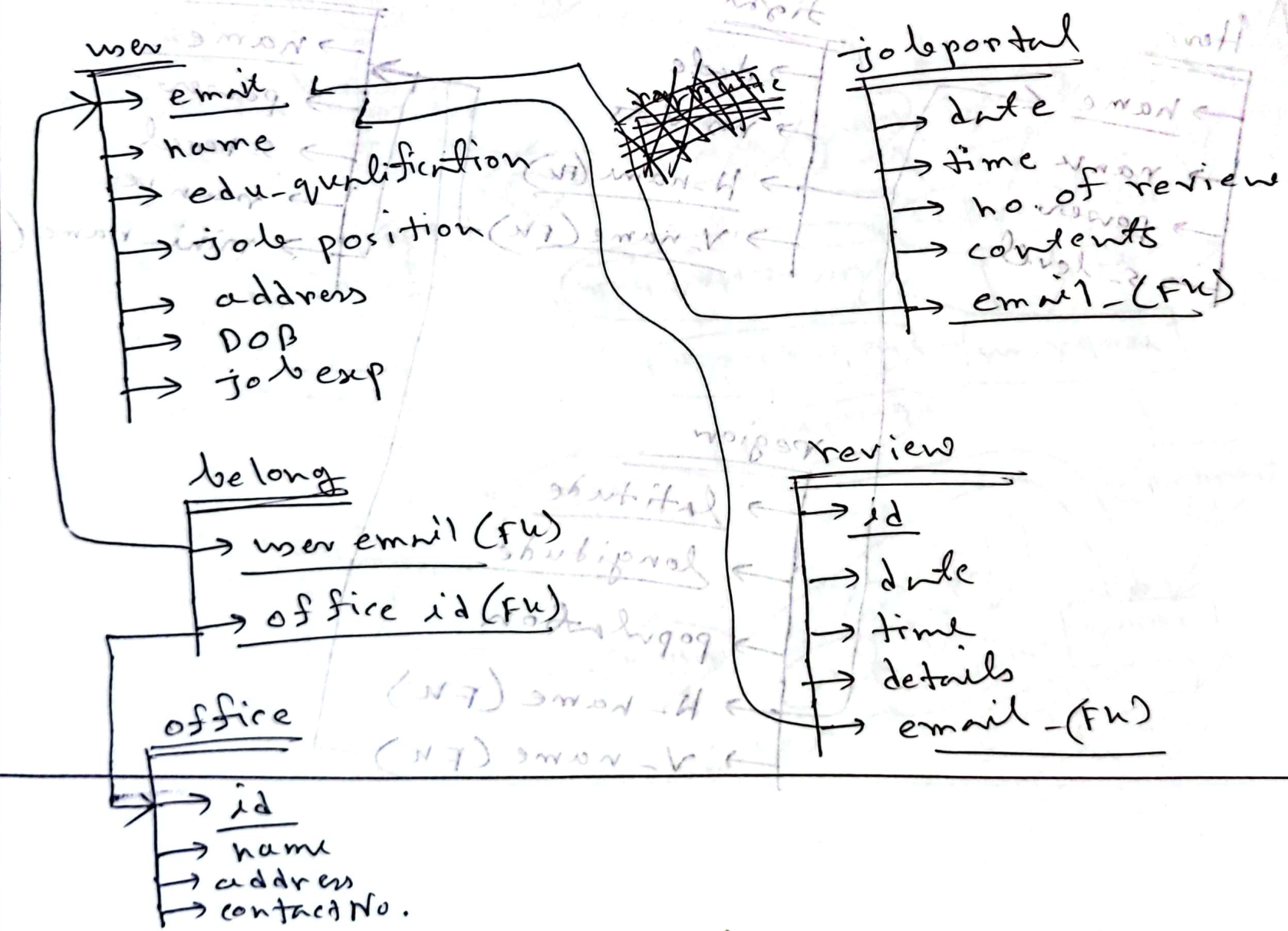

R_Foliage
Foliage
Region Name (FK)
Region
Region Name
Climate
Character
CharName
Level
Exp Points
Type
Acc_Name (FK)
Last Played
Created ON
Region Name (FK)
Account
Acc_name
Password
Last Signed on
Subs Name
Subs Address
Item_Inv
ID Num
Modifier
Acc_Name (FK)
Region Name (FK)
Item Name (FK)
Char_Name (FK)
ID_Num (FK)
When created
Item
Item Name
Item type
Item Damage
Creep_Inv
ID_Num
Acc_Name (FK)
Region Name (FK)
Item Name (FK)
Char_Name (FK)
ID Num (FK)

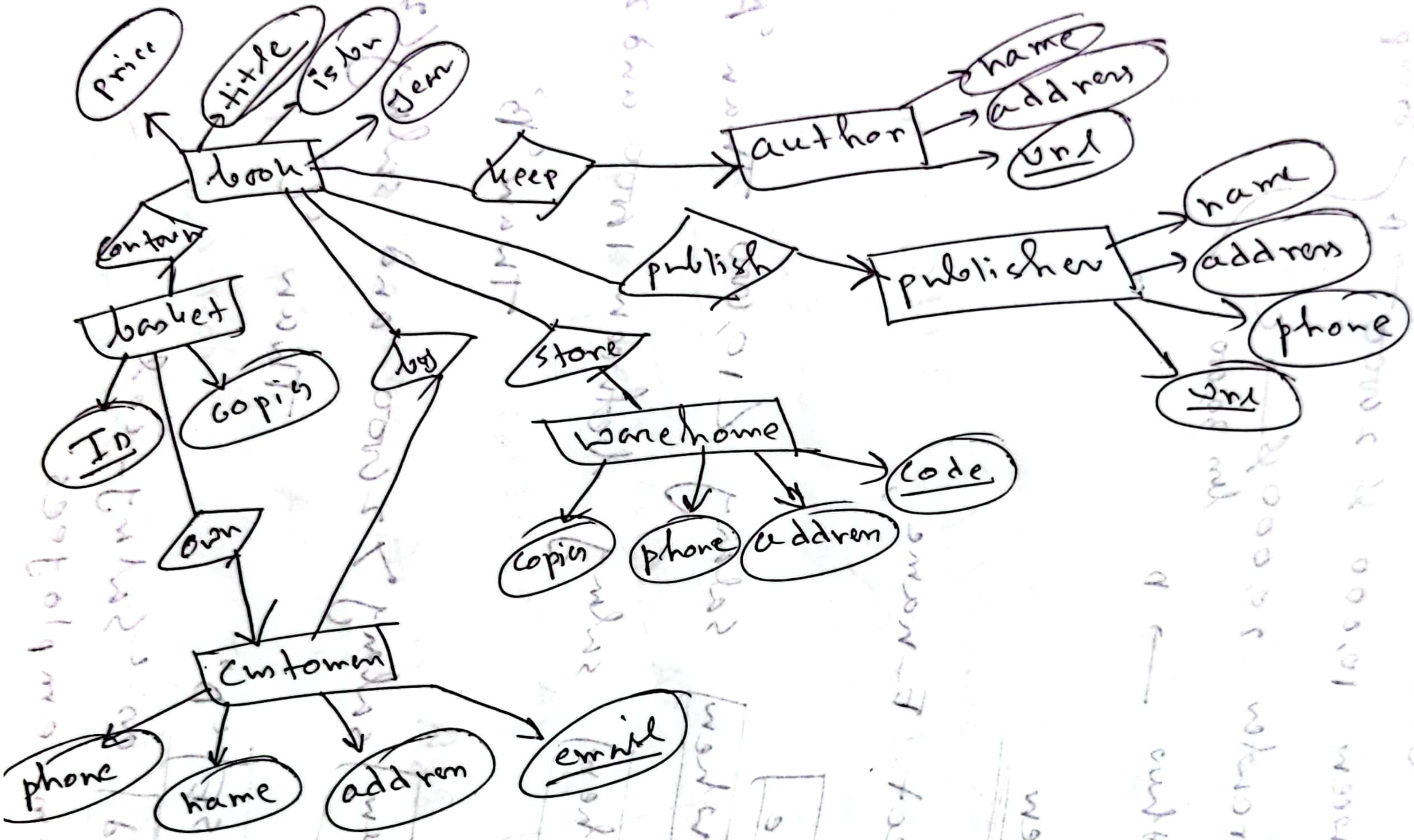

Sum 23
branch number
phone
address
city
zipcode
road num
branch
assined
manage
employee
manages
staff_person
staff id
name
title
expriene
pay
datof hire
available
register
brand name
model number
rental cars
license_plate num
drivr name
status
daily rental cost
category
ambulance
privnte
mini bus
customers
customer id
names
reg date
address
branch name
rent
rental-number
rented car
renters name
phone number
license plate number
time
date

Practice-5


Hero
name
rank
power
s-level
fight
date
results
H_name (FK)
V_name (FK)
villain
name
power
s-level
rank
mini_name (FK)
region
latitude
longitude
population
H_name (FK)
V_name (FK)

7


user
email
name
edu-qualification
job position
address
DOB
job exp
jobportal
date
time
no. of review
contents
email_(FK)
belong
user email (FK)
office id (FK)
review
id
date
time
details
email_(FK)
office
id
name
address
contact No.



price
title
isbn
Jern
book
keep
author
name
address
Url
contain
basket
copies
ID
own
buy
publish
publisher
name
address
phone
Url
store
warehouse
code
copies
phone
address
Customer
phone
name
address
email

C.W
2.3.24

Salary less 10,000 → A
Between 10,000 & 20000 → B
greater 20,000 & less 30,000 → C
other → D

group

when

Query

```
select E_name,
Case
  when salary < 10,000 Than 'A'
  when salary between 1000 and 2,000
                        Than 'B'
  when salary > 2000 and salary < 3,000
                        Than 'C'
  ELSE 'D'
End as 'Salary_cat'
From employees;
```

E-NAME

A (Name)

B "

C "

Salary col

A

B

C

* DATEDIFF (difference between two date)

subquery

SELECT * | Col1, Col2 ... As '' |

Col1 + Col2 As '' | case when |

From table_name As '', Agg

repeat { JOIN table 2 As ''
ON condition
Where condition (row filter) }

group by (col_name;
Having condition (# grp filter)
Order by colname ASC/DESC
Limit 2, 5

Subquery

* salary > E-id 20 salary

G.W
5.3.24

DBMS

Sub Query multicolumn

ALL/ANY

Select name from employees
where salary > All/ANY (Select salary From employees where d_id = 50)

Relational algebra

→ Project Π (select)
→ select σ (condition)
→ Natural join ($\bowtie$)
→ Cross join X
→ Conditional operator ($=, >, <, \neq, \leq, \geq$)
→ logical operator ($\wedge, \vee, \neg$)
and or not

P.T.O

* select name, id from Employees

⇒ $\Pi_{(names, id)}$ (Employees)

* where id = 50;

⇒ $\Pi_{(name\ id)}$ ($\sigma_{id=50}$ (Employees))

* where id = 50 && d_name = 'CSE';

⇒ $\Pi_{(name\ id)}$ ($\sigma_{id=50 \wedge d\_name = 'CSE'}$ (Employees))

* Natural join

$\Pi_{(\ldots)}$ (Dept $\bowtie$ course)

σ ← condition

* DML → Data files, which store the database itself also the data are real.

* DDL → Data dictonary, which stores metadata about the structure of the database, in particular the schema of the database.

* Types of Data models:

High level conceptual data model

→ E-R model

→ Object oriented model.

* Record based logical data models:

→ Relational model - collection of tables.

→ Hierchical model → n, n trees.

→ Network model → n records & links (graph)

* Foreign key are used to ensure reference / referential integrity → the condition in which every foreign key entry must either be null or a valid value.

* Associative Entity set:

used in many to many relation.

C.W

DBMS

1. Normalization *
2. Transaction
3. Hashing
4. B^+ tree *

} final topic

C.W 23.3.24

DBMS

Normalization

* functional dependency

course → RoomNo,

course — Independent

Time — dependent

SID → course

SID	course	room	time
1	3522	408	10
2	3522	726	11
3	3522	408	10
4	3522	408	10
5	3522	726	11

SID	course
1	3522
3	3522
4	3522
5	3522

course	room	time
3522	408	10
3522	726	11

Axioms / Properties

Armstrong's Axioms → 3

(i) Reflexivity rule

$\beta \subseteq \alpha$, $\alpha \rightarrow \beta$ (holds)

$AB \rightarrow B$, (valid)

[B is the subset of AB]

(ii) Augmentation

$\alpha \rightarrow \beta$

$AB \rightarrow CD$ [If $AB \rightarrow CD$ holds]

[valid]

(iii) Transitivity

$A \rightarrow B$, $B \rightarrow C$

$A \rightarrow C$ [valid]

(ii) Union rule

$A \to B$, $A \to C$

$A \to BC$ (valid)

(iii) Decomposition rule

$A \to BC$

$A \to B$, $A \to C$ (valid)

(iii) Pseudo — transitivity rule

$A \to B$, $CB \to D$ (holds)

$CA \to D$

* FD → (Functional Dependency)

F^+ → FD (All possible combination)

↓

closer

C.W
30.3.24

DBMS

Attribute closure

$R = (A, B, C, G, H, I)$

$F = \{A \to B,\ A \to C,\ CG \to H,\ CG \to I,\ B \to H\}$

$(AG)^+ = AG$ [Reflexivity]

$= ABG \Rightarrow A \to B$

$= ABCG \Rightarrow A \to C$

$= ABCGH \Rightarrow CG \to H$

$= ABCGHI \Rightarrow CG \to I$

$(AB)^+ = AB \Rightarrow AB$

$= ACB \Rightarrow A \to C$

$= ACBH \Rightarrow B \to H$

* Functional Dependency
* super key
* valid / invalid

* $R = (A, B, C, D, E)$

$F = \{B \to AC,\ C \to AB,\ AB \to D,\ BD \to A,\ AD \to C,\ EA \to D\}$

$A^+ = A$

$B^+ = BACD$

$C^+ = CABD$

$D^+ = D$

$E^+ = ED$

$AB^+ = AB \to ABCD$ [super key হবে]

$AB^+ \neq ABCDE$

$BE^+ = ABCDE$ [Super key]

$AC \to E$ (possible কিনা)

$(AC)^+$ বের করতে হবে

(AC^+ এর মধ্যে E থাকলে Valid)

[Candidate key]

$R = (A, B, C, D, E)$, (B, C)

$F = \{A \rightarrow C, CD \rightarrow B\}$

α = [যেই যেই যেইটার FA থেকে Right element গুলো R থেকে বাদ যাবে]

$\alpha = \{A, D, E\}$

β = [α যেগুলো থেকে যেইটার বাদ দিয়ে, যেইটা F এর বামে থাকলে নিব না] [দুই side এই থাকলে বাদ]

$\beta = \{B\} \rightarrow X$ [কে হবে না]

$\alpha = (ADE^+) = ABCDE$

[কমা বাদে এর ভেতর]

** $R = (A, B, C, D, E)$ (A, B, C, D)

$F = \{A \to C,\ C \to BD,\ D \to A\}$

$E^+ = E$ ✗

$EA^+ = ABCDE = SK = CK$

$EB^+ = X$

$EC^+ = ABCDE = SK = CK$

$ED^+ = ABCDE = SK = CK$

** $R = \{P, Q, R, S, T, U\}$

$F = \{PQ \to RTV,\ PR \to S,\ V \to P,\ R \to S,\ ST \to PV\}$

$\alpha = \{Q\}$

$\beta = \{\}$

$Q^+ = Q$ ✗

$Qp^+ = QPRTVS = SK = CK$

$QR^+ = QRS$ ✗

$QS^+ = QS$ ✗

$QT^+ = QT$ ✗

$QV^+ = QVPRTS = SK = CK$

~~QPR~~$^+$

$QR^+ = QR$ X

$QRS^+ = QRS$ X

$QRT^+ = QRTSPU = SK = CK$

$QSR^+ =$ X

$QST^+ = QSTPUR = SK = CK$

$QTR^+ =$ X

$QTS =$ X

Minimal Cover

$R = \{\}$

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

$R = \{ A, B, C, D, E, F \}$

$F = \{ ABC \to CDEF,\ C \to E,\ A \to B,\ D \to F \}$

(i), (ii) ~~$ABC \to C$~~

$ABC \to D$

$ABC \to E$

$ABC \to F$

$C \to E$

$A \to B$

$D \to F$

① Decompose Right side single Attribute

② Reflexivity Attr.

