

Dbma

Hr = human resources.

SAL commands.

SELECT * from employees — ~~SQL command~~

SELECT ^{(for into} * from employees — ~~object browser~~

Select
" 183 of."
word logical clause
Selected from where
most share

↓
selected clause

from "
()
limit

statement
select attributes
from table
where condition

~~order by~~
~~top n no~~
~~(of) value~~

DBMS - theory

Database User.

• Naive User.

• Application developer.

• Sophisticated user.

• Specialized



Types of database

• Hierarchical database

• network "

• relational "

• Object oriented database

• object relational "

~~Int. user~~

• End user database

• operational database



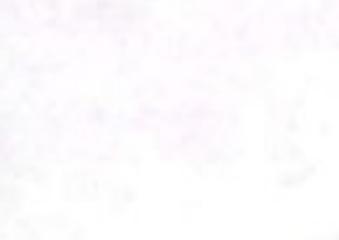
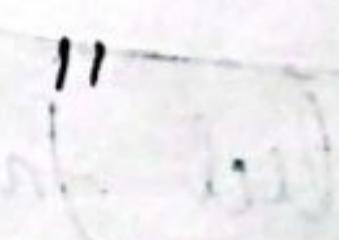
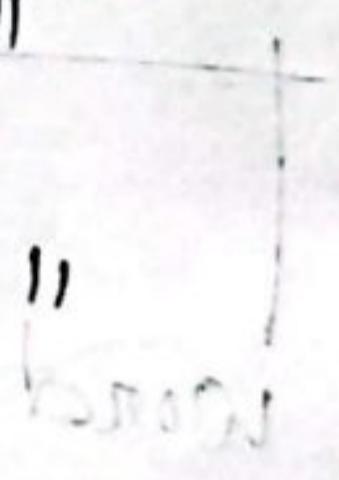
graph database

No SQL "

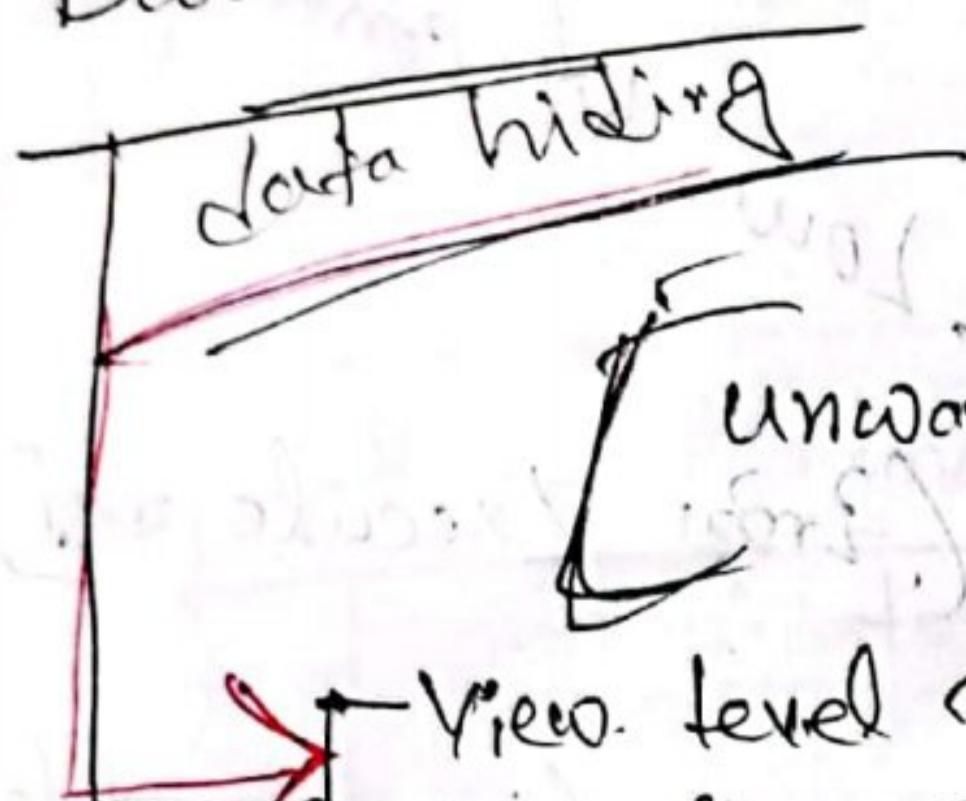
• Mobile "

• Spatial "

• Temporal "



Data abstraction: unwanted data hide concept.



Ex: (fb profile only me) (Public).

unwanted data can be hidden using this

View level → minimum data part

logical → data part of user

physical → structure part - Developer

Like
like data part
change
change

high lvl abstraction
hide or
not (part)

DDL = Data Definition language.

DML = Data manipulation language.

Data files: which store the data base itself.

Data dictionary: which store metadata about the structure of the database.

Abstraction EXP

view → sara ja de

logical → jamaal bhi

physical → developer (2) software

Theory

functional components

Query processor → DDL Interpreter (Can Execute DDL)

→ DML Compiler (Line by line translate [SQL])

→ Query Evaluation Engine

Storage

Query manager

authorization and integrity manager

(write to access)

(check appropriate rights).

2 → Transaction manager

(-তে একটি অঙ্গ যোগ করো)

3 → Buffer manager

(Temporal memory)

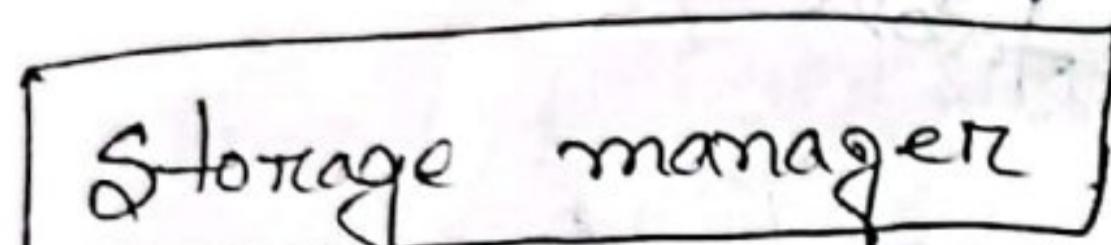
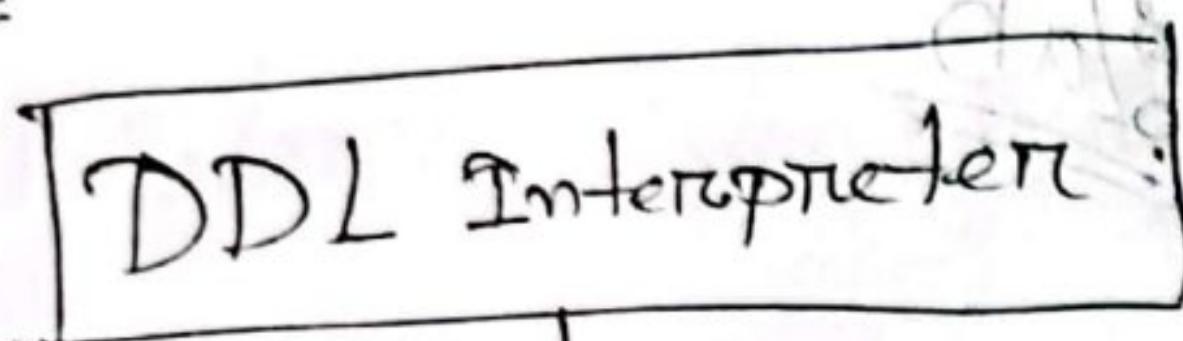
TRU - Data memory (OTC)

(স্থায়ী রাখ)

4 → File manager

(file extension স্থিত করো ফাইল (ুক্য))

#



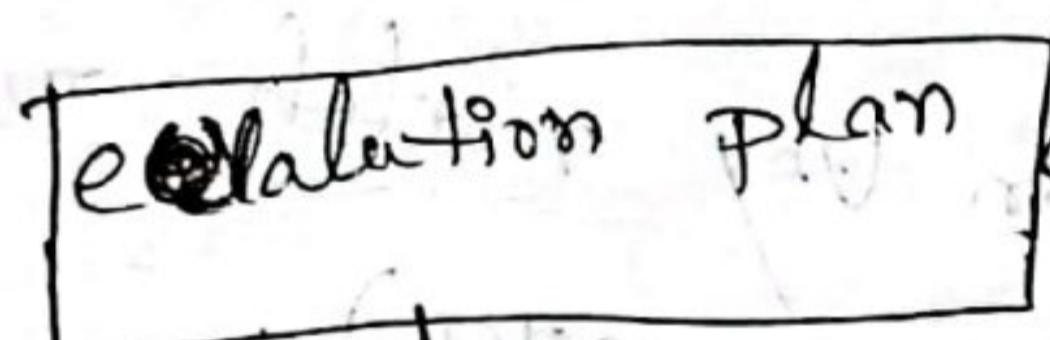
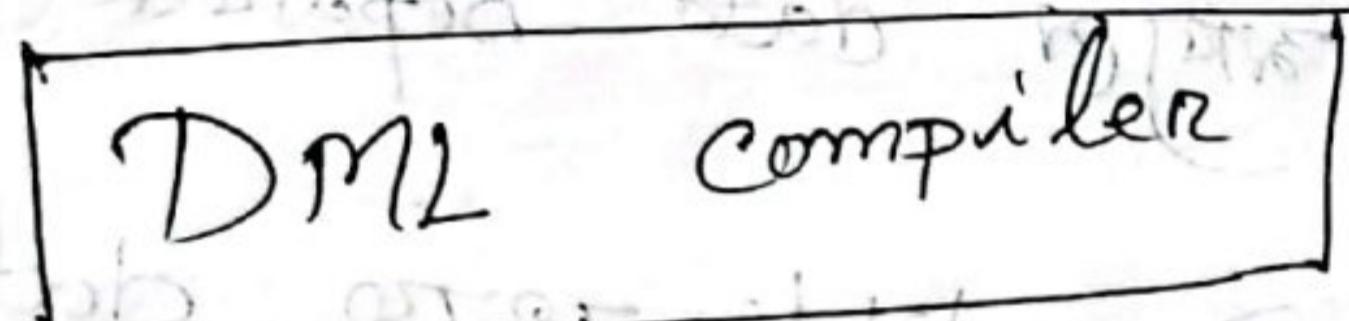
authorization and integrity manager