$ABC \to C$ ✗

③ Extreniom Attribute

~~Attr~~

(ii) $A \to B$

$AC \to B$ ~~Attr~~.

(ii) (iii)

$A \to B$; $AB \to C$

(iii)

$ABC \to D$

$ABC \to E$

$ABC \to F$

(iii) (ii)

$C \to E$

$A \to B$

$D \to F$

$A \to D$ $A^+ = AB$ | $C^+ = CE$

$A \to F$ $A^+ X$ $C^+ = X$

$C \to E$

$A \to B$

$D \to F$

যদি A এর জন্য C তে D পাই তবে A বাদ।

(iv) $(AC)^+$ But D consider করবে না।

$(AC)^+ = ACBEF$ [D পাই নাই]

So fine হবে

AC^+ and F তাহলে super key হবে।

$AC^+ = ACDEBF$

$AC \rightarrow F$ (super key)

CK (short cut)

2NF

① Multivalued Attribute থাকবে না।

② Duplicate column ✗

③ ~~row~~ ✗

④ rowOrder maintain ✗

⑤ Column Order ~ ✗

⑥ Datatype same থাকবে।

Normalization

1 NF

✓2 NF

3 NF

B CNF / 3.5 NF

P.T.O

2NF

(i) 1NF হতে হবে।

(ii) No partial dependency.

⇒ Proper subset of CK →

Now prime Attribute

R = (A, B, C, D, E)

ABC = {AB, AC, BC, A, B, C}

এর কম কিছু যা সবার element হয়

Now prime Attribute

ABC যা কিছু তা হবে।

P.T.O

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

F = { A → B, B → C, C → D, D → E }

(ABCDEF)⁺ = ABCDEF (CK)

(AF)⁺ = ABCDEF ✓ (CK)

A⁺ = ABCDE ✗

F⁺ = F ✗

prime attribute = (A, F)

= (B, C, D, A, E)

Nown

C.W
16.4.24

N.P.A = Non Prime Attribute
P.A = " " DBMS

3NF

→ 2NF হতে হবে
→ No transitive dependency

False (N.P.A → N.P.A) L.H.S R.H.S

যদি R.H.S non-prime, A → B

→ L.H.S = Super key

or

P.H.S = P.A

Ex

R(A, B, C, D)

F = {A → B, B → C, C → D}

ABCD+ = {ABCD}

A+ = {ABCD} CK = {A} = SK

PA = {A}

No 3NF

$R = (A, B, C, D, E)$

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

$ABCDE^+ = \{A\ B\ C\ D\ E\}$

$AE^+ = \{\ \}$

DE^+

CE^+

BE

$Sk \to Ck = \{AE, DE, CE, BE\}$

$P.A = \{A, B, C, D, E\}$

P.T.O

* BCNF / 3.5 NF

→ 3 NF

→ L.H.S → (Must be SK)

R (A, B, C)

F = { A → B, B → C, C → A }

$ABC^+ = \{ABC\}$

$A^+ = \{ABC\}$

A

↗

C

↗

B

SK = CK = { A, B, C }

* R = (A, B, C, D, E)

F = { A → C, B → D, AC → D, CD → E, E → A }

AB$\cancel{CDE}^{+}$ = { A, B, C, D, E }

AB+ = { A, B, C, D, E }

{ AB, EB, AD, ED }

AB → AD

↑

EB

↑

ED

P.A = { A, B, D, E }

NPA = { C }

BCNF X

3NF X

2NF X

2NF ✓

BCNF

$R = \{ A, B, C \}$

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

~~Decomp~~

$AB^{+} = \{ ABC \}$

$A^{+} = \{ ABC \}$

$B \to C$ (violet)

$R_1 = \{ B, C \}$

$F_1 = \{ B \to C \}$

$Sk = Ck$

$= \{ B \}$

*

$R = \{A, B, C, D, E\}$

$F = \{A \rightarrow B, \; BC \rightarrow D\}$

$(AB)^+ = \{A\,B\,C\,D\,E\}$

AC E

$Sk = CK = \{AC\,E\}$

R

$R_1(A, B)$

$F_1 = \{A \rightarrow B\}$

$Sk = CK = \{A\}$

$R_2(A, C, D, E)$

$F_2 = \{BC \rightarrow D\}$

$AC \rightarrow D$

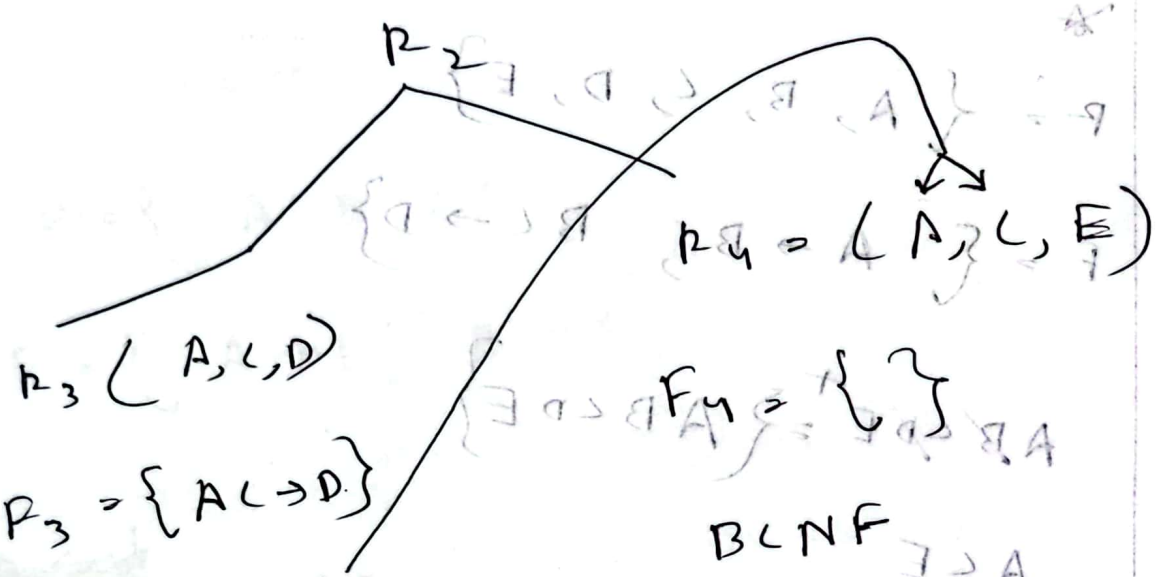
$Sk = CK = ACE$

BCNF

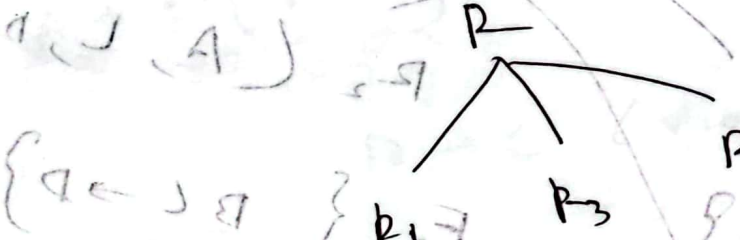
$\neq \{A \rightarrow B$ BCNF $X\}$.

(Ans)

P.T.O



BCNF



$F_1 \cup F_2 \cup F_3 = F$ (Preserve the Dependency)

$= \{A \to B, AC \to D\} \neq F$

(Not)

→ Decomposition

Lossless (3 or 2nd)

→ $R_1 \cup R_3 \cup R_4 = R$

→ R_1, R_3, R_4 → common attribute

→ common attribute → candidate key

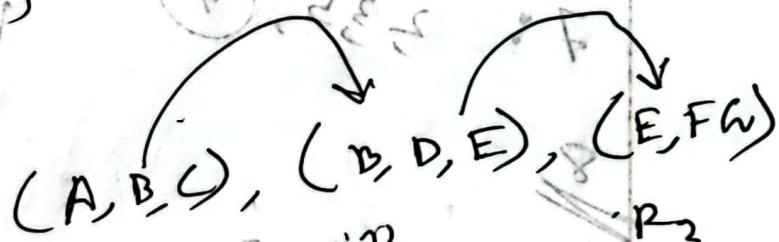
~~#~~

→ {(A, B) ∪ (A, C, D) ∪ (A, C, E)} ✓

= {A, B, C, D, E}

→ {A, C}

→ { }



J.W

20.11.24

B+ tree

Insert

1. leaf node (copy up)
2. Non-leaf / Root node (push up)

Insert

7: 2, 3, 5, (7)

8

Middle: 3

Left child: 1, 2

Right n: 3, 5, 8 (copy up)

up: 3

Leaf

1, 2, 3, 5, 8

Root

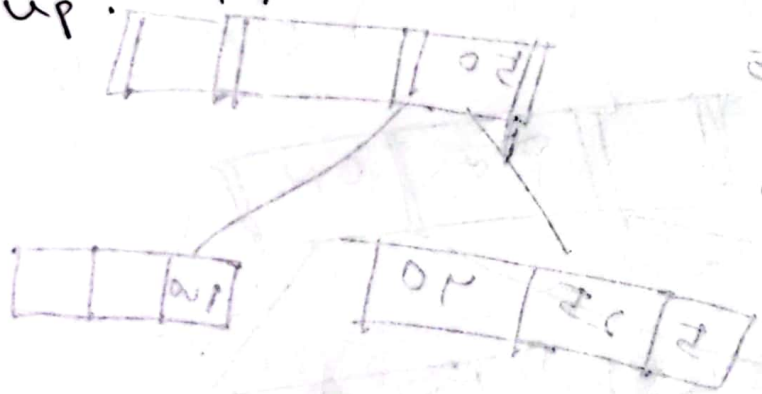
3, 13, 17, 24, 30

Middle: 17

left: 3, 13

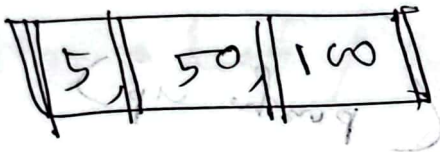
Right: 24, 30 (push up)

up: 17



5, 50, 100, 25, 40, 45, 150,

80, 30, 15, 35



25

5, 50, 100, 25

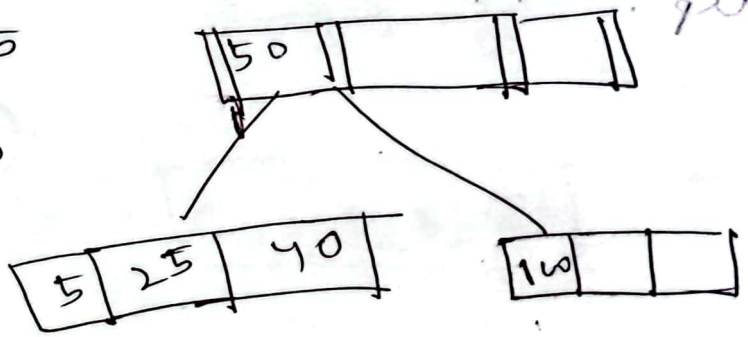
5, 25, 50, 100

M = 50

L = 5, 25

R = 100

up = 50



40

45

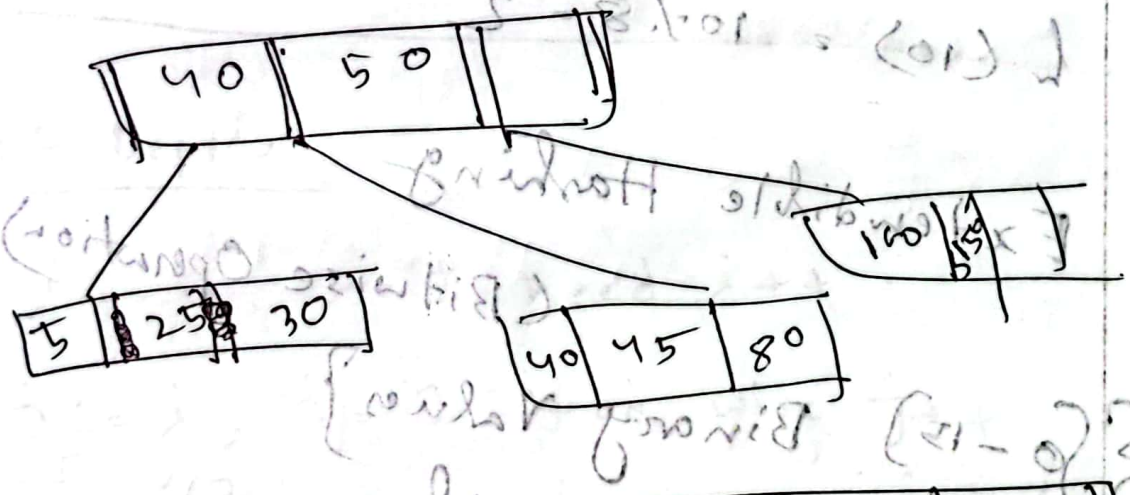
5, 25, 40, 45

M = 40

L = 5, 25

R = 40, 45

U = 40



15

5, 25, 30, 15

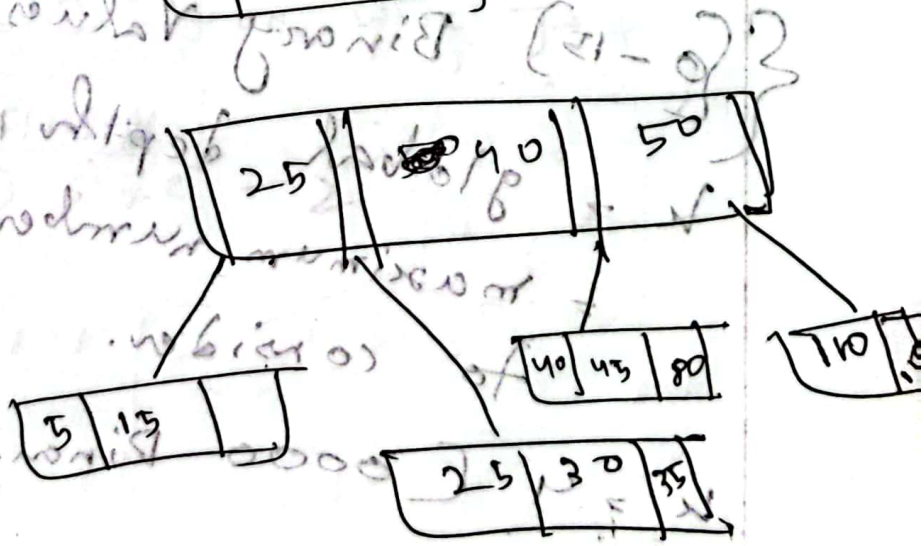
5, 15, 25, 30

M = 25

L = 5, 15

R = 25, 30

U = 25



* dense indexing (pointing)
* sparse indexing (no pointing)

Dynamic Hashing

Hash function

$h(k) = k \% 8$

$h(100) = 100 \% 8 = 4$

$h(10) = 10 \% 8 = 2$

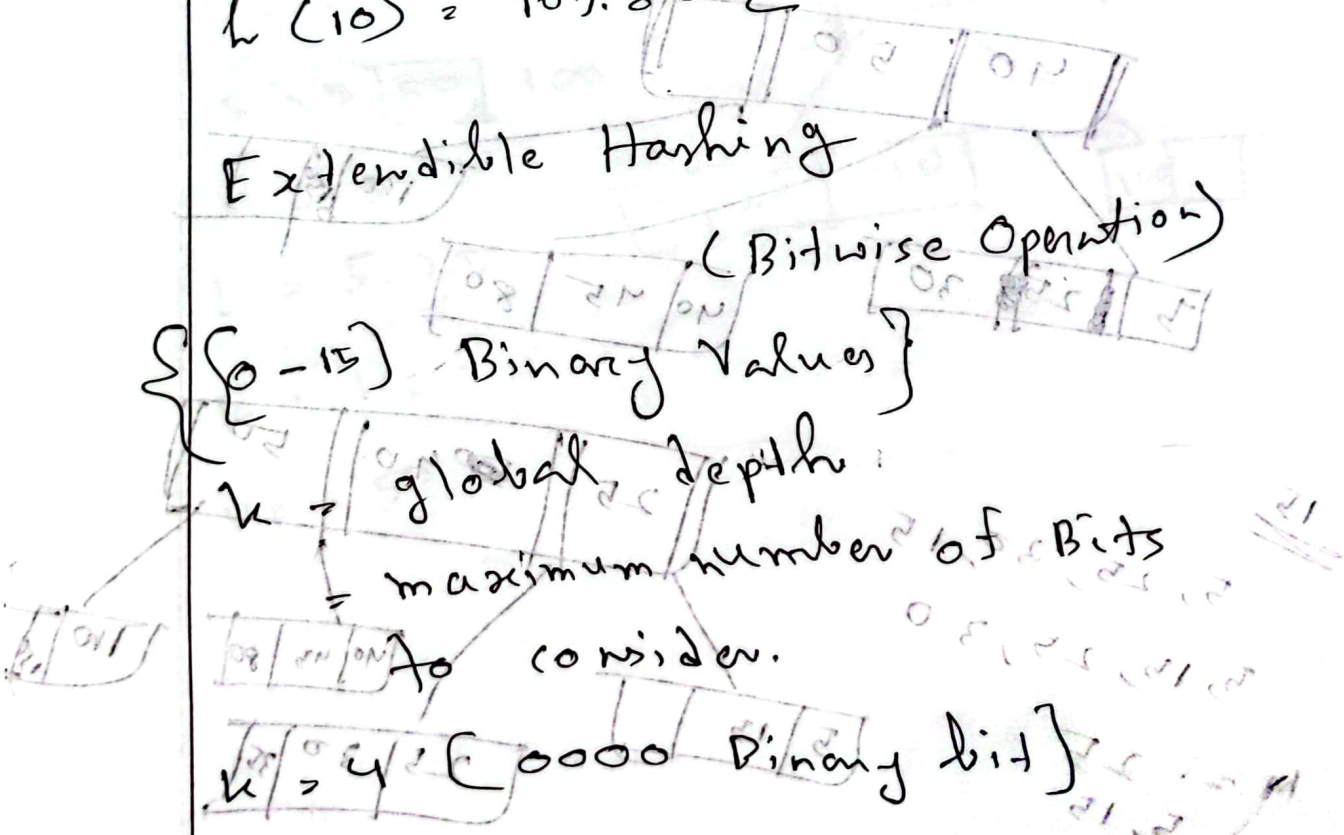
Extendible Hashing

(Bitwise Operation)

{(0-15) Binary Values}

k = global depth
= maximum number of Bits to consider.

k = 4 {0000 Binary bit}



$h(k) = k \% 8$ = Result within (0 - 7)

Maximum = 7 ; 7 binary = 111

Bit = 3, k = 3

$h(k) = k \% 12 + 3$ = Result within (3 - 14)

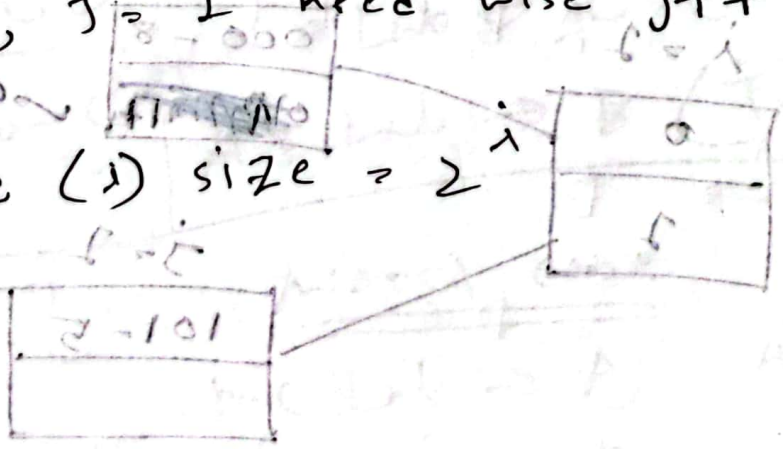
Maximum = 14 ; 14 binary = 1110

Bit = 4, k = 4

Local Depth

i, $i <= k$, $i = 1$ head $i++$

j, $j <= i$, $j = 1$ head wise $j++$



Hash table (1) size = 2^i

Insertion

8, $h(8) = 8\%.8 = 0$ Bin = 000

5, $h(5) = 5\%.8 = 5$ Bin = 101

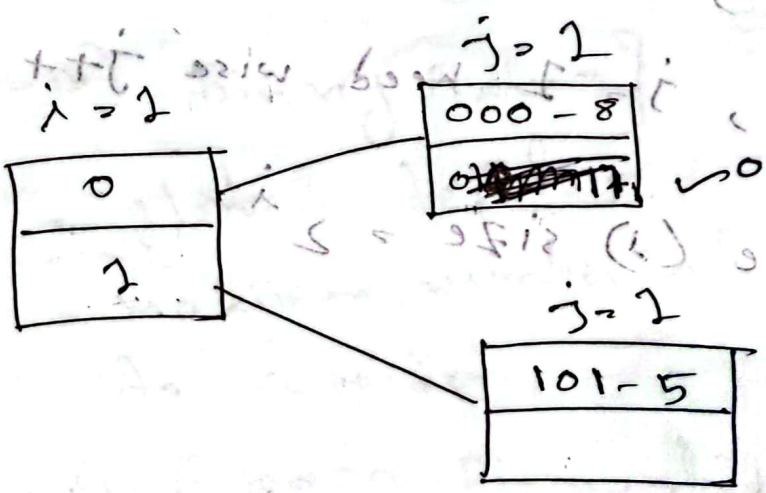
11, $h(11) = 11\%.8 = 3$ Bin = 011

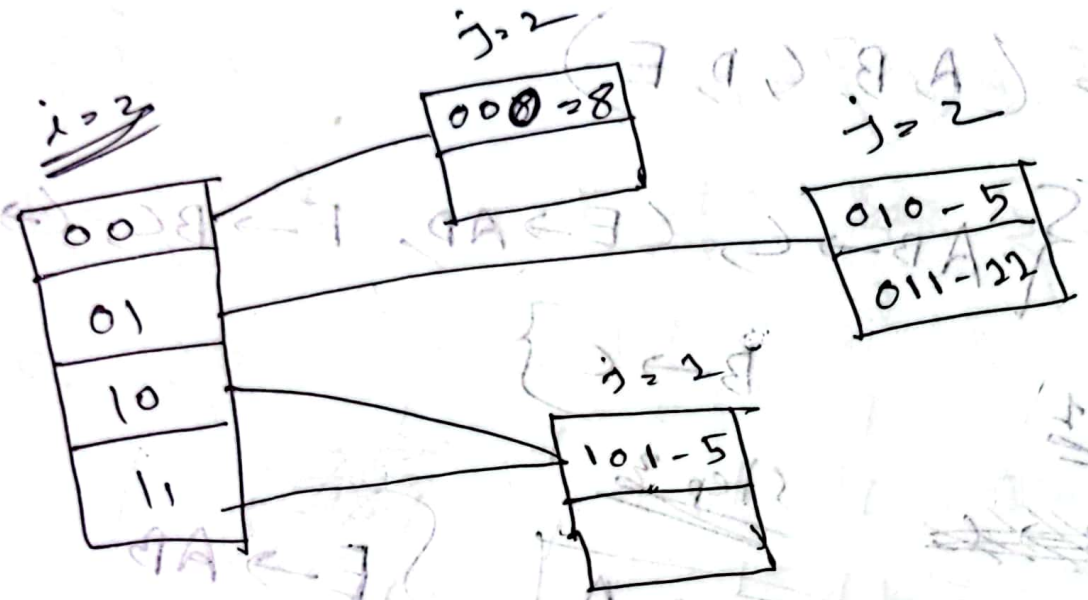
20, $h(20) = 20\%.8 = 4$ Bin = 100

Overflow

$j < i$ $j++$

$j = i$, $++i$ than $j++$





main

CN – A (Sat, Thu)

AI – A (Sat, Thu)

Micro – A (Sat, Thu)

AI Lab – A (W)

CN Lab – C (Tue)

Micro Lab – C (W)

Optional

AI → C, F (Sun, Wed)

MC → B, C (Sun, Wed)

CN → B (Sun, Wed)

MC Lab → C (8:30 – 11) W

CN Lab → C (Tue)

AI Lab → B (Tue)

AI → C, F

MC → B, C

CN → B

worst case

MC Lab → A (Tue)

CN Lab → D (Tue)

AI Lab → D, F (Sat)

$A \rightarrow B$, $AB \rightarrow C$

$A \rightarrow B$, $AC \rightarrow B$

$R = (A, B, C, D, E)$

$F = \{ AB \rightarrow C,\ CE \rightarrow AD,\ E \rightarrow BC,\ C \rightarrow D,\ B \rightarrow C \}$

step 1

~~AB → C~~

~~E → A~~

~~E → D~~

E → B

E → C

C → D

B → C

step 2

E → A

E → B

step 3

E → AB

~~E → C~~

C → D

B → C

$R_1 = \{ C, D \}$

$R_2 = \{ A, B, E, C \}$

BCNF

$F_D = \{ C \leftrightarrow D \}$

$F_D = \{ E \rightarrow AB,\ B \rightarrow C \}$

$E \rightarrow ABC$ ~~B → C~~

$B^+ = BC$ ... $R_4 = \{B, A, E, C\}$

$R_3 = \{B\}$ ... $F_D = \{E \rightarrow AD\}$

$F_D = \{B \rightarrow C\}$

R → R_1, R_3, R_4

---

* $F = \{ABC \rightarrow CDEF,\ C \rightarrow E,\ A \rightarrow B,\ D \rightarrow F\}$

step 1

~~$ABC \rightarrow C$~~

$ABC \rightarrow D$

$ABC \rightarrow E$

$ABC \rightarrow F$

$C \rightarrow E$

$A \rightarrow B$

$D \rightarrow F$

step 2

~~$ABC \rightarrow D$~~

~~$ABC \rightarrow E$~~

~~$ABC \rightarrow F$~~

$C \rightarrow E$

$A \rightarrow B$

$D \rightarrow F$

step 3

$A \rightarrow B$, $AB \rightarrow C$

$A \rightarrow B$, $AC \rightarrow B$

$AC \rightarrow D$

~~$AC \rightarrow F$~~

$C \rightarrow E$

$A \rightarrow B$

$D \rightarrow F$

step 4

$AC \rightarrow D$ ✓

$C \rightarrow E$ ✓

$A \rightarrow B$ ✓

$D \rightarrow F$ ✓

$F = \{ AB \to CD,\ B \to C,\ BC \to D,\ CD \to EF,\ E \to F \}$

1-1	1-2	1-3
~~AB → C~~	~~B → C~~	~~B → CD~~
AB → D	E → F	E → F
B → C	AB → D	~~A~~
~~BC~~ → D	B → D	~~B~~
CD → E	CD → E	CD → E
CD → F	~~CD → F~~	
E → F	~~E → F~~	

* $F = \{A \to BC,\ CD \to E,\ E \to C,\ D \to AEH,\ ABH \to BD,\ DH \to BC\}$

S-1

$A \to B$

$A \to C$

~~$CD \to E$~~

$E \to C$

$D \to A$

$D \to E$

$D \to H$

~~$ABH \to B$~~

$ABH \to D$

~~$DH \to B$~~

~~$DH \to C$~~

S-2

$A \to B$

$A \to C$

$E \to C$

$D \to A$

$D \to E$

$D \to H$

~~$ABH \to D$~~

~~$D \to B$~~

~~$D \to C$~~

Minimal cover

$A \to BC$ ✓

$E \to C$ ✓

$D \to AEH$ ✓

$AH \to D$ ✓

2NF

* Non-Prime Attribute ≠ proper subset of ck

3NF

R.H.S = Prime Attribute

L.H.S = if R.H.S Non Prime, L.H.S super key.