transaction manager

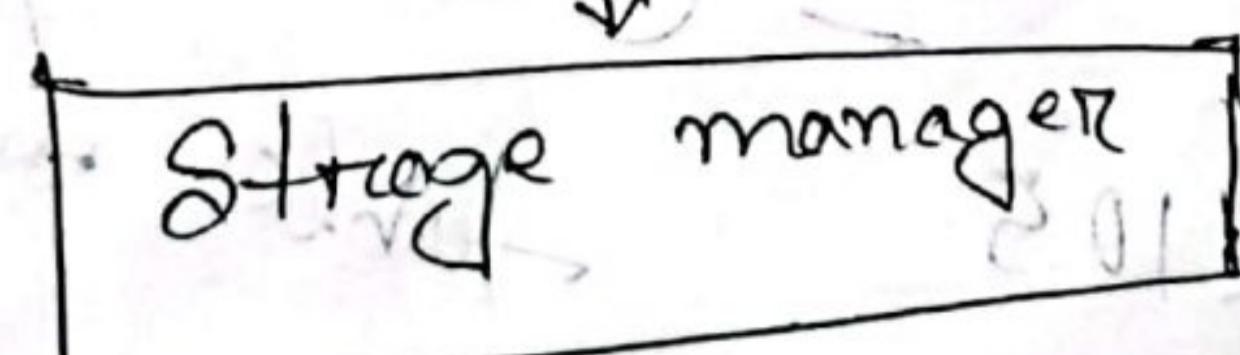
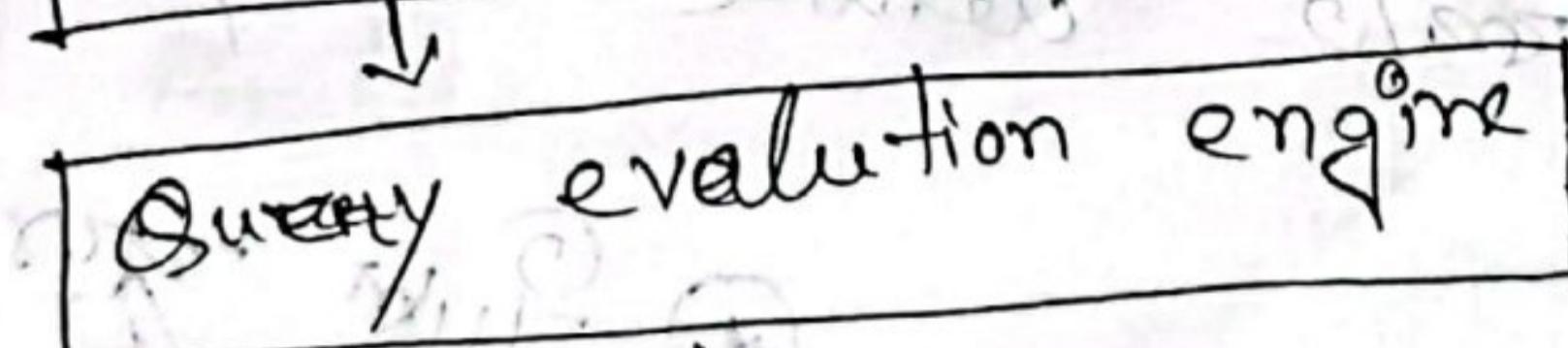
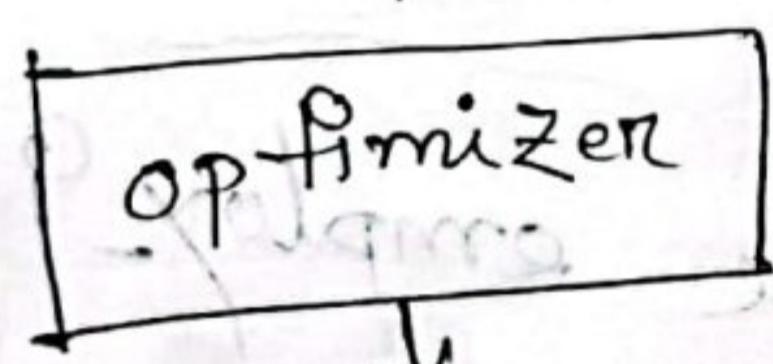
buffer manager

file manager

#



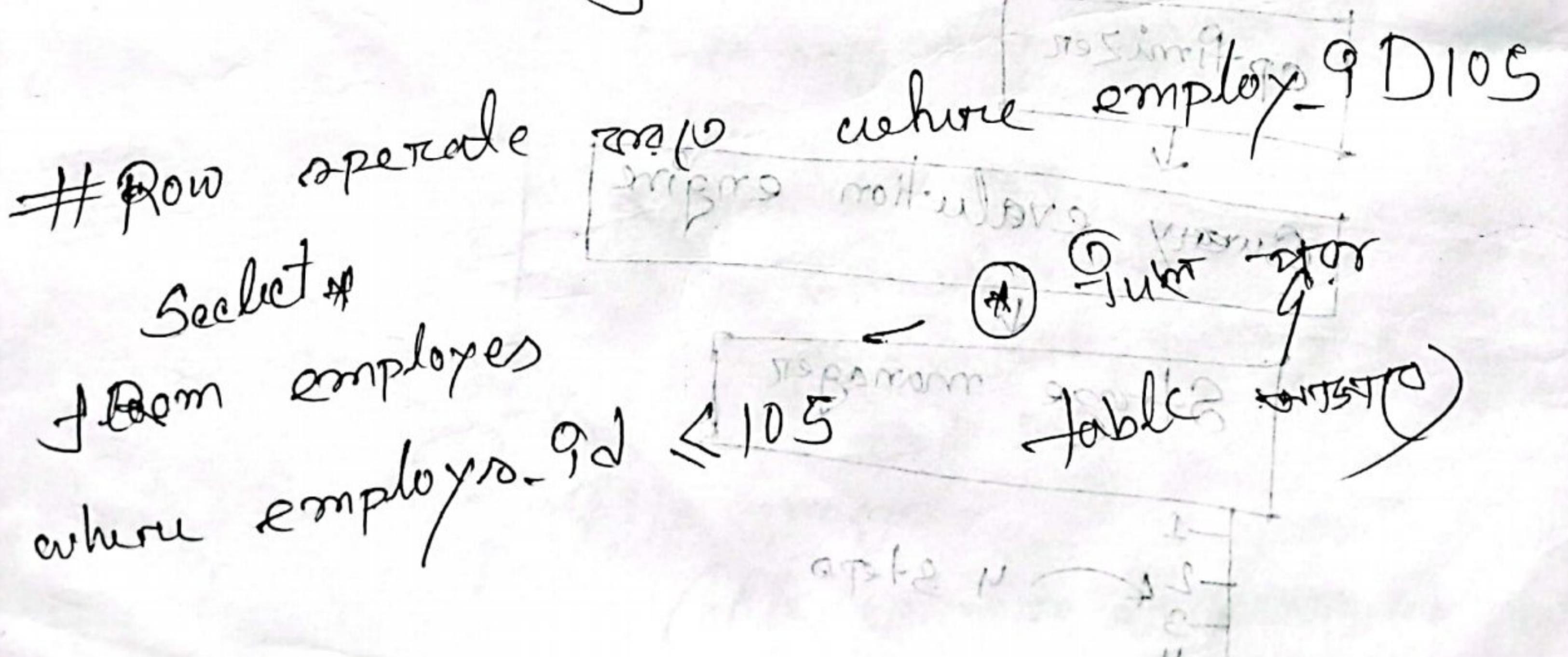
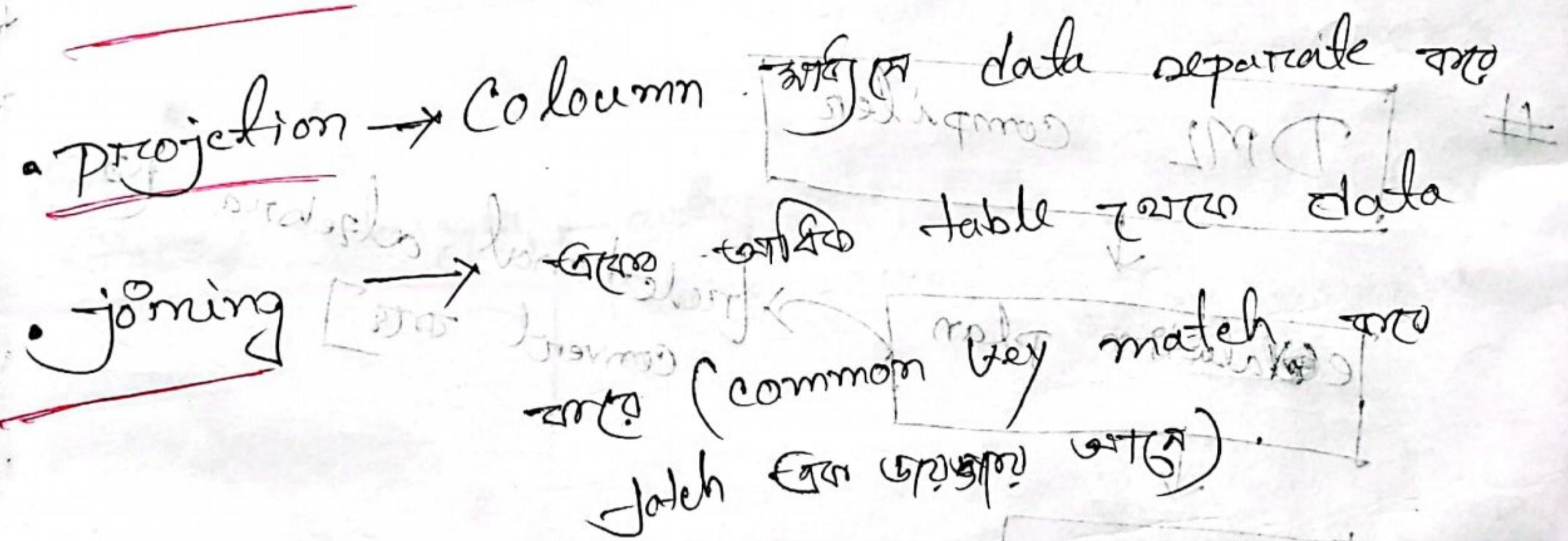
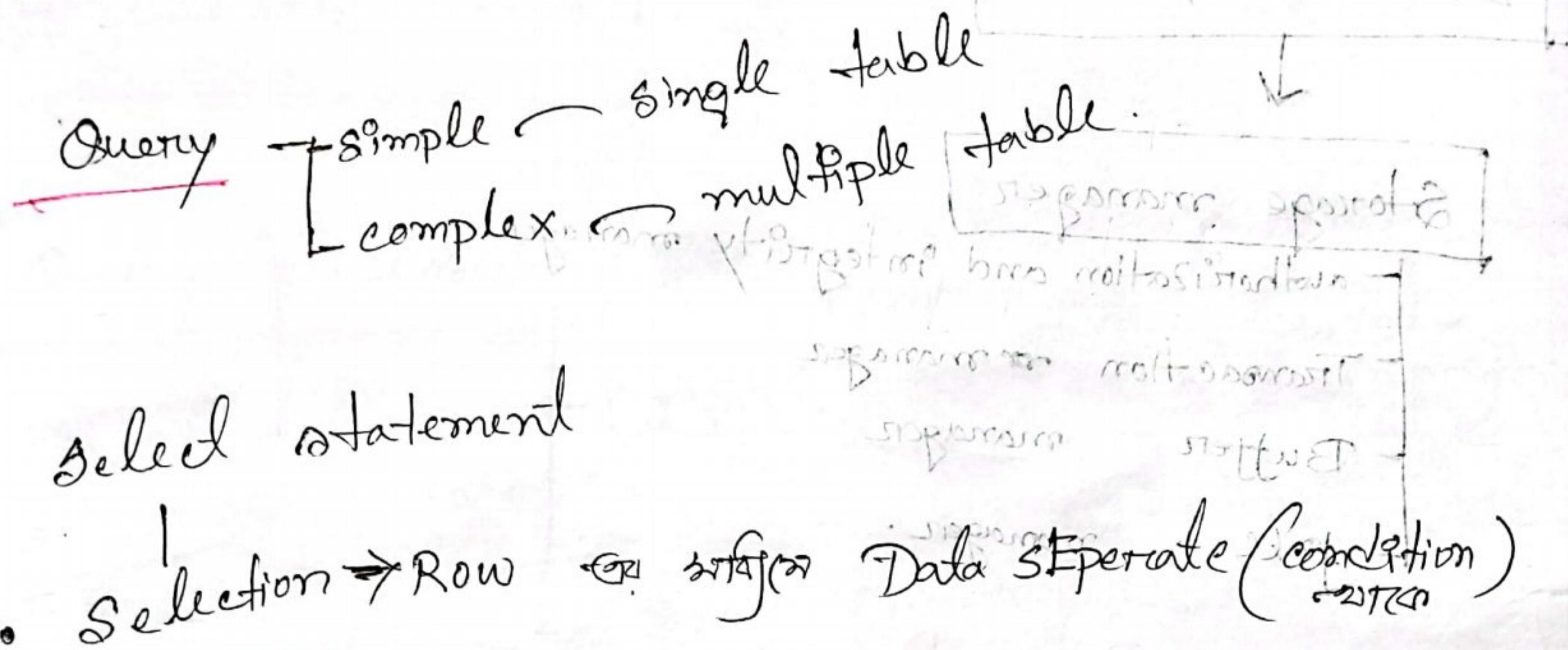
[relational algebra
convert SQL]



1
2
3
4 steps

Simple note of using oracle lab

at simple ICP



Salary, salary \Rightarrow 12 \rightarrow salary - (new)

Employee-ID (company ID) / as company ID
Space (from) Rename - 25

π - Project } relational algebra
- Selection } symbol (subset operation)

π employee-ID, first name
from employees \rightarrow (employees) \rightarrow from 20.

Relational algebra

SQL

Select A₁, A₂ π T₁ A₁, A₂

from T₁ T₂

where condition \rightarrow T₁ \times T₂ ... \rightarrow condition
where

operation used on oracle (lab)

• Between ~~.....~~ and [where Employee-ID between
105 and 109 ↪ codes.
output ↪ (105 thru 109 show একটা)]

• IN (set of value) [where employee-ID IN (103,
105, 108) ↪ ২৩ টি (১) এর মধ্যে]

• Is Null [where 105 is null
↳ প্রতিটি null এর মধ্যে
show একটা]

• Like [pattern matching করে]
['_a.%'] → ২য় word a Show একটা
['%_a.'] → আগে একটি একটি (a)
দ্বারা Show একটা

WHERE FIRST

১০ (ii) ফিলে প্রয়োগ করা এবং

concat. এরা (২) same
করা হয়।

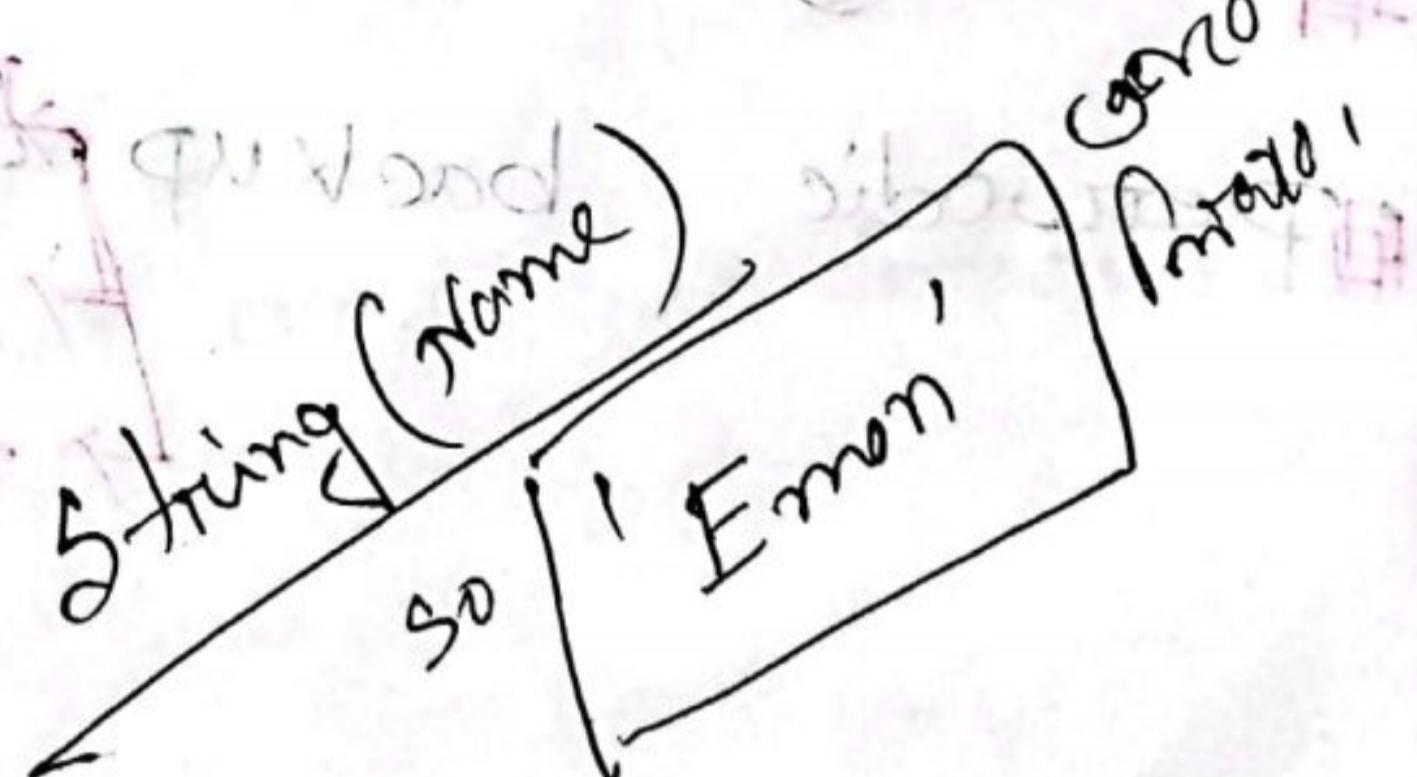
Select first_name || last_name as full_name
from employees;

How to create table:

Create table batch52

for create table

(
id int unique
name varchar(20)
age int
)



insert into batch52 (id, name, age) values (51, 'emon', 21)
insert into batch52 (id, name, age) values (52, 'Emon', 21)

↑
for insert data in table.

DBMS Theory

Database administrator (DBA)

▪ Schema definition → overall [design] of Database

▪ Storage structure and access method

[পরিবেশ করে নথ্যে ও অস্ত্র]

definition.

▪ physical structure modification.

↳ [Upgrade করে নতুন ফাক্টরি ওয়ালি]

[করে রাখে]

▪ Authorization [করে করে পেটে নে করে]

▪ Periodic backup

[data backup করে]

↳ maintenance কর, আবার সেভ করে

↳ memory management, file management

[করে]

</div

Data model

Entity = important thing

What is data model?

is a logical diagram represents database.

Two types of data model are used worldwide

- Schema diagram
- Entity-relationship diagram

Necessity of Data model?

Design database using 5 things

- 1 Application Area or Domain, and sub domain.
- 2 sub domain.
- 3 identify entity sets.
- 4 identify the attribute.
- 5 identify the relationship.

constants

জীবন প্রক্রিয়া

মানুষের প্রক্রিয়া

যদি কোনো প্রক্রিয়া

live CCPA এর constant

DBMS Lab

• Select

- **DISTINCT**: Select DISTINCT First - Name

কোনো ক্ষেত্রে এই পদ্ধতি
ব্যবহার করা হবে।
ব্যবহার করা Show করা

repeat value avoid করা

- **DISTINCT** (keyword) এর অঙ্গে count (function)
use করলে সবুজ data এর Show করা।

Select count (DISTINCT) JOB-ID)

= Count & Distinct space

- কোনো কিংবা এবং Just একটি word Show করা।
Syntax - [JOB ID এর 'IT' Job এর লাগানো Show]

Select *

FROM employees

where job-ID Like '%IT%'

এই পথে ফিল হলে এই আগে পড়ে
ফিল এর মাঝে Show

• Function : ORA සේ ඔවුන් () මගින් විවෘත කළ

• Single - row function → පෙනීම් තුළ අඟු නැඟු කළ බඳු
- පෙනීම් තුළ විට return කළ

• Multi - row function : → මෙම multi row පිළි දායු බඳු
- multi row return කළ,

HYPES

a. character manipulation function [character පිළි තුදා
කාඟ ඇති].

case manipulation

upper

lower

replace

instead

character manipulation

length

LPED

RPED

substr

initr

b. Number function

round

trunc

mod

c. Date function

month between

sys_date

last_day

d. conversion function

to character

to number

e. Special function / general purpose function.

DECODE

case

NVL

(extract) position

length

initial

last

middle

right

trim

ltrim

rtrim

lpad

rpad

ltrim

rtrim

lpad

<

* Junction গুরুত্বে () ক্ষেত্র এবং সূচী

Select upper (first name)

SD ক্ষেত্রে ক্ষেত্র

Select upper ('true')

From dual

Select upper (:Input text)

From dual

ক্ষেত্রে Input টিকে হাতে show

init cap এবং ক্ষেত্র ক্ষেত্র word ক্ষেত্রে ক্ষেত্র

Select initcap (:Input text)

From dual

initcap এর কাজ 1st word র প্রথম বর্ণ -
Select *
from ~~other~~ employee;

where first_name = initcap('lex')

where lower(first_name) = lower(:input)

where lower(first_name) = lower('daniel')

lower মেরা কি
first_name কিরুৎ হতে চাহিএ

ক্ষণের মধ্যে পরিবর্তন

প্রথম
বর্ণ
ক্ষণের
চেয়ে

character manipulation

concat function, (II) concat operator

joining some two columns or word

Select concat (First_Name, [' ', last_name])
from employees

Run SQL → Name - Name

space for

→ Select first_name || ' ' || last_name

Some output

find the employee having 'Abel'

Select * from employees.

Where first_name || ' ' || last_name like '%Abel%'

↓ last name will also will come out like

'(Abel)' will give 6/A (6 is 2)

#Length: '918' - (p, hi, dot) introduce which.

~~6021~~ Select length (first-name).

→ mirror earth wind show zero

LPAD

Select LPAD ('Selby', 10, '#')

Substr

~~Substr(first-name,1,3) || Substr(last-name,1,3)~~

Selected Substrates (final name 1,5)

Q Name one word for 'con'

1,5 ~~UWTA~~

1 (25cm) 5 cm word allow
অন্তর

- Where Substr(Job-id, 4) = 'REP';

মাত্র Jobid এর ৪'র Rep আকাগুলা শব্দের মধ্যে

Numerical function:

Round

[Data base expect result]

সংখ্যাক মান

দশমিক

এক সংখ্যা
যোগীতা

TAKI

Select round (32.45, 1)

From dual ← dummy table . round figure

Trunc (commission per cent) commission.

(Figure trunc ফাংকশনটি এক পঁয় এবং 1st word
করে ধরে)

Mod (Salary, 43)

from employ.

mod form [43]

নতুন salary এর 43 টাকা

অণুক্রম

অঙ্কুর অঞ্চল

প্রতি

কোর্স বিষয় কোর্স কোর্স

DBMS Theory

E.R. diagram.

An Entity relationship diagram.

unripe for use

~~Strong~~

Wear

~~Entity set~~

attribute

~~Explanation~~

primary key

unique

Q (or
underline
line
matrix ID
type)

multivalued attribute

A₁

\bar{A}_2

composite attribute

~~derived~~

attribute

- It has been no exist in database.