BCNF

L.H.S → [Super key]

$F = \{A \to B,\ D \to E,\ B \hookrightarrow D,\ \cancel{AD \to E}\}$

$(A \to B,\ D \to E,\ B \hookrightarrow D)$

$R = \{A, B, C, D, E\}$

$A^+ = AB$

$R_1 = \{A, B\}$

$F_1 = (A \to B)$

BCNF

$D^+ = DE$

$R_3 = \{D, E\}$

$F_D = (D \to E)$

BCNF

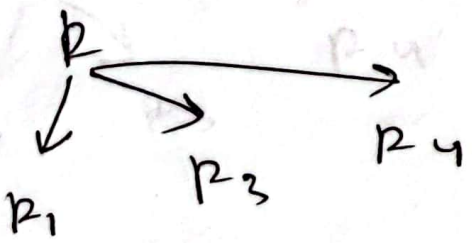
$R_2 = \{C, D, E, A\}$

$F_2 = (AC \to D, D \to E)$

$R_4 = \{A, C, D\}$

$F_D = \{AC \to D\}$

BCNF



$F = \{ AB \to C, E \to AD, E \to B, C \to D, B \to C \}$

$F_c = \{ E \to AD, E \to B, B \to C \}$

$AB \to C$

$E \to AB$

$E \to AD$

$C \to D$

$E \to B$

$B \to C$

$E \to C$

$C \to D$

$B \to C$

$R = \{A, B, C, D, E\}$

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

$C^+ = CD$

$R_1 = \{C, D\}$

$F = \{C \to D\}$

BCNF

$R_2 = \{C, A, B, E\}$

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

$B^+ = BCA$

$R_3 = \{B, C\}$

$F = \{B \to C\}$

BCNF

$R_4 = \{A, E, B\}$

$F = \{E \to AB\}$

BCNF

R → R_1, R_3, R_4

C.W
4.5.24

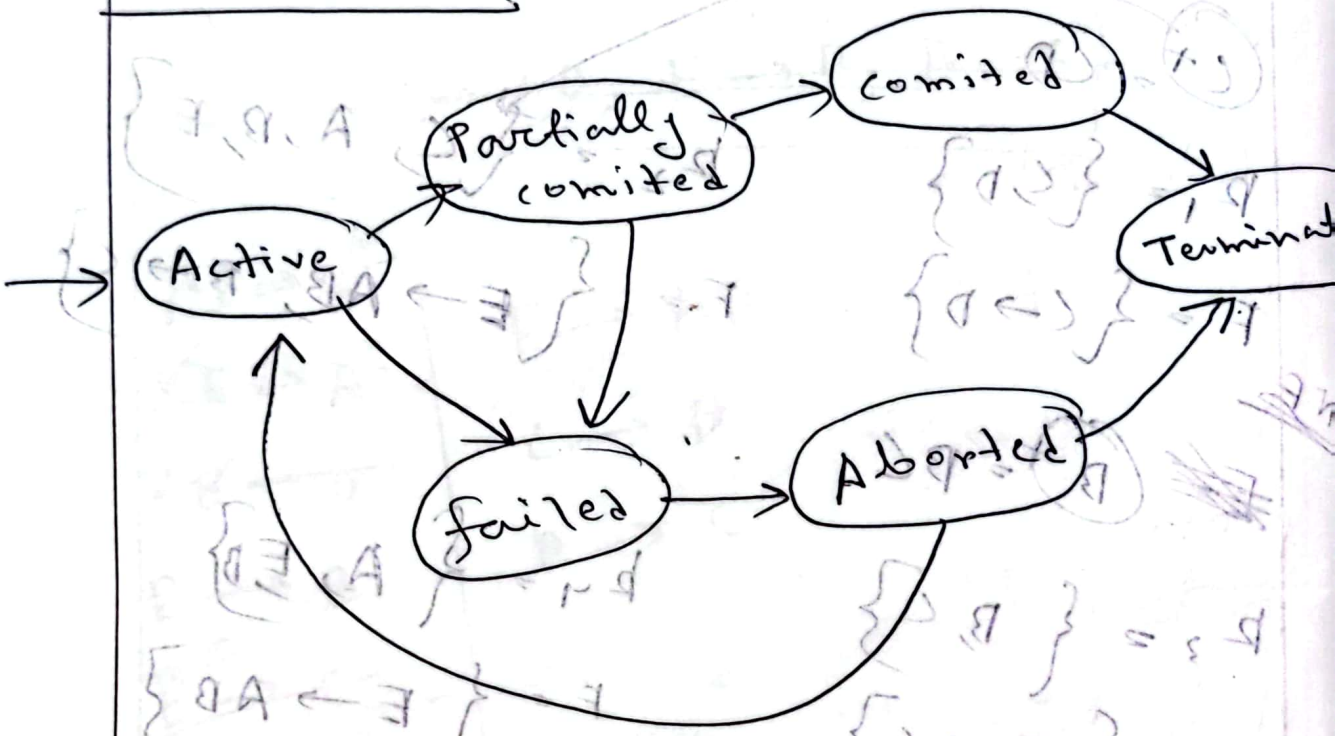
01

DBMS

* Indexing

6.5.24

Transaction



* Cycle thakle serial

↓

graph

closed hashing / open addressing

* Linear Probing

$$h(k,i) = (h'(k) + i) \% m; \quad i = 0, 1, \ldots m-1$$

$$h'(k) = k \% 7$$

$$h(k,i) = \{(k \% 7) + i\} \% 7$$

{76, 93, 40, 47, 10, 55}

0	1	2	3	4	5	6
47	55	93	10		40	76

76

$\{(76 \% 7) + 0\} \% 7$

$6 \% 7 = 6 \quad [i = 0]$

93

$\{(93 \% 7) + 0\} \% 7 \quad [i = 0]$

$2 \% 7 = 2$

40

$\{(40 \% 7) + 0\} \% 7 \quad [i = 0]$

$5 \% 7 = 5$

47

$\{(47 \% 7) + 0\} \% 7 \quad [i = 0]$

$= (5 + 7) \% 7 = 7 \quad [i = 1]$

$= 6 \% 7 = 6$

$= 7 \% 7 = 0 \quad [i = 2]$

10

$\{(10 \% 7) + 0\} \% 7 \quad [i = 0]$

$= 3 \% 7 = 3$

55

$\{(55 \% 7) + 0\} \% 7 \quad [i = 0]$

$= 6$

$= (6 + 2) \% 7 \quad [i = 2]$

$= 8 \% 7 = 1$

Indexing

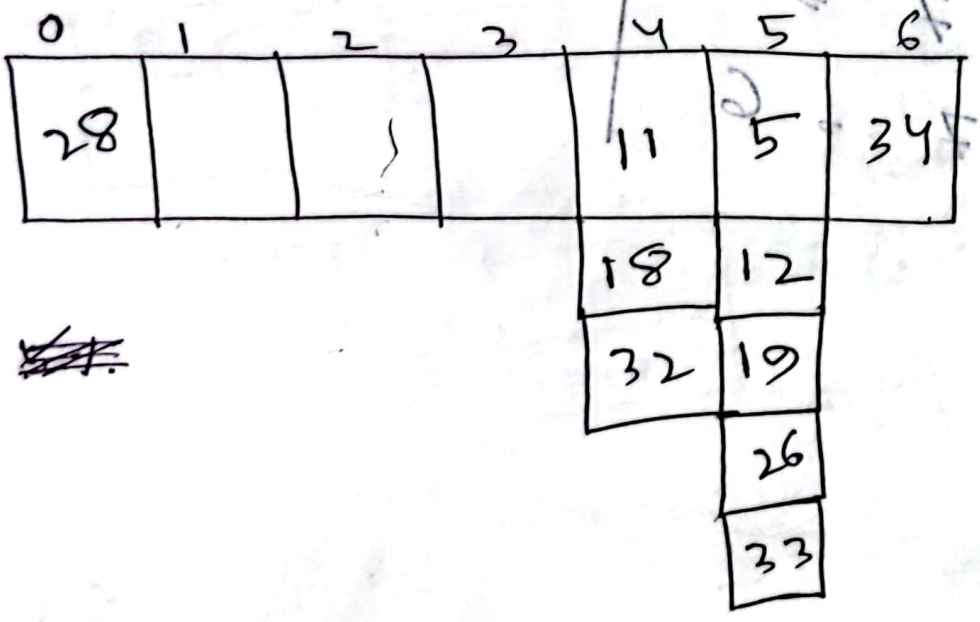
* Order index → (B^+ tree)
* unorder index

primary index

* Index structure

(Index key, pointer)

2.



$5 \% 7 = 5$

$11 \% 7 = 4$

$12 \% 7 = 5$

$19 \% 7 = 5$

$26 \% 7 = 5$

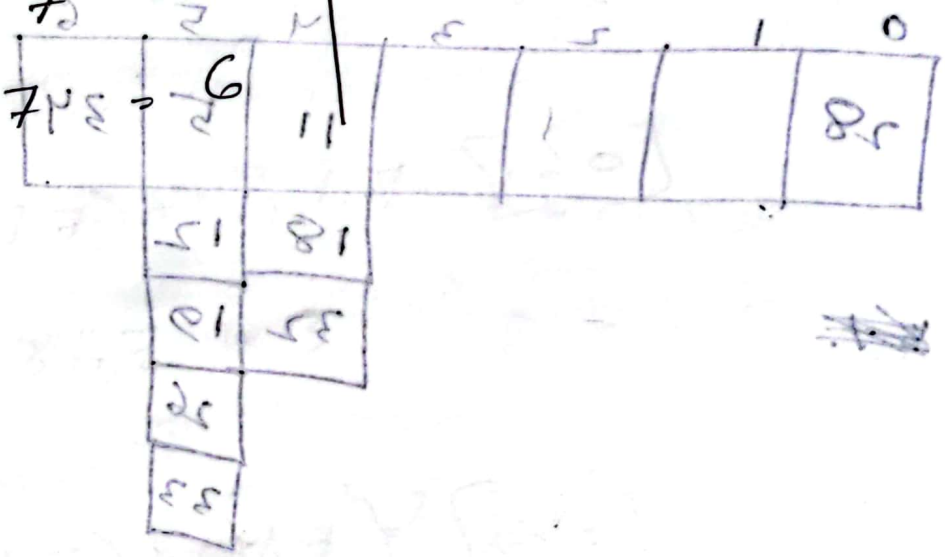
$28 \% 7 = 0$

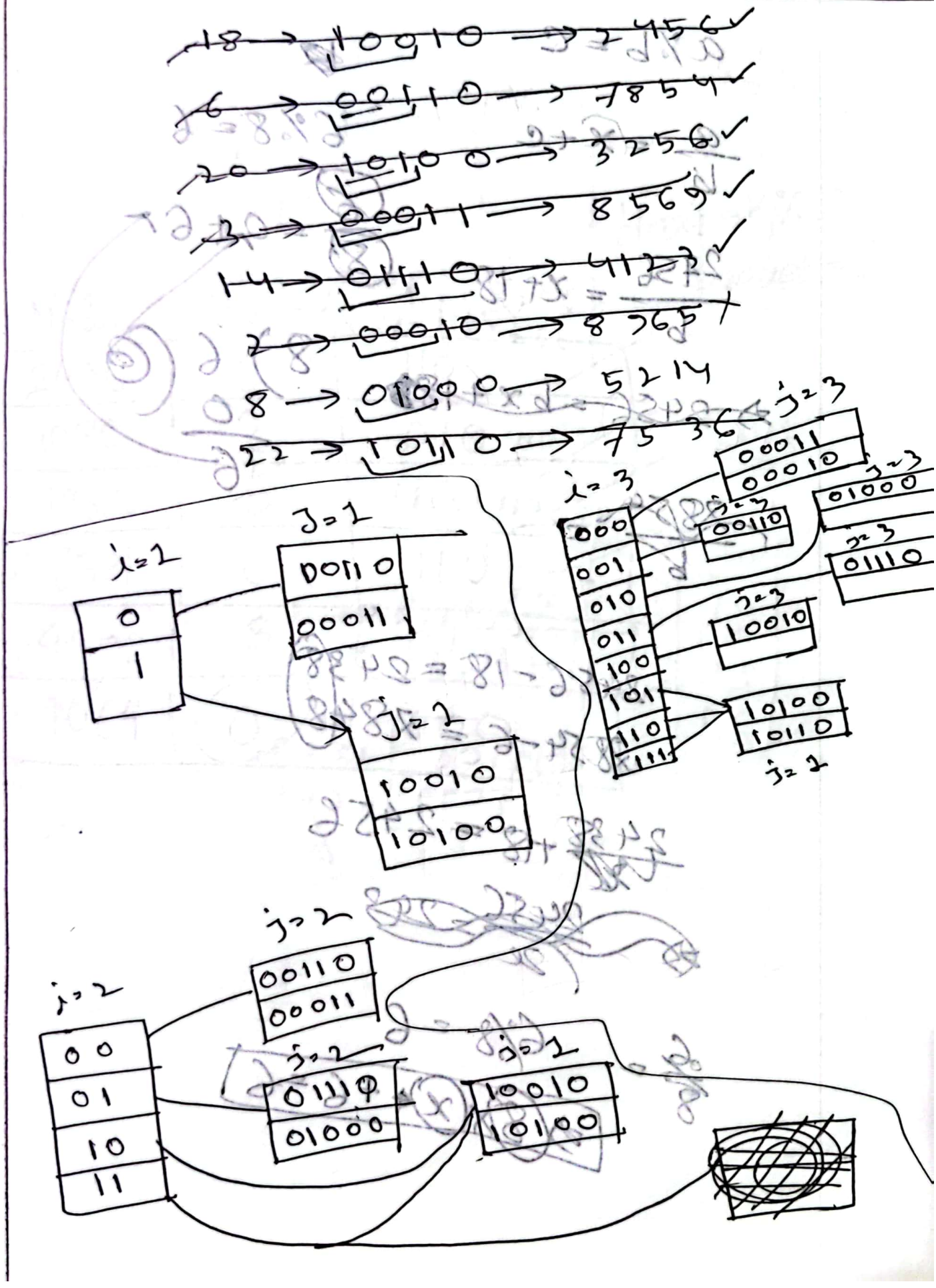
$18 \% 7 = 4$

$33 \% 7 = 5$

$34 \% 7 = 6$

$32 \% 7 = 4$




18 → 10010 → 2456
6 → 00110 → 7854
20 → 10100 → 3256
3 → 00011 → 8569
14 → 01110 → 4112
2 → 00010 → 8765
8 → 01000 → 5214
22 → 10110 → 7536
i=2
0
1
j=1
00110
00011
j=2
10010
10100
i=3
000
001
010
011
100
101
110
111
j=3
00011
00010
01000
01110
00110
10010
10100
10110
i=2
00
01
10
11
j=2
00110
00011
01110
01000
j=1
10010
10100