\mathbb{J}, N

Name _____

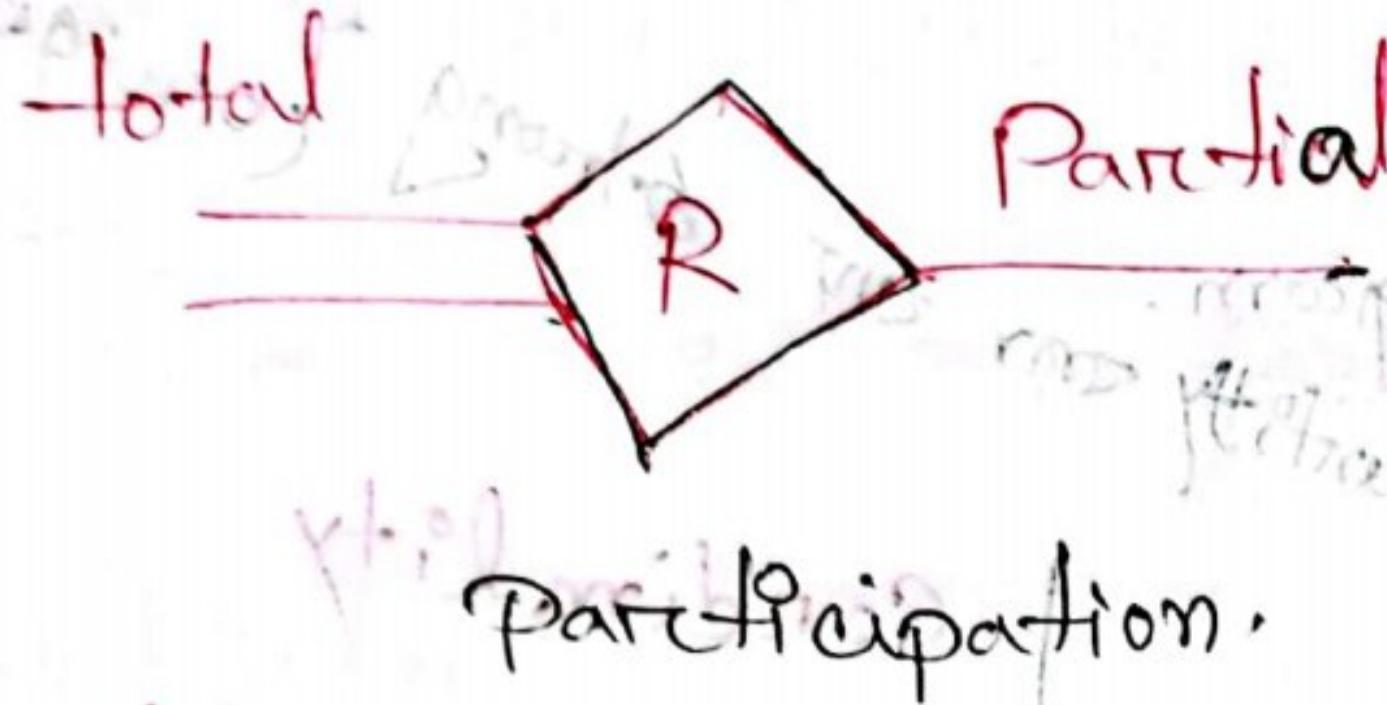
L. Name

~~বাবা~~ ~~পুরুষ~~ ~~পুরুষ~~ ~~পুরুষ~~ ~~পুরুষ~~

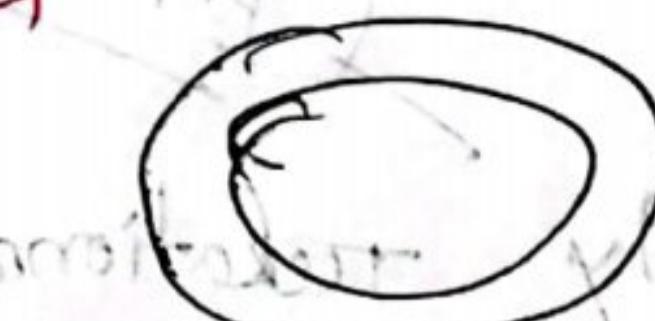
~~বিবরণ ও অন্যান্য পদক্ষেপ~~

—مَا يَعْلَمُونَ

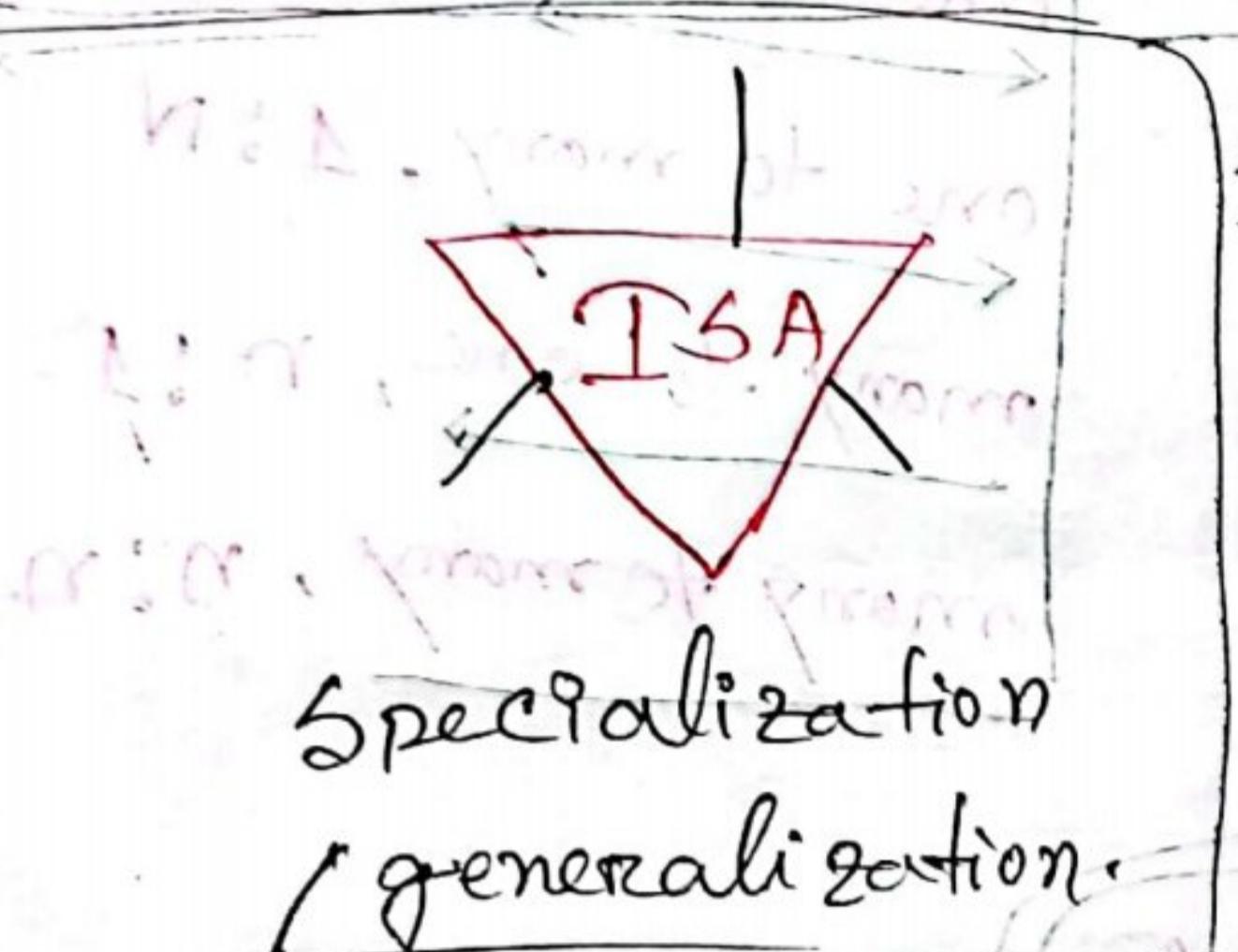
like age west
date of Birth (or)