$a \% b = c$

$\frac{a}{b} = x + c$

$\frac{2456}{b} = x + 18$

$2456 = bx + 18$

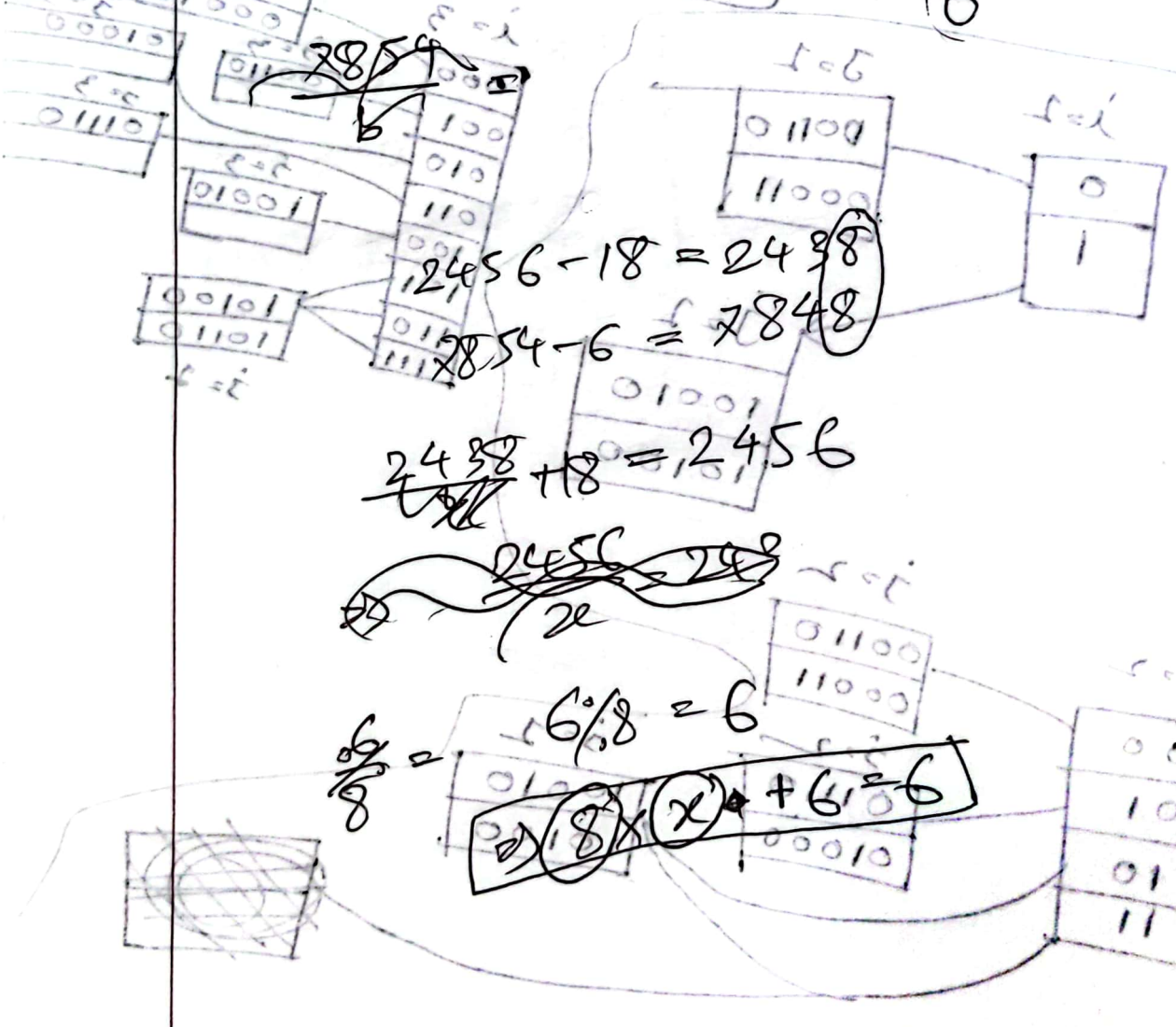
$6 \% 8 = 6$

$= 0 + 6$

8) 6

0

6



Exercise 3.2.2, 3.2.1, 3.2.6, 8, 9

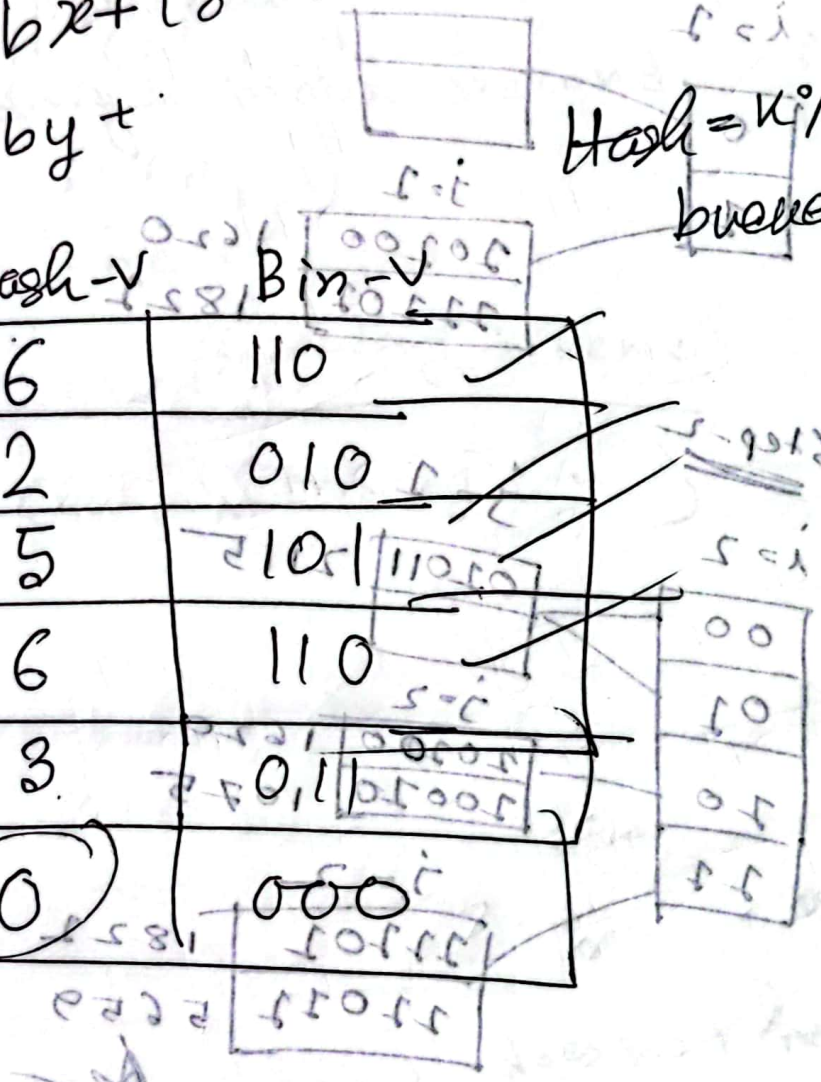
$bx + 18 = 2456$

$by +$

Hash = K % 7

bucket = 2

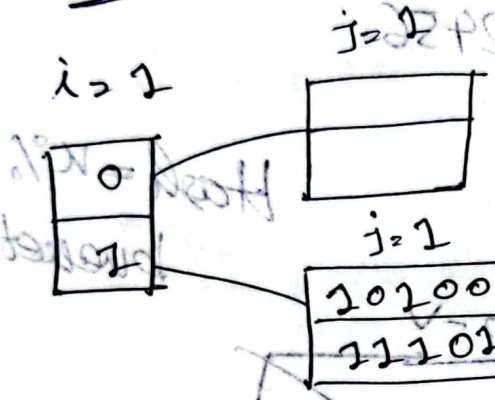
Value	Hash-V	Bin-V
2456	6	110
1080	2	010
1922	5	101
6481	6	110
4392	3	011
1064	0	000



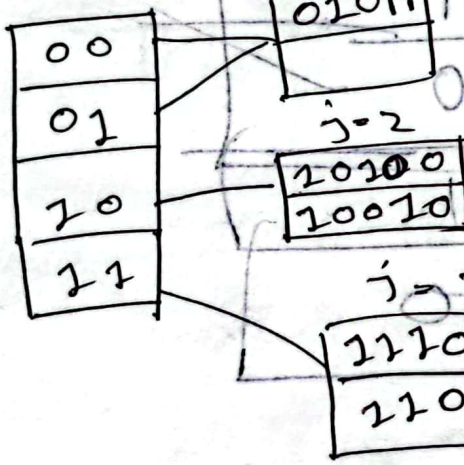
Spr-22

2 (b)

step-1



step-2



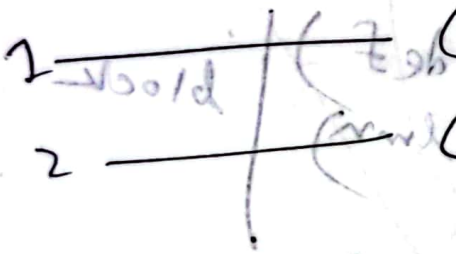
Dense index

Hard disk per data record has a corresponding index record.

(one to one)

(Secondary + Dense)

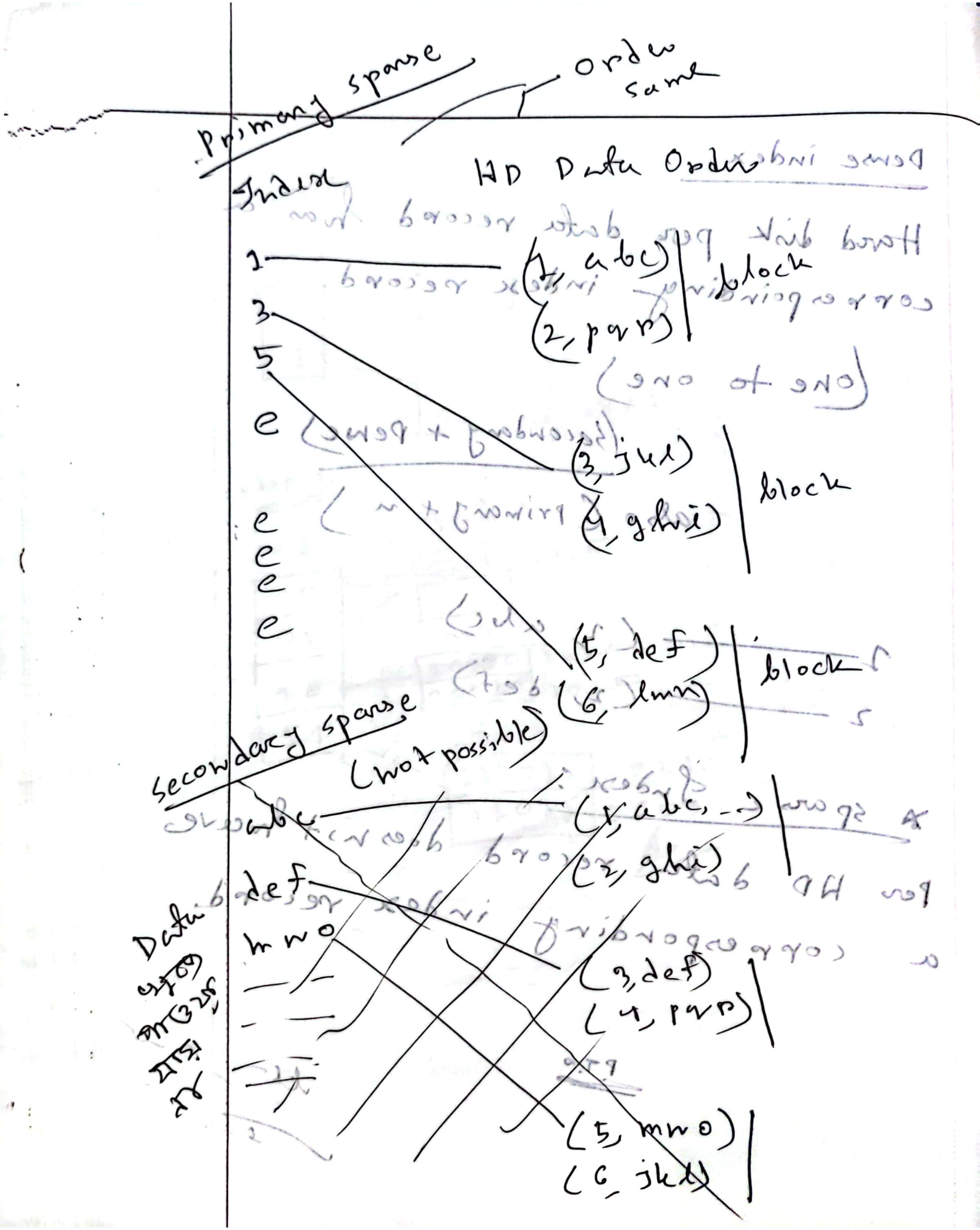
(also (primary + u))



* sparse index:

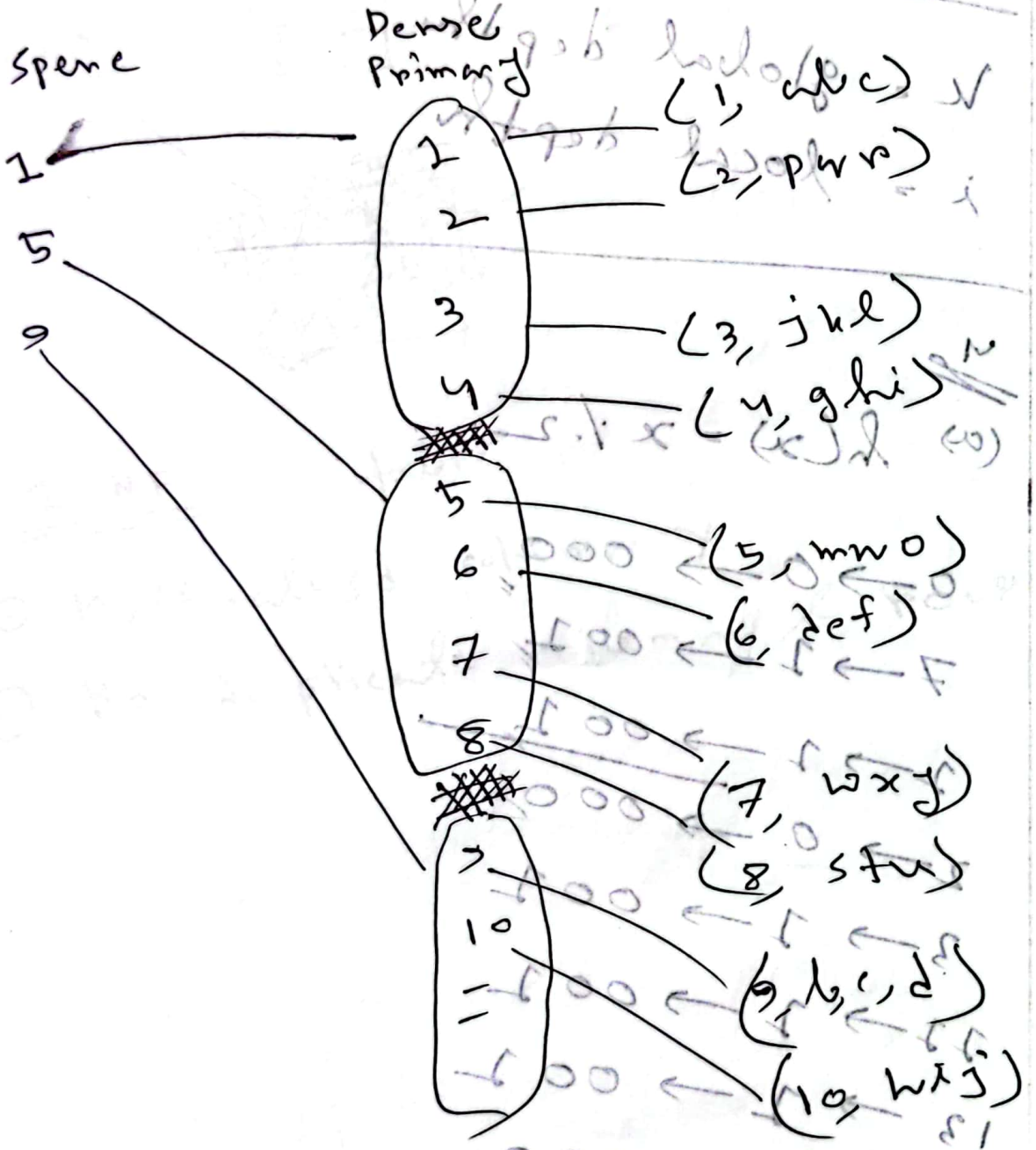
Per HD data record doesn't have a corresponding index record.