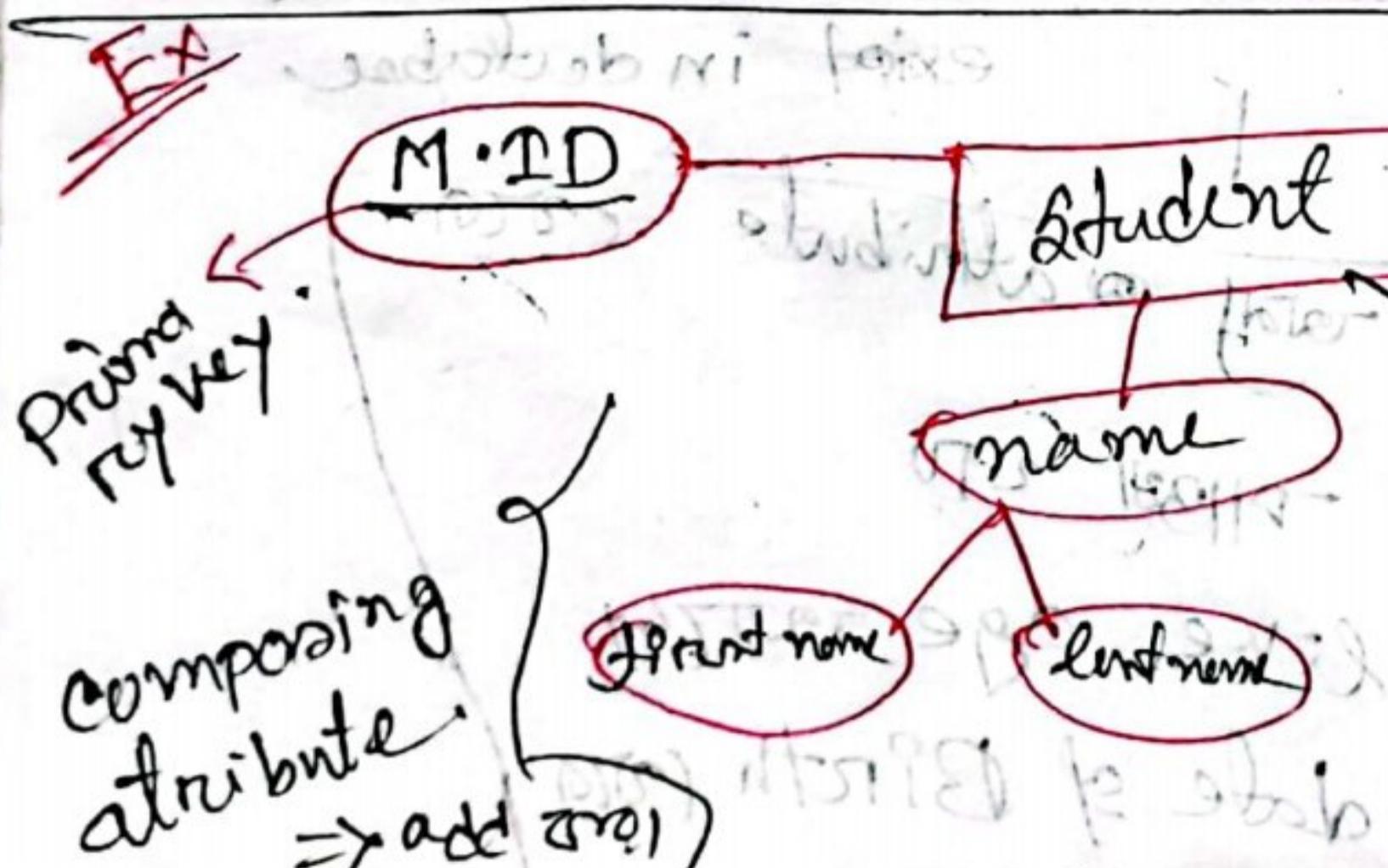
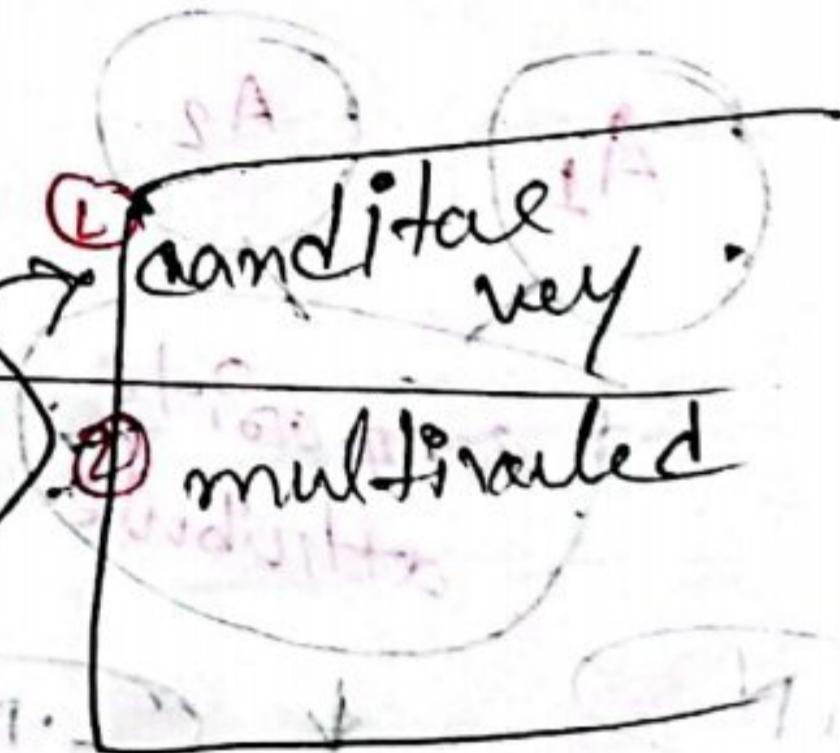
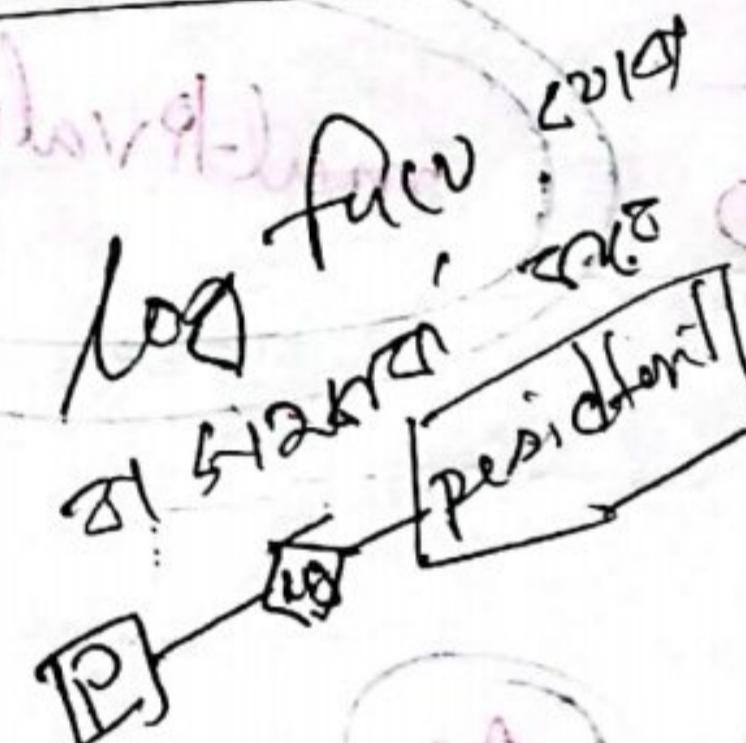
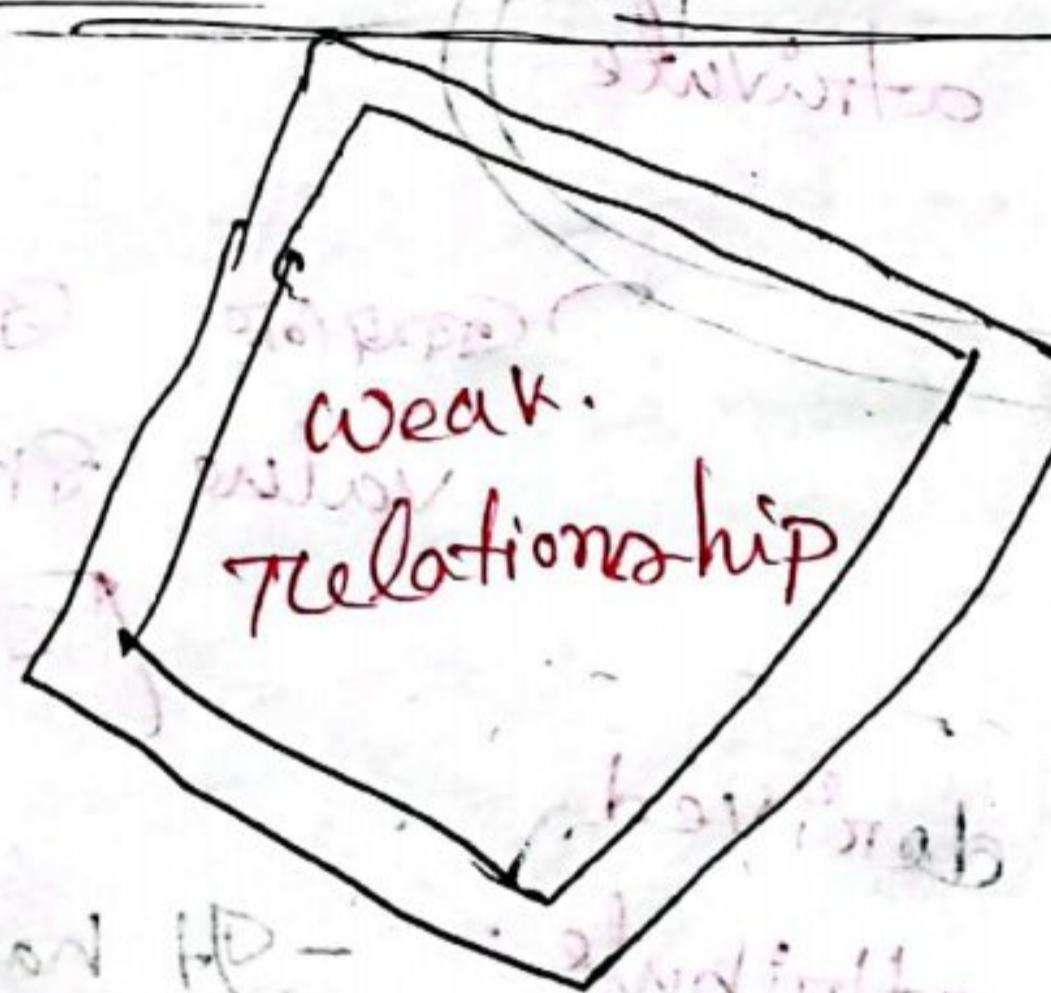
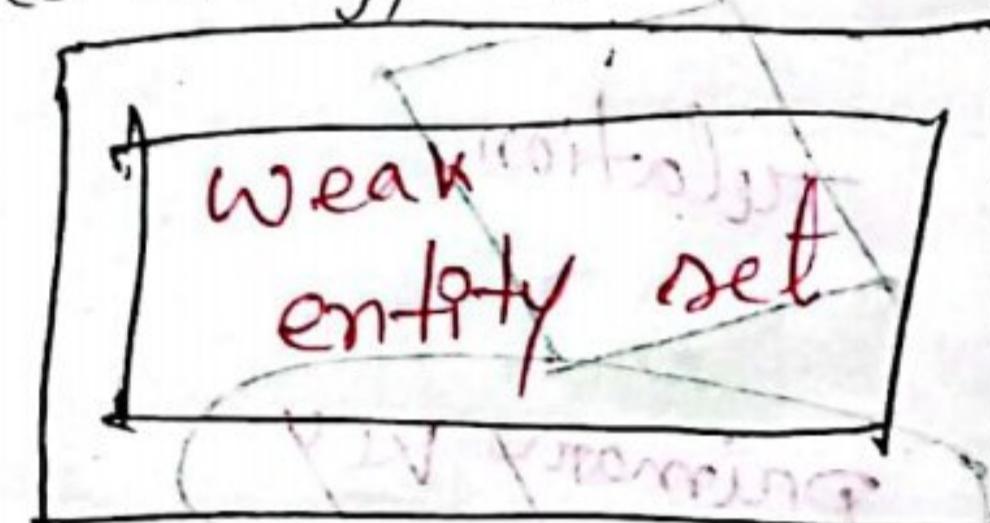
Primary key
foreign key



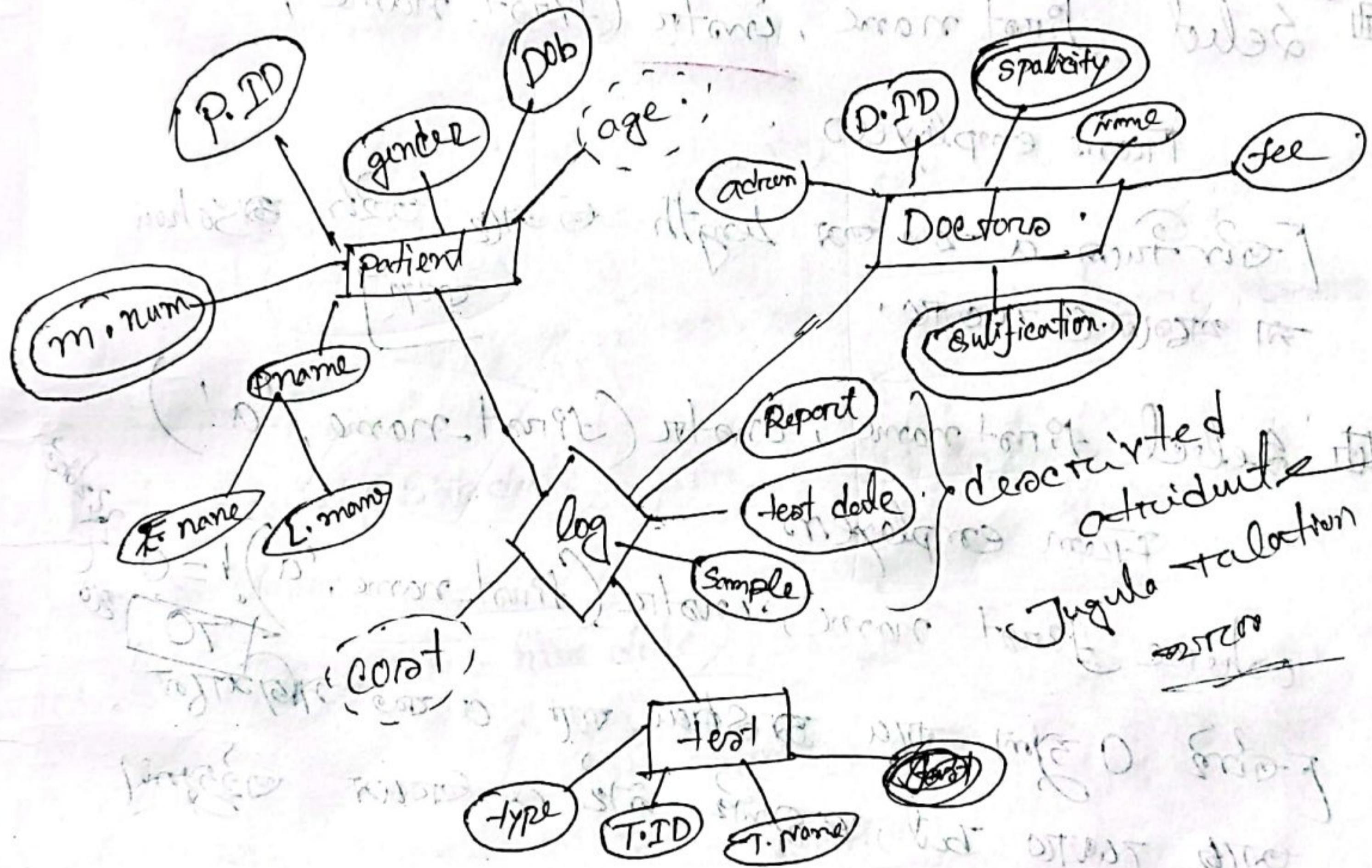
candidate key



2 entities must have separate
to identify each entity, like
payment, loan



Hospital delta base



Noun entity
verb relation

DBMS Lab

instr

Select first name , instr (first_name, 'a')

From employees

[Return a row or length of the string somehow
in character set]

Select first name , instr (first_name, 'a')

From employees

where first_name , instr (first_name, 'a') != 0

[Find 0 char string to show that, a is a character
and two but one character string]

Or
0 or 1 character.

Date function

sysdate

Select sysdate } returns date type
from dual . }

• Select Syadate - hire-date

from employee

For 6 months job करो (6 * 30)

• Select round((Syadate - hire-date)/365)

round some result will
be 6 * 30 = 180, 180, round figure तो
180 नहीं

• Select round((Syadate - hire date)/365)

from employees

where round((Syadate - hire date)/365) > 30

* 12 फिर
मासिक वर्षो

[१८० में से ३०]

180, round करो, 6 * 30

, salary, salary + 500

निम्न 30 वर्षो का अपना बायो जीवन

500 (2151-270)

#Dead function

Book 22 page (3 chapter)

#Select round (months between (syodate, hiredate))

ফিলে ছাঁট show করো

1/12 রাতের বায়ু রেড

Show রেড

Next date

Next day (syodate, ~~friday~~ Friday),

ফিলে তাপমাত্রা date করে ফের করো

date করে করো Show করো

last day

Last day (hire-date)

hire day to last day

add months. (00:00' + value) works. IT IS 2024

Selected add months (Syndate, 3)

00:00:00 + 00:00:00
00:00:00 [0:0]

2024-01-01 00:00:00
00:00:00 dateformat

THIS IS HOW YOU ADD NEW BIG RIGT RIGT DATE PL 1000

Conversion

fromdate todate / noformat modul

To-Char (Syndate, 'Jm DD MM YYYY' Day)

fromdate

[0001-01-01] todate & format

YYYYDDMM

format use 2024

[DDDD]

00021 month name of maximum

[from 2024-01-01]

use take 00021 month

00021 < (IS (maximum * (value + value)) month

#Select To char (Salary, '99,999.00')

From employees

(E. abaya) instance
gross to pay salary [.00]
TAXES DED
[0.9] remain

NULL एवं मात्र अपरिवर्तनीय होता करना NULL देखा जाए

nvl (commission_pct, salary)



Select , salary + salary * nvl (commission_pct, 0)

From employees.

gross commission (२५%) इसे एक संख्या के रूप में लिखें

gross salary (१८८०)

#find the employees whose salary with

commission is more than 15000.

Select salary
from employees

salary > 20,000
c2, emp_id (1 वाले)

where (salary + salary * commission_pct) > 15000

NVL2, COALESCE, CASE, Decode.

NVL2

Select salary, salary + salary * NVL2 (compl, 0, 0)

NVL2 (① 2nd word vi filo or
complimented to show reso
② word ③ ④ Null value
④ 7uano

① or ② compl - 7uano
③ reso
④ 2nd vi filo or
⑤ 7uano, ⑥ 7uano

COALESCE : when first not null - vi filo show reso

Select employee id, first name, last name.
coalesce (first name, last name) as first not null name
from employees.

CASE: Select first name, job id,
case job id when 'IT PROG' then salary +
salary * 0.1.

else salary End Received Salary
running.

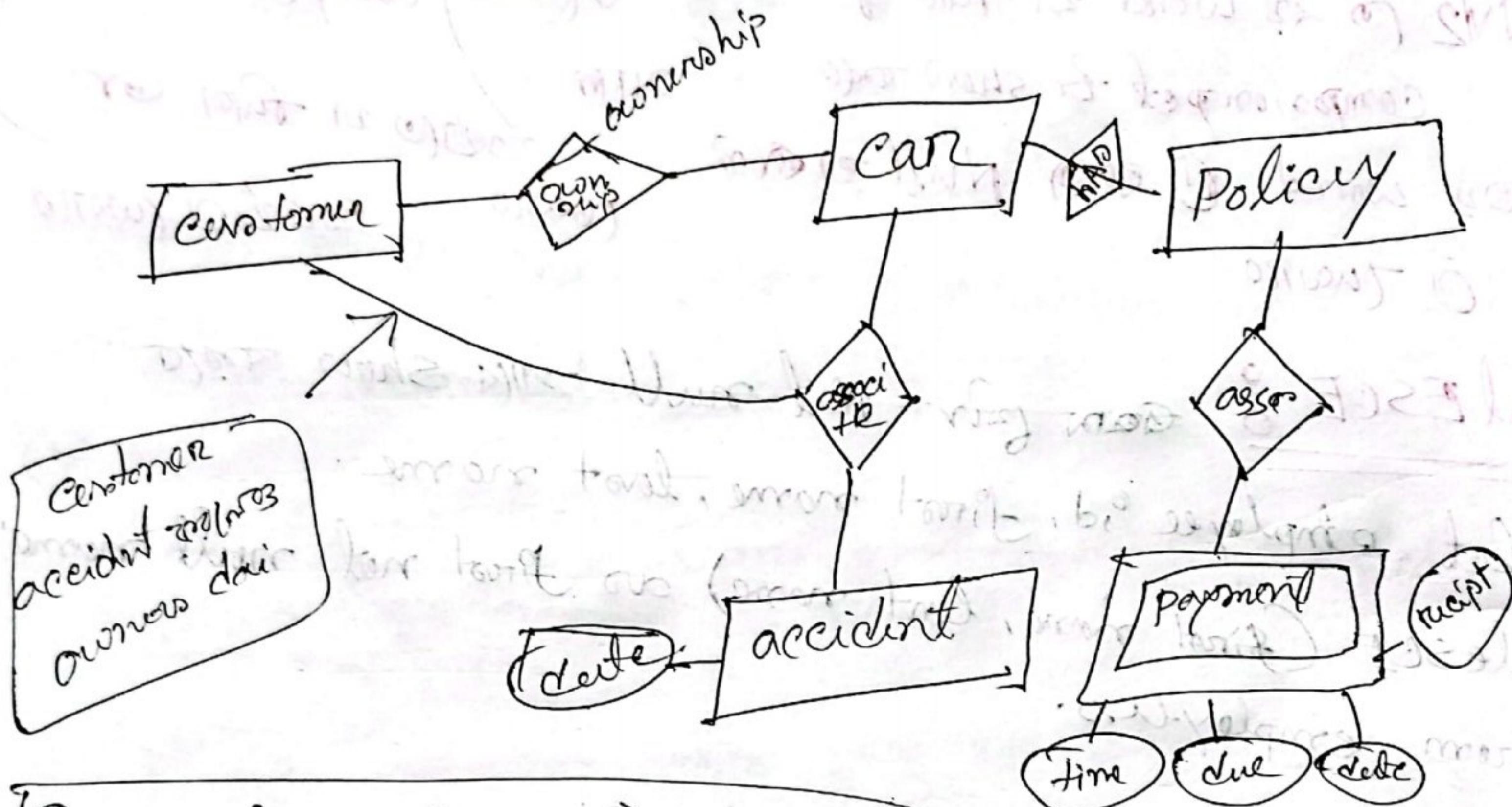
If this do this

case job id when 'IT PROG' then salary * 0.1

case vi filo or 7uano Show and

DBMS theory

car accident policy

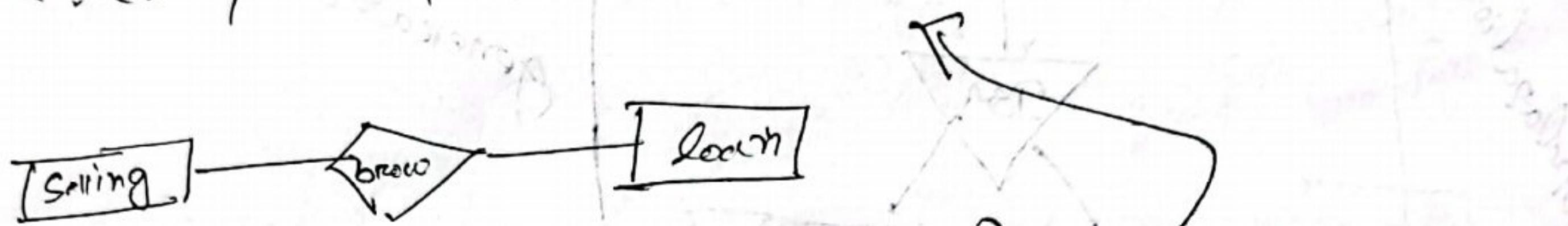


① payment
weak entity

Policy is entity and payment (defend weak entity)

Participation constraint: କାହା ଥିଲେ କଣ୍ଠେ ଯାଇବା ପାଇଁ

କେବଳ କଣ୍ଠେ participation constraint ଏବଂ

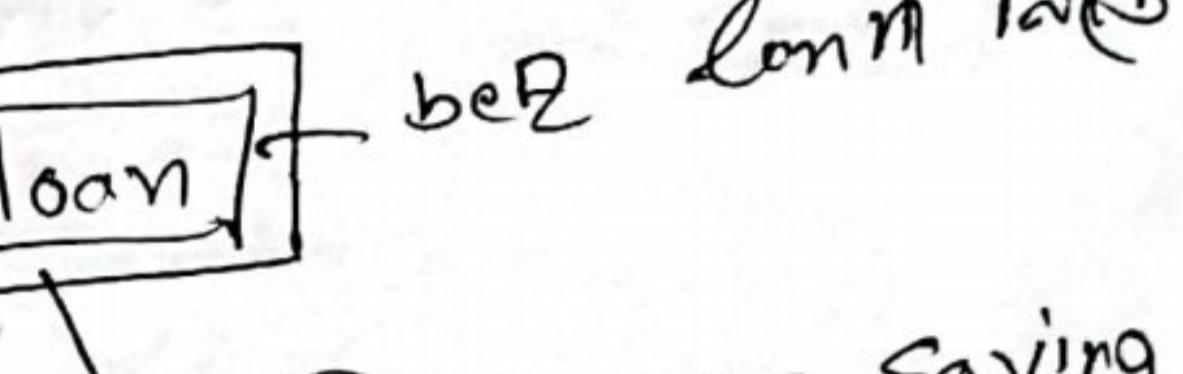
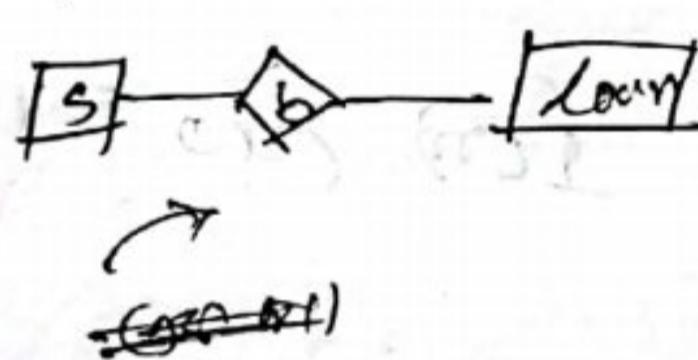


loan acc କୁଟିଛି saving ନାହାଦୋ ଏବଂ କାହା
loan & buy

• too-fypo

• total.