P.T.O


Primary sparse
order same
Index
HD Data Ordered
1
3
5
e
e
e
e
e
(1, abc)
(2, pqrs)
block
(3, jkl)
(4, ghi)
block
(5, def)
(6, lmn)
block
5
Secondary sparse
(not possible)
abc
def
lmn
(1, abc, ...)
(2, ghi)
(3, def)
(4, pqrs)
(5, mno)
(6, jkl)

(B+ tree Exm)

* Multi level indexing



(* Faster data search.)

Extendable Hashing

k → global depth

i → local depth

4

(a) $h(x) = x \% 2$

0 → 0 → 000

7 → 1 → 001

1 → 1 → 001

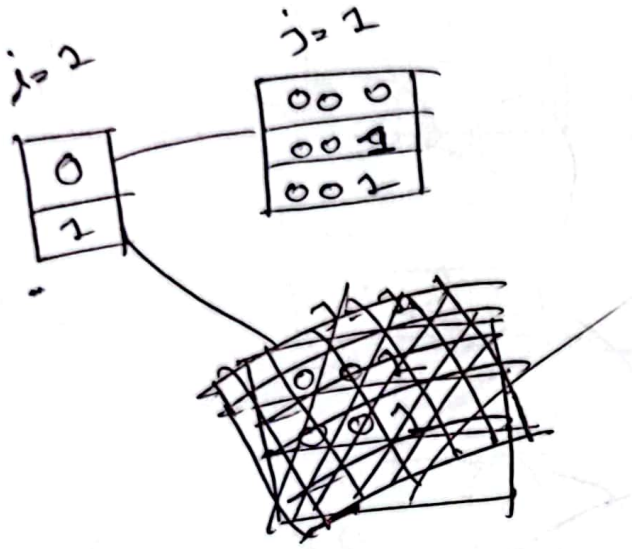
2 → 0 → 000

3 → 1 → 001

11 → 1 → 001

13 → 1 → 001

22 → 0 → 000

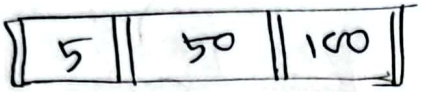


1 NF table

① Multivalued column ✗

② No duplicate column & rows

5, 50, 100, 25, 40, 45, 150, 80, 30, 15, 35



25 → copy up

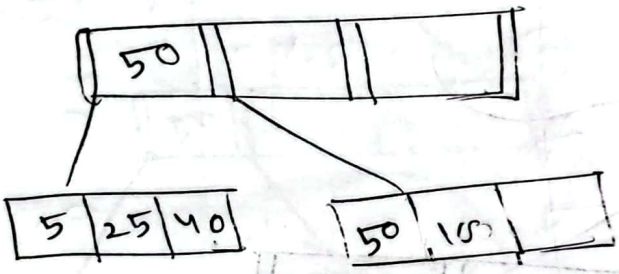
5, 25, 50, 100

M - 50

L - 5, 25

R - 50, 100

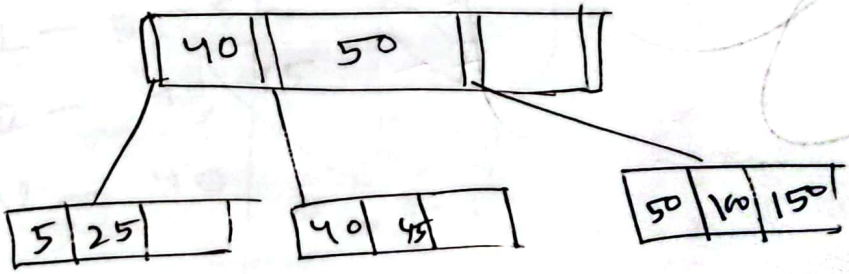
V - 50



45 → copy up

5, 25, 40, 45

M - 40, L - 5, 25, R - 40, 45, V - 40

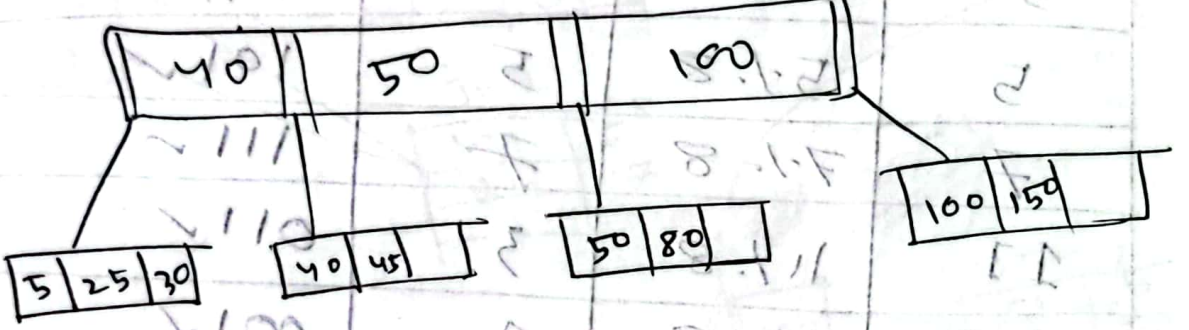


Good

80 → copy up

50, 80, 100, 150

M - 100; L - 50, 80, R - 100, 150, U - 100



15 → copy up

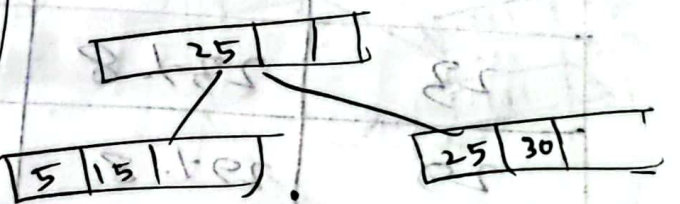
5, 15, 25, 30

M - 25

L - 5, 15

R - 25, 30

U - 25



25 → push up

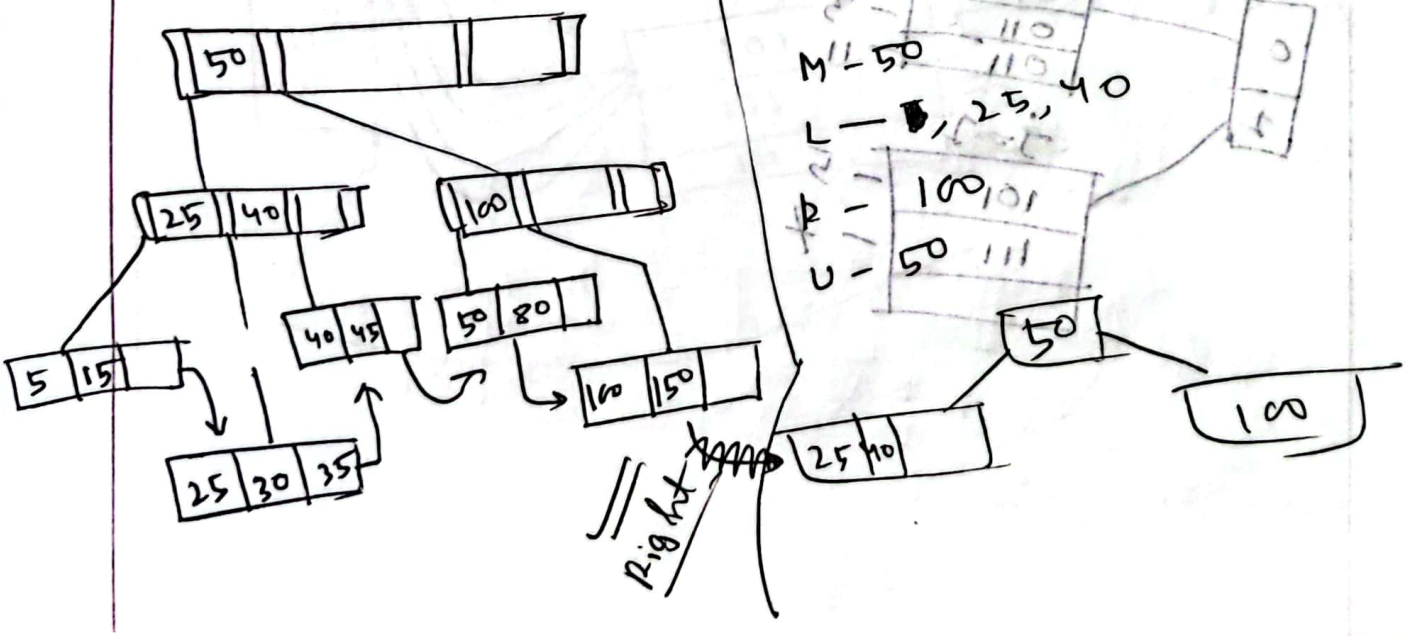
25, 40, 50, 100

M/L - 50

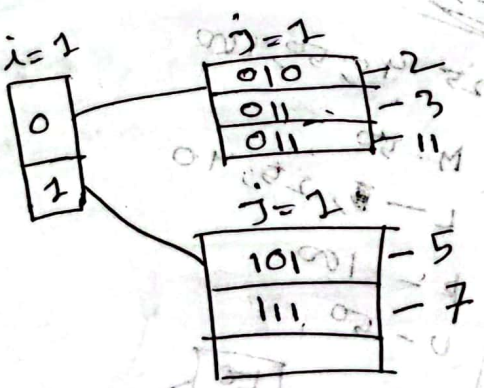
L - 25, 40

R - 100

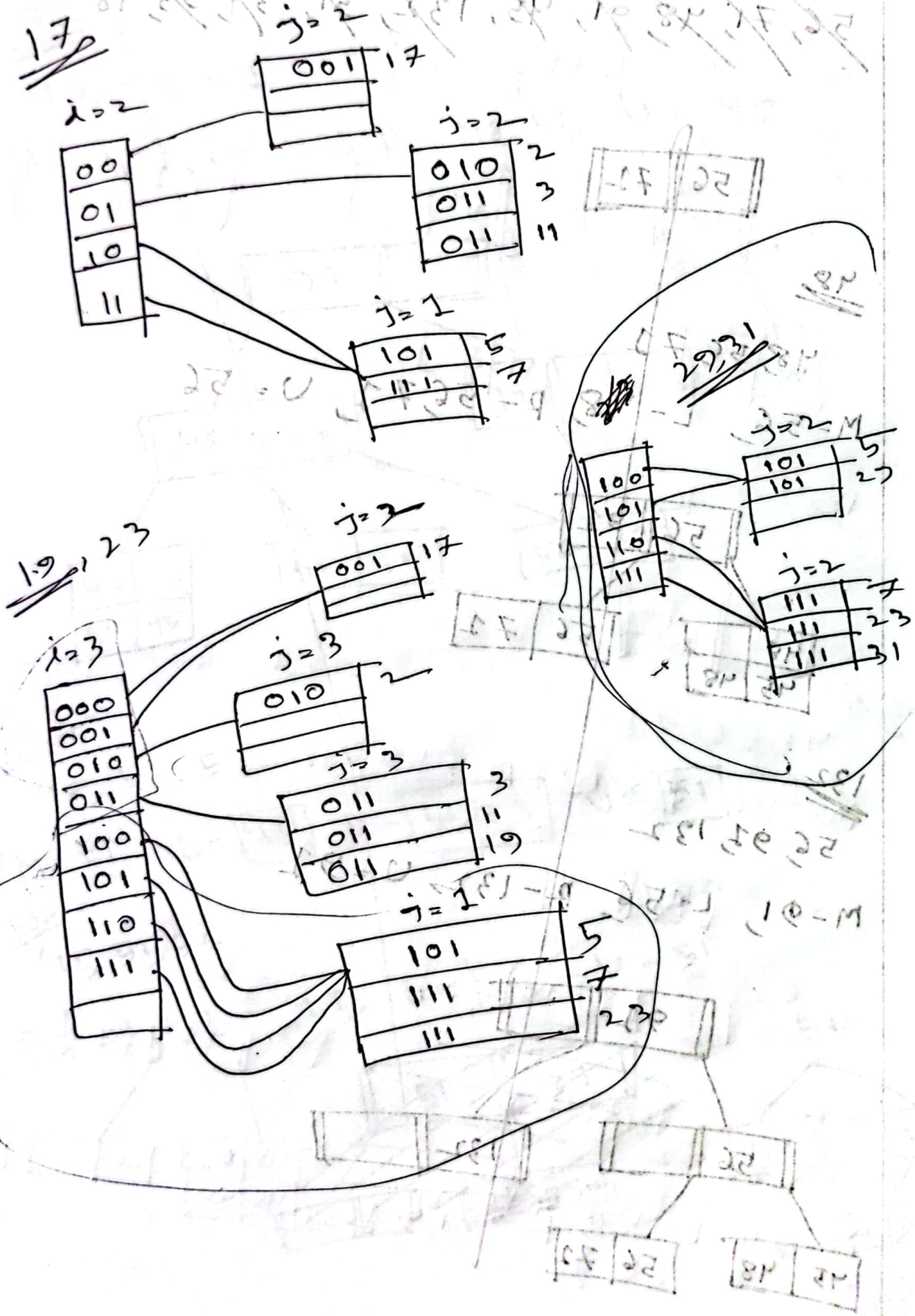
U - 50

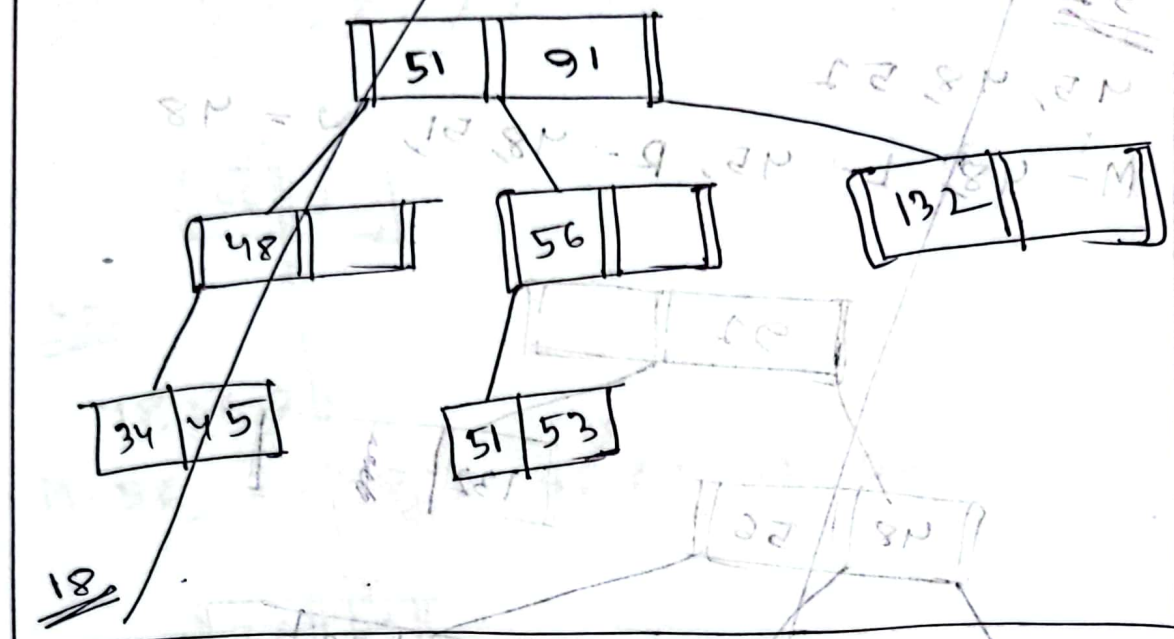


key	h(key)	Am	
2	2%.8	2	010
3.	3%.8	3	011 ✓
5	5%.8	5	101 ✓
7	7%.8	7	111 ✓
11	11%.8	3	011 ✓
17	17%.8	1	001 ✓
19	19%.8	3	011 ✓
23	23%.8	7	111 ✓
29	29%.8	5	101 ✓
31	31%.8	7	111