• partial



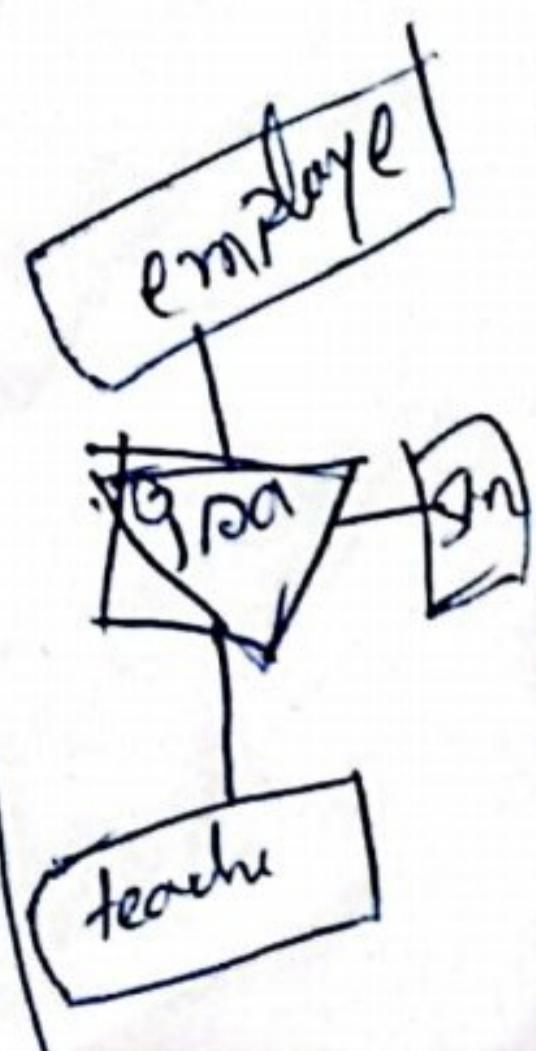
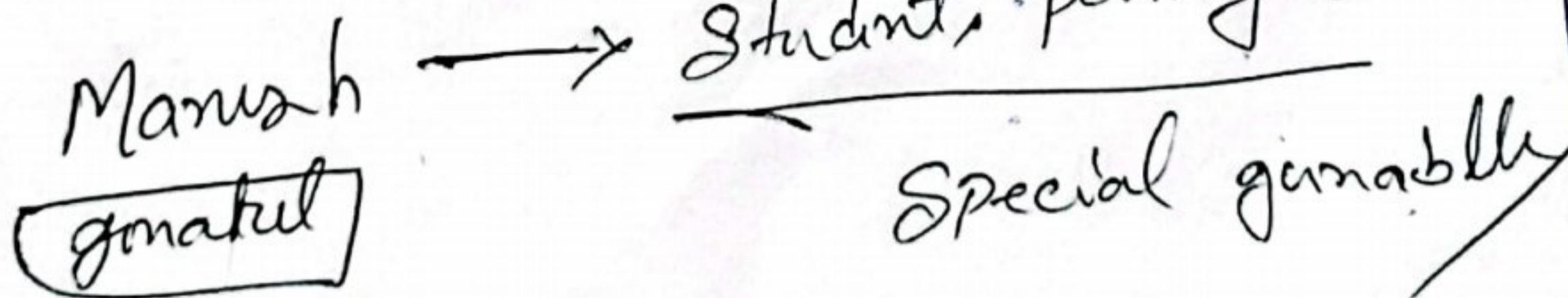
• loan କୁଟିଛି Saving
• ମିଶ୍ର ଫଳାଫଳ ନାହିଁ
• ତାହା କୁଟିଛି weak entity.

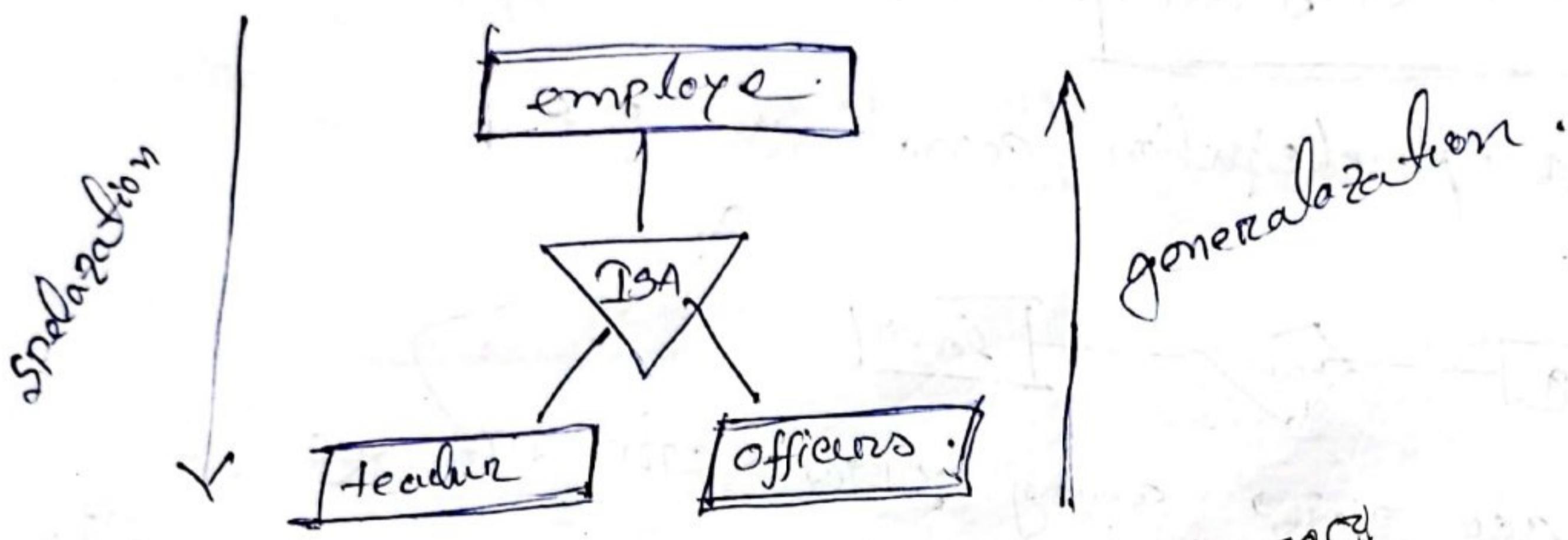
Specialization and generalization

• ଏହି process କିମ୍ବା general group କେବଳ special grp

• ଏହି attribute କୁଟିଛି special grp convert କରିବାକୁ

Specialization





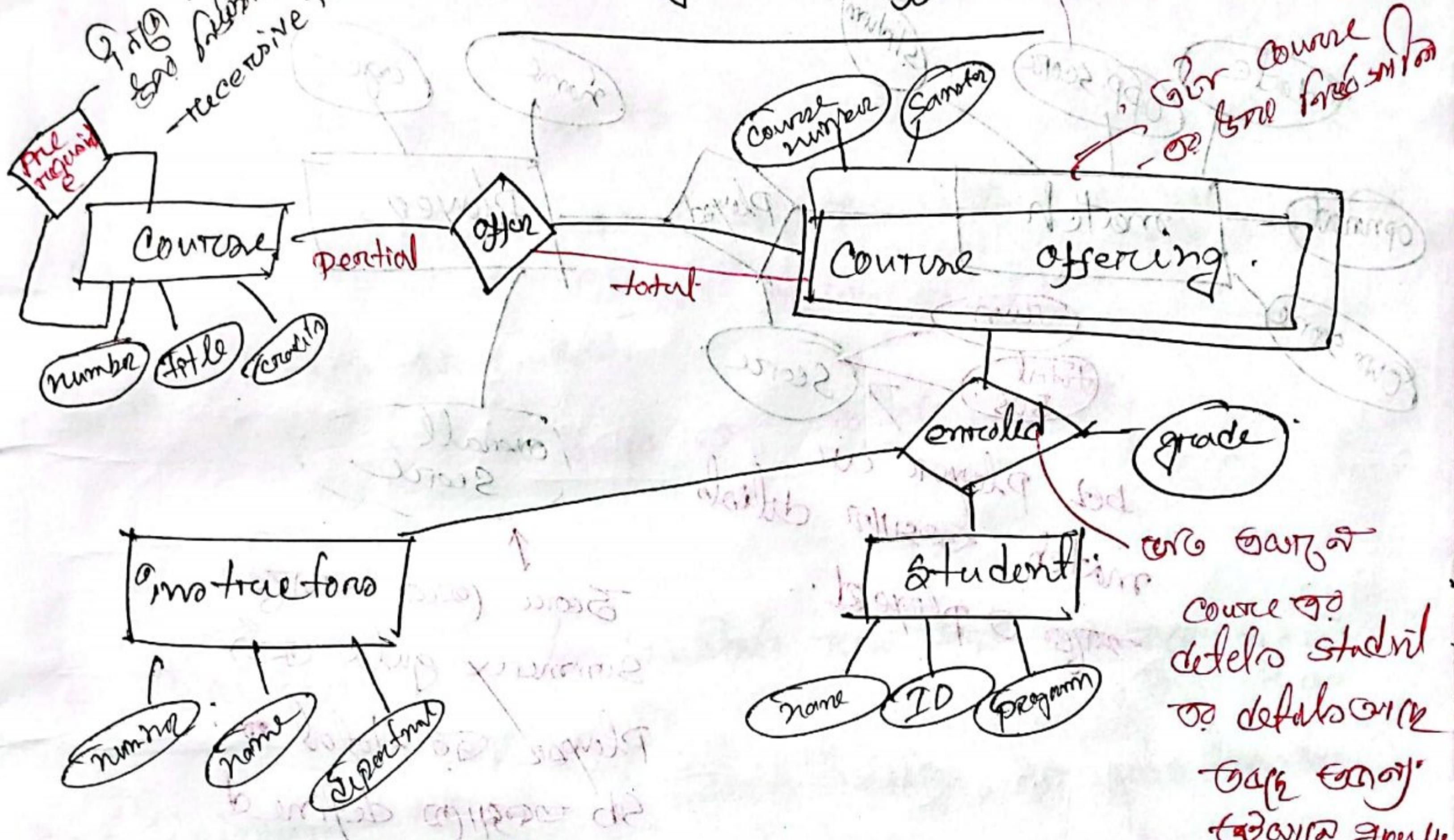
- user defined
- condition defined
- Disjoint for particular time
- overlapping

DBMS theory

Q18
Explain what is
recursivve relation.

Ans: If A is a relation
then $A \times A$ is also a relation.

The Registrar's office



Q19) Explain the difference between primary key and composite key.

Ans: Primary key is a single attribute or group of attributes which uniquely identifies a record in a database table.

Composite key is a group of attributes which uniquely identifies a record in a database table.

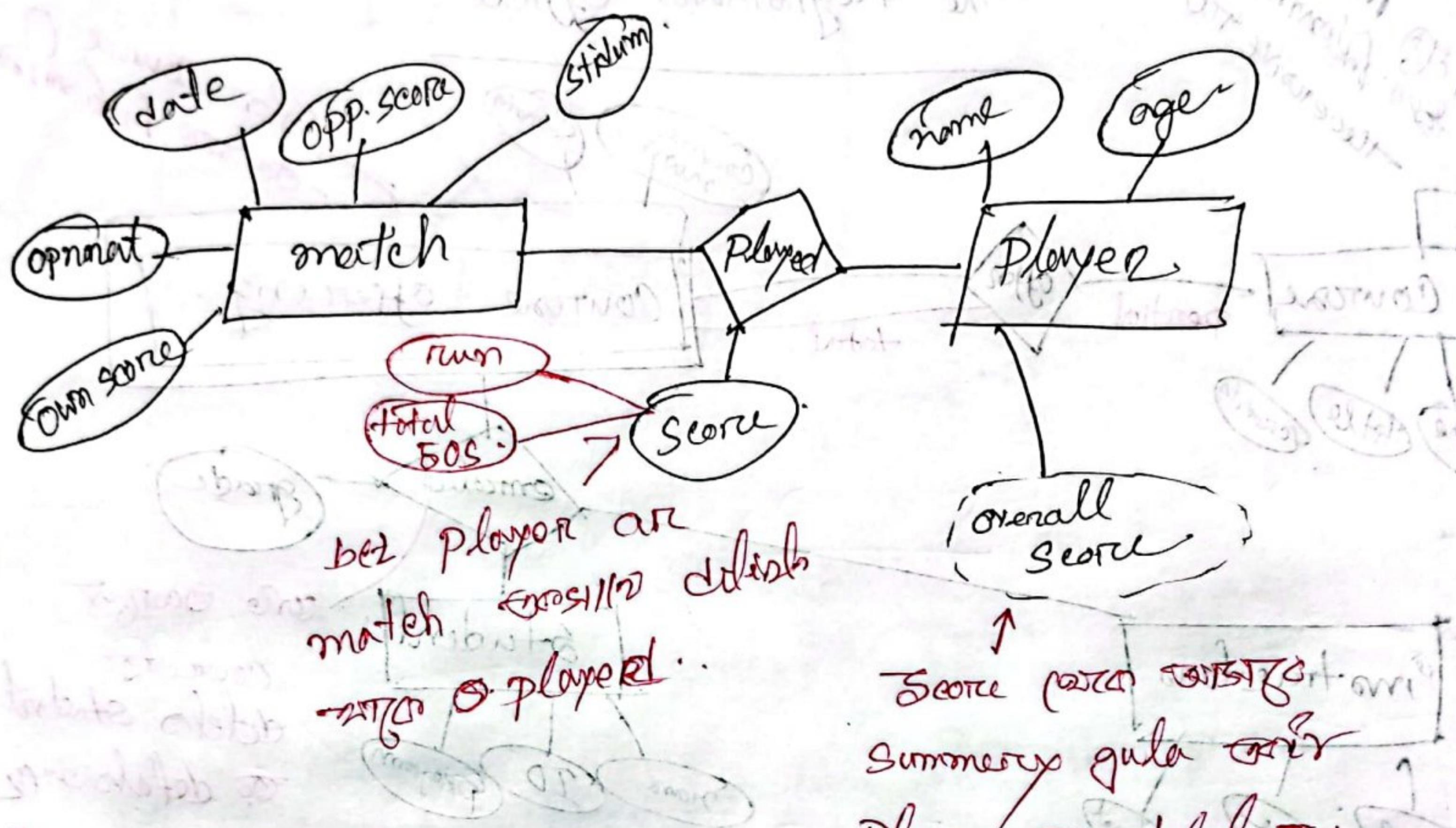
Ans: Primary key is a single attribute or group of attributes which uniquely identifies a record in a database table.

Composite key is a group of attributes which uniquely identifies a record in a database table.

Ans: Primary key is a single attribute or group of attributes which uniquely identifies a record in a database table.

Composite key is a group of attributes which uniquely identifies a record in a database table.

Match diagram



→ mru(g)

DBMS Lab

Ex - find job id Gr(1) Gov show report Case case and
accidental query

selected first name, job id

case job id when 'AD Pres' then AD-Doctor group
Job minimum a group

group by job id sum needed,

21 sum or, group function (are)

Multicolumn function: multiple rows from one row generate one result.

• Minimum : Min (Group function, Aggregate function)

- select min(salary), max(salary), sum(salary)

• Maximum - Max

sum (CPI), works with

• Sum - sum

same column to average - libarated

• average - Avg

count (ID) - count

Group function

Select Job-ID, (Salary)sum, Max(Salary), Min(Salary)
from emploess.

group by Job-ID

↳ Group by Job-ID

→ when multiple row तो Group by Job-ID

then

Q2 जैसा क्षेत्र

having Job-ID IN 'IT PROG'

Individual for cuarto (where function वाला हो या तो cuarto)

live where Job-ID > 110

or वार वार

Individual cuarto वा -cuarto रिट

Having Job-ID IN "पुरुष"

having = where
some data

Multi-table join Data Search

- common field exist in both tables
- join multiple tables, start from primary key

Select *

From employees, department

where e. Department-Id = d. department-Id

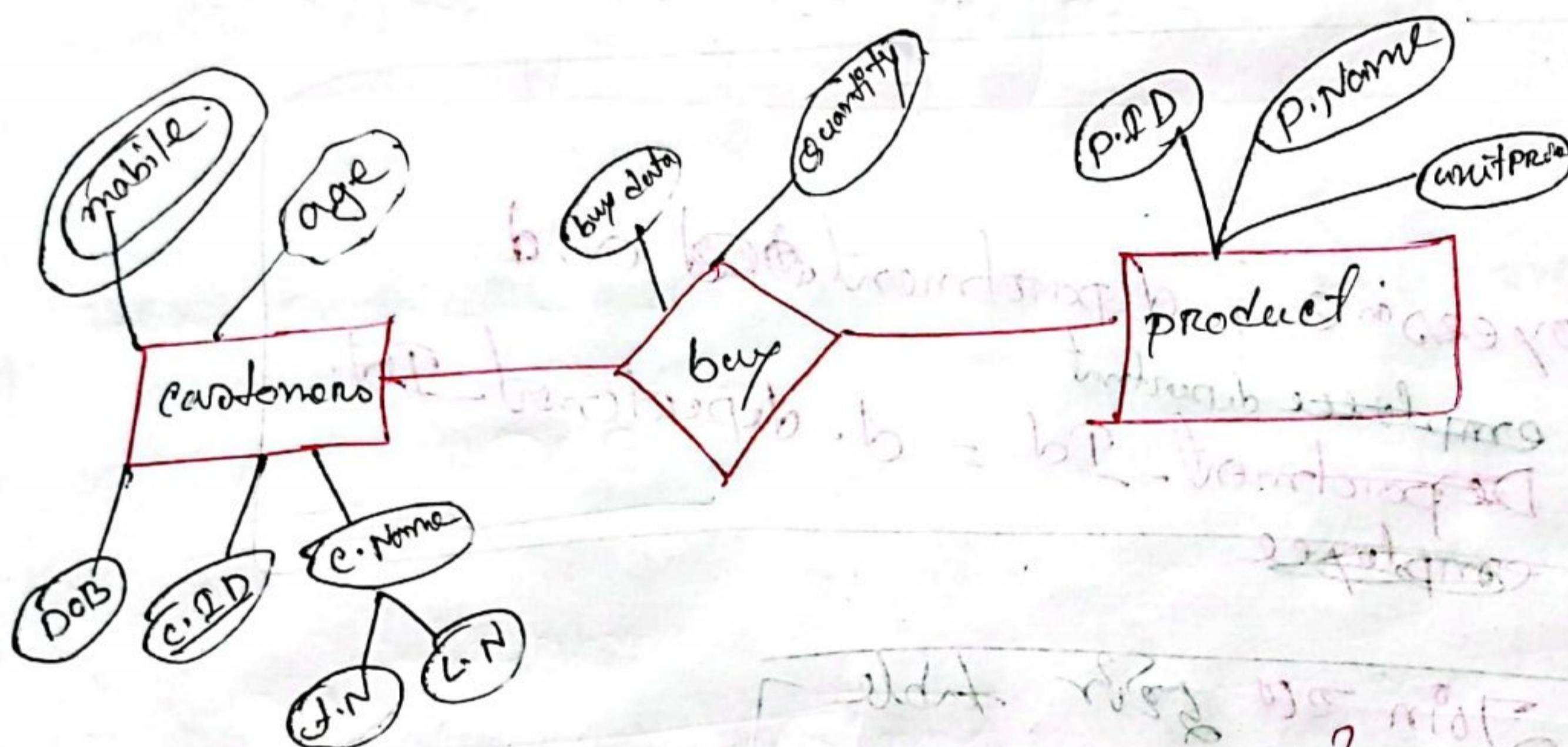
Join 2nd table

DBMS Theory

III converting

Grid to Schema

End to delectable.



III Entity and Relation ^{with table over mstg.}

III Entity more then expressive to schema. Important

III Real elimination of relational database

- 1. multivalued
- 2. composite attributes
- 3. Specialization / generalization

egm 22 w

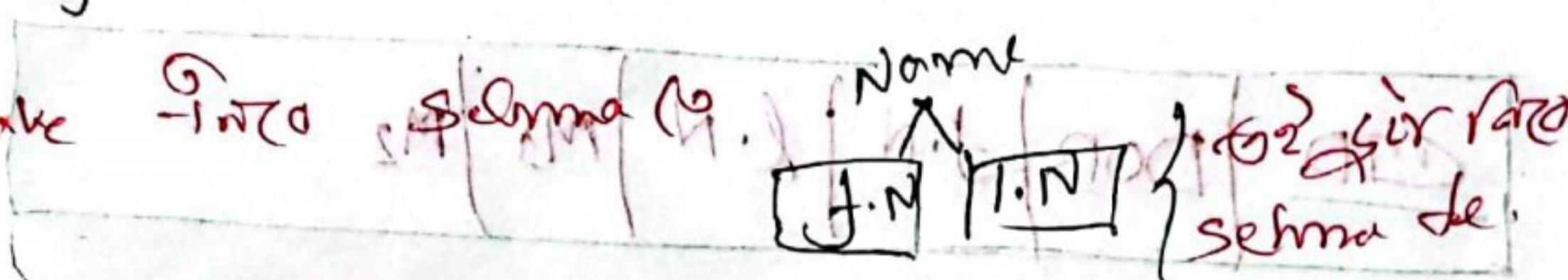
egm Implement
egm object oriented

database large.

Schema to multivalued, composite, derived attributes
—specification, generalization

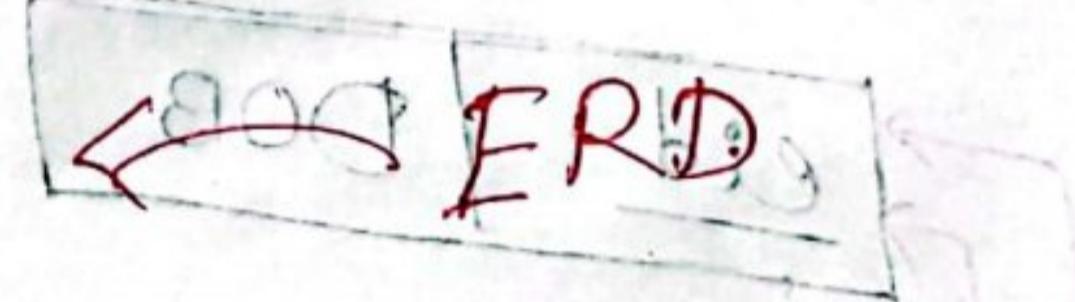