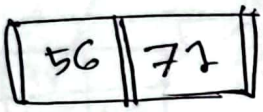


17


i=2
00
01
10
11
j=2
001
17
j=2
010
011
011
2
3
11
j=1
101
111
5
7
i=3
000
001
010
011
100
101
110
111
j=3
010
j=3
011
011
011
3
11
19
j=1
101
111
111
5
7
23
j=2
100
101
110
111
j=2
101
101
5
23
j=2
111
111
111
7
23
31



18



48

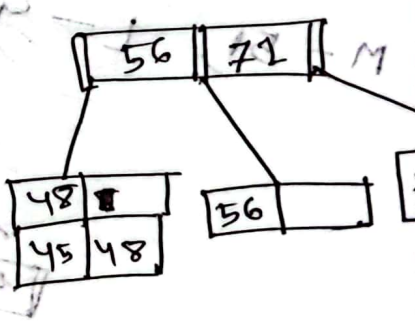
48, 56, 71

M - 56, L → 48, R → 56, 71

91

56, 71, 91

M - 71, L - 56, R → 71, 91

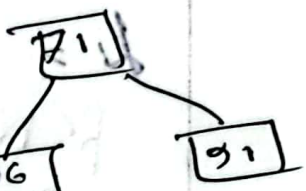


56, 71, 91

M → 71

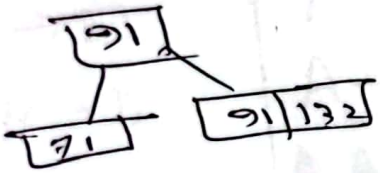
L → 56

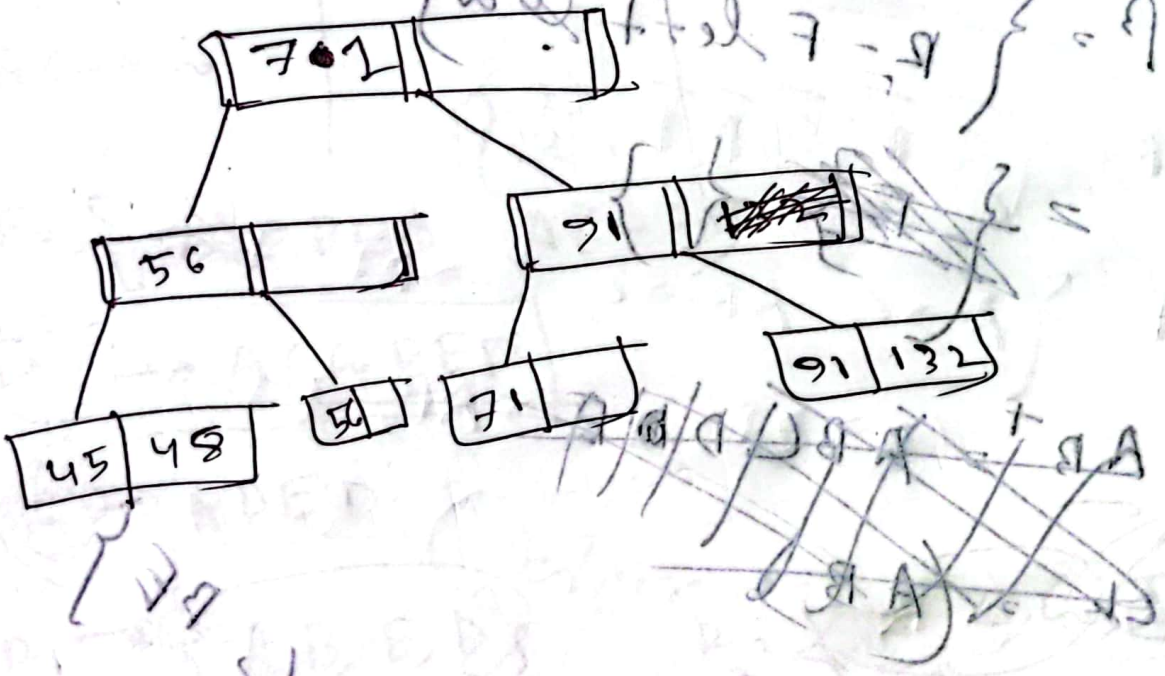
R → 91



91, 132

M → 91, L → 71, R → 91, 132





$R_3 = \{A, C, W\}$

$F_3 = \{A \rightarrow C, CW \rightarrow A\}$

Summer-23

$R = \{\underline{A}, \underline{B}, \underline{C}, \underline{D}, E, \underline{I}\}$

$F = \{A \rightarrow C,\ AB \rightarrow C,\ C \rightarrow DI,\ CD \rightarrow DI,\ EC \rightarrow AB,\ EI \rightarrow C\}$

$\alpha = (C, R - F \text{ right})$

$\{W\}^+ = \{W, E\}$

$\rho = \{R - F \text{ left } \alpha\}$

$= \{\}$

$E^+ = E$ $\{1, J, B, A\} = 4$

$EA^+ = \{EACDIB$ $A \leftarrow 1,\ J \leftarrow 1A\} = 1$

$EB^+ = EB$ X

$EC^+ = ECDIAB$

$ED^+ = ED$ X

$EJ^+ = EICDAB$

$EBD^+ = \{EBD\}$ X

$CK = \{EA, EC, EJ\}$

2NF check:

Partial Dependency check:

Proper subset = $\{E, A, C, J,$ ~~AA, AA,~~ EA,

CK = ~~E, E, A, A, C I~~, $\{\}\}$

Now prime Attribute = $\{B, D\}$

proper subset of CK ≠ Non prime

(2NF)

$A \to C$, $AB \to C$, $C \to DI$, $CD \to I$, $EC \to AD$, $EI \to C$

$A \to C$

~~AB~~

$C \to D$

$C \to I$

~~AC~~

$EC \to A$

$EC \to B$

$EI \to C$

$\{AC, C \to DI, EC \to AB, EI \to C\}$

$R = \{A, B, C, D, E, I\}$

$A \to ACDI$

$C \to CDI$

$EC \to ABECDI$

$EI \to CEIABD$

$R_1 = \{C, D, I\}$

$F_1 = \{C \to DI\}$

BCNF

$R_2 = \{C, A, E, B\}$

$F_2 = \{EC \to AB\}$

~~Not BCNF~~

BCNF

A	B	C	D
1	1	2	3
1	2	2	3
1	3	2	3
2	4	5	6
5	6	7	8

F_D {A → D, D → A C, A → C}

R = {A, B, C, D}

α = {R − F Right}
= {B}

β = {R − F left}
= { }

Transitive D check

(i) R.H.S → Prime Attribute

(ii) L.H.S → R.H.S Non prime → L.H.S SK

Pertial D check

(i) Proper subset of CK ≠ Non-prime attribute

$B^+ = B$

$BA^+ = ABDC$

$BC^+ = BCDA$

$BD^+ = BDAC$

P.A → {A, B, C, D}

N.P.A → { }

CK: {BA, BC, BD}

2NF Check:

Proper subset of CK = {A, B, C, D}

Non prime attribute { }

0	00
1	10
0	01
1	11

{2NF}

3NF

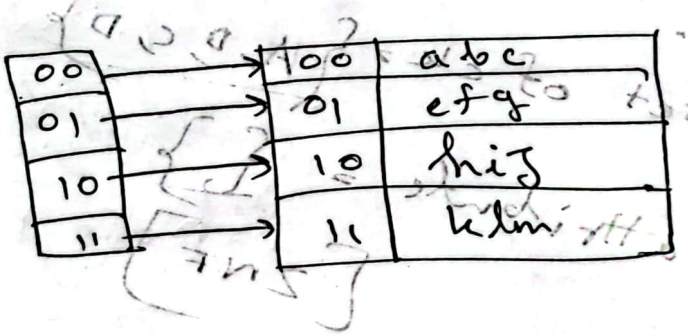
R.H.S = Prime Attribute [cover]

[3NF]

Sum-22

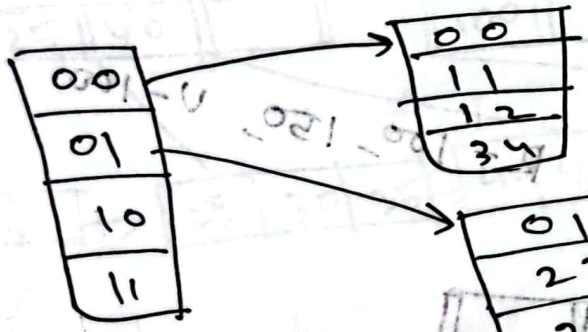
① Dense Index:

When from one table every block is connected with another table's every block than this type of indexing is known as Dense Indexing.

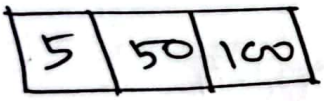


Sparse Index:

When one table's one block is connected with another's table's different block than this type of indexing is known as sparse indexing.



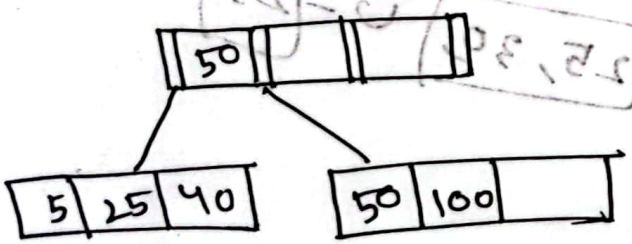
* 5, 50, 100, 25, 40, 45, 150, 80, 30, 15, 35



25

5, 25, 50, 100,

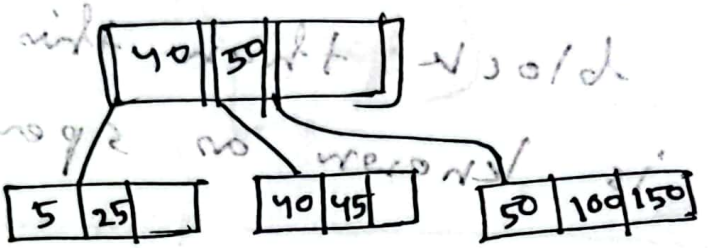
M - 50, L - 5, 25, R - 50, 100,



45

5, 25, 40, 45

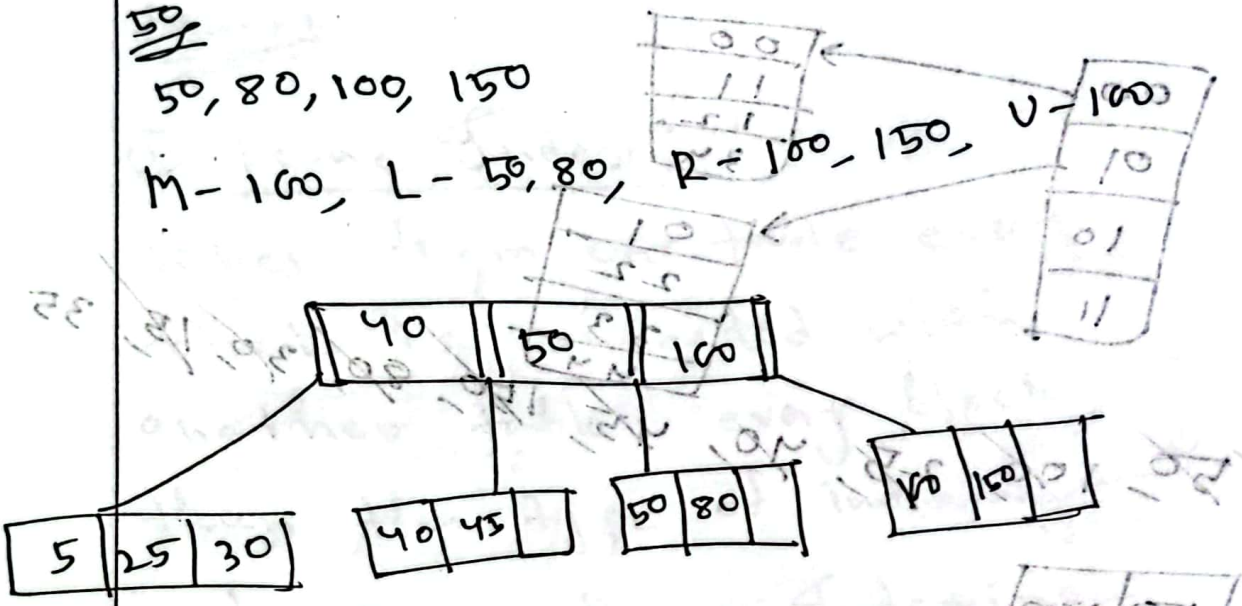
M = 40, L = 5, 25, R = 40, 45, U = 40



50

50, 80, 100, 150

M - 100, L - 50, 80, R - 100, 150, U - 100

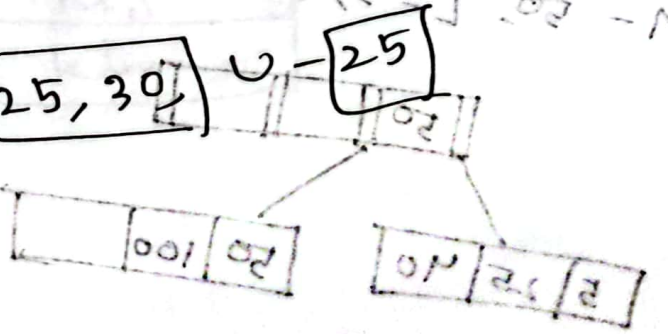


15

5, 25, 30 - 15

5, 15, 25, 30

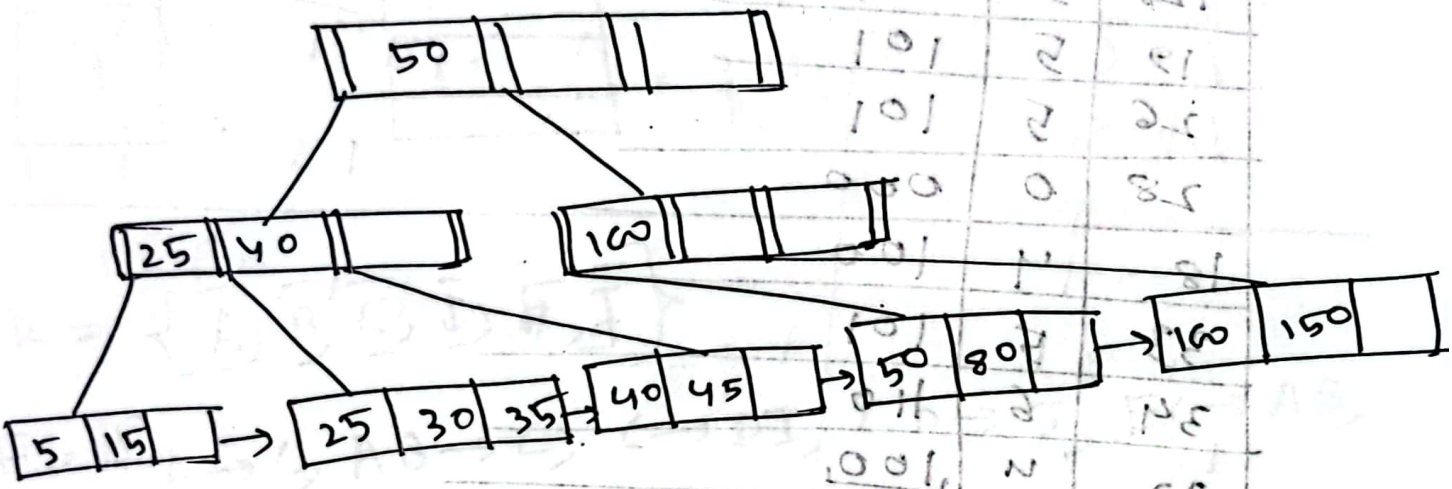
M = 25, L - 5, 15, R - 25, 30, U - 25

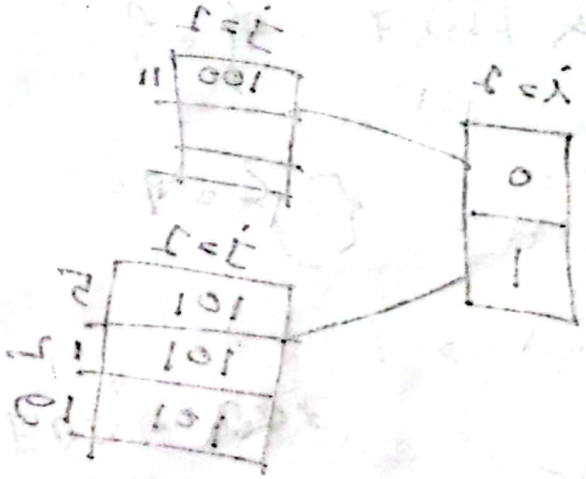


25 → push up

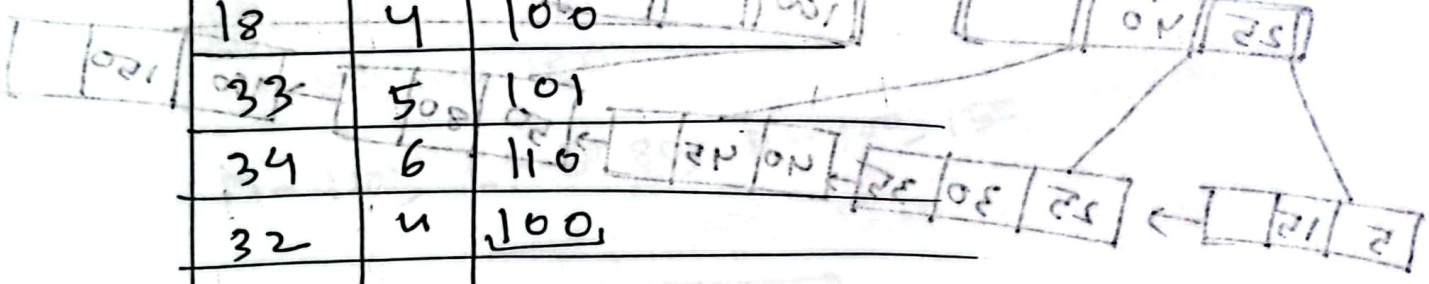
25, 40, 50, 100

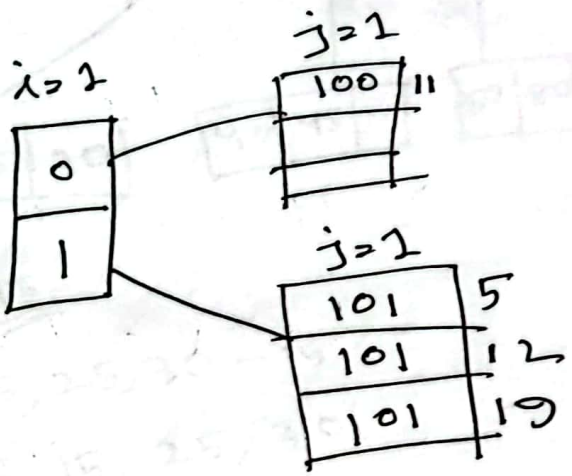
M = 50, L - 25, 40, R - 100

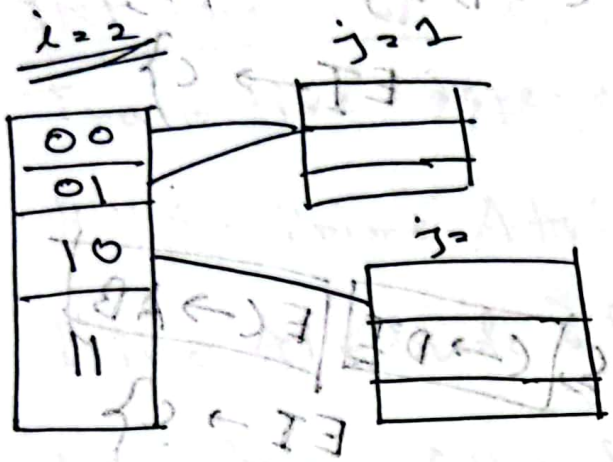




5	5	101
11	4	100
12	5	101
19	5	101
26	5	101
28	0	000
18	4	100
33	5	101
34	6	110
32	4	100







$R = \{A, B, C, D, E, I\}$

$F = \{A \to C,\ AB \to C,\ C \to DI,\ CD \to I,\ EC \to AB,\ EI \to \}$

$\alpha = \{R - F\ Right\} = \{E\}$, $\beta = \{R - F\ Left\} = \{\}$

$E^+ = E$ X

$EA^+ = AECDIB$

$AEB^+ = BE$ X

$EC^+ = CEDIAB$

$ED^+ = DE$ X

$EI^+ = EICDAB$

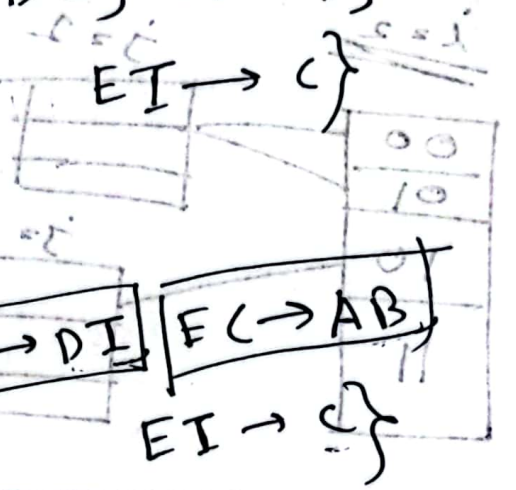
$EBD^+ = BDE$

$CK = \{EA, EC, EI\}$

$ACD^+ = ACDI$

$DCI^+ = BCID$

$\{A \to C,\ AB \to C,\ C \to DI,\ CD \to I,\ EC \to AB,\ EI \to C\}$



Decomposition

$A \to C$

~~$AB \to C$~~

$C \to D$

$C \to I$

~~$CD \to I$~~

$EC \to A$

$EC \to B$

$EI \to C$

$F_{D_C} = \{A \to C,\ C \to DI,\ EC \to AB,\ EI \to C\}$

$R = (A, B, C, D, E, I)$

3.5 NF

L.H.S = {Superkey}

Check $\{E\}^+$

$A^+ \to ACDI$

Not 3.5 NF

3NF check:

Transitive dependency check:

R.H.S = Prime Attribute

Prime Attribute = { A, C, E, I }

Non Prime Att = { B, D }

R.H.S ≠ P.A

[Not 3NF]

2NF check:

Partial dependency check:

Proper subset of ck = { A, C, E, I }

Non prime attribute = { B, D }

~~proper subset of ck ≠ Non prime attribute~~

~~[2NF]~~

{C → D I}

{E → A B}

proper subset of ck ⇒ Non prime attribute

[Not 2NF]

[Must be 1 NF.]

$AB \rightarrow C$, $AD \rightarrow GH$, $BD \rightarrow EF$,

$A \rightarrow I$, $H \rightarrow J$, $I \rightarrow BD$

Decomp

$AB \rightarrow C$

$AD \rightarrow G$

$AD \rightarrow H$

$BD \rightarrow E$

$BD \rightarrow F$

$H \rightarrow J$

$I \rightarrow BD$

$A \rightarrow I$

$R = \{A, B, C, D, E, F, G, H, J, J\}$

$A \rightarrow C$

$A^+ \rightarrow AIBDCEFHGJ$

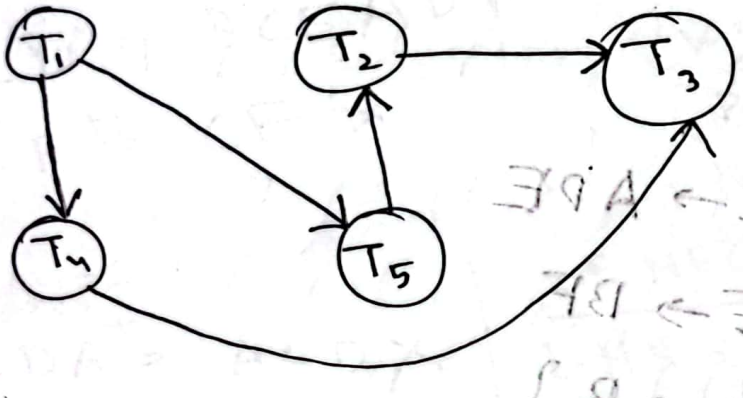
$\rightarrow$ valid

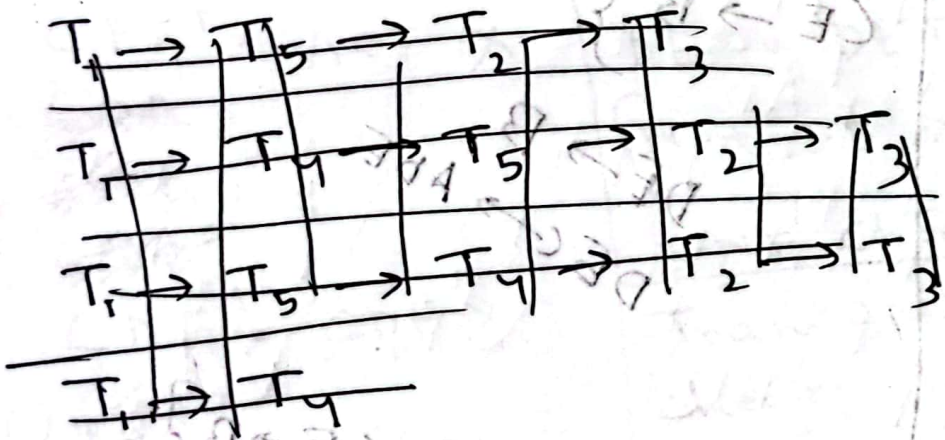
$I^+ \rightarrow IBDEF$

$\rightarrow$ Not valid

* $G^+ \rightarrow G$

Not prime Attribute.





$T_1 \rightarrow T_4 \rightarrow T_5 \rightarrow T_2 \rightarrow T_3$

$T_1 \rightarrow T_5 \rightarrow T_4 \rightarrow T_2 \rightarrow T_3$

$R = \{ A, B, C, D, E, F \}$

$F = \{ ABC \rightarrow DE,\ BC \rightarrow A,\ DE \rightarrow BF,\ CE \rightarrow BE \}$

Decomp

~~$ABC \rightarrow D$~~
~~$ABC \rightarrow E$~~
$BC \rightarrow A$
$DE \rightarrow B$
$DE \rightarrow F$
$CE \rightarrow B$
~~$CE \rightarrow E$~~

$\{ BC \rightarrow ADE.$
$DE \rightarrow BF$
$CE \rightarrow B \}$

$DE \rightarrow BF$
$DE \rightarrow ADE.$

$F = \{ BC \rightarrow ADE,\ DE \rightarrow BF,\ CE \rightarrow B \}$

$\alpha = \{ R - F \text{ right} \}$ $\beta = \{ R - F \text{ left} \ \& \ \alpha \}$

$= \{ C \}$ $= \{ A, F \}$

$C^+ = C$

$CA^+ = CA$

$CB^+ = CBADF$

$CD^+ = CD$

$CE^+ = BCEADF$

$CF^+ = CF$

CK = {BC, E}

P.A = {B, C, E}

N.P.A = {A, D, F}

~~CA~~

$CDA^+ = ACDX$

$CFA^+ = ACFX$

~~CDA~~

~~CDB~~

~~CDC~~

$CDF^+ = CDFX$

~~CDE~~

~~CDEF~~

~~CDEF~~

(Must be 2NF)

3.F NF Check

L.H.S = SK

$DE^+ = DEBF$ X

(Not 3.5 NF)

3 NF Check

Transitive dependency check:

R.H.S = Prime Attribute

R.H.S ≠ P.A

(Not 3 NF)

2NF check

Partial depend check

P.S.C.K = {B, C, E, {}}

P.S.C.K → N.P.A

(Not 2NF)

$R = \{ A, B, C, D, E, F \}$

$F = \{ BC \to ADE,\ DE \to BF,\ CE \to B \}$

$DE \to BDEF$

$R_1 = \{ B, D, E, F \}$ $\quad R_2 = \{ D, E, A, C \}$

$F_1 = \{ DE \to BF \}$ $\quad F_2 = \{ B \}$

$BC \to ABCDEF$

$CE \to BCEAD$

$R_1 = \{ B, D, E, F \}$

$F_1 = \{ DE \to BF \}$

BCNF

$R_2 = \{ A, C, D, E \}$

$F_2 = \{ DE \to ADE \}$

(BCNF)

Spring 22

* Decomposition

if $A \rightarrow BC$

then $A \rightarrow B$, $A \rightarrow C$

* Lossless Decomposition

→ $R = R_1 \cup R_2 \cup R_3$

And common element between R_1, R_2 & R_3

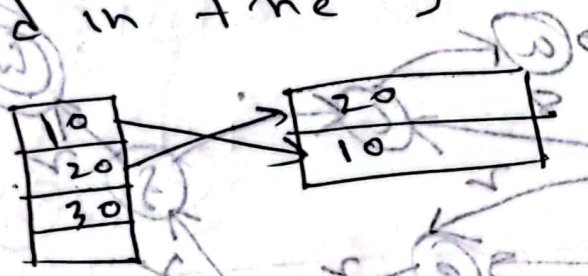
$R = \{A, B, C, D, E\}$

$R_1 = \{A, B\}$, $R_2 = \{B, C, D\}$, $R_3 = \{D, E\}$

* A secondary index is always a dense index

→ Ordered indices whose search key specifies an order different from the sequential order of the file are called 2ndary indices

or non-clustering indices.
2ndary index must be dense, with an index entry for every search key value and a pointer to every record in the file.



(unordered)

* Search key:

An attribute or set of attributes used to look up records in a file/ disk block/ pages is called as search key.

* Record

An index record consists of a search key value and pointers to one or more data records with that value as their search-key value.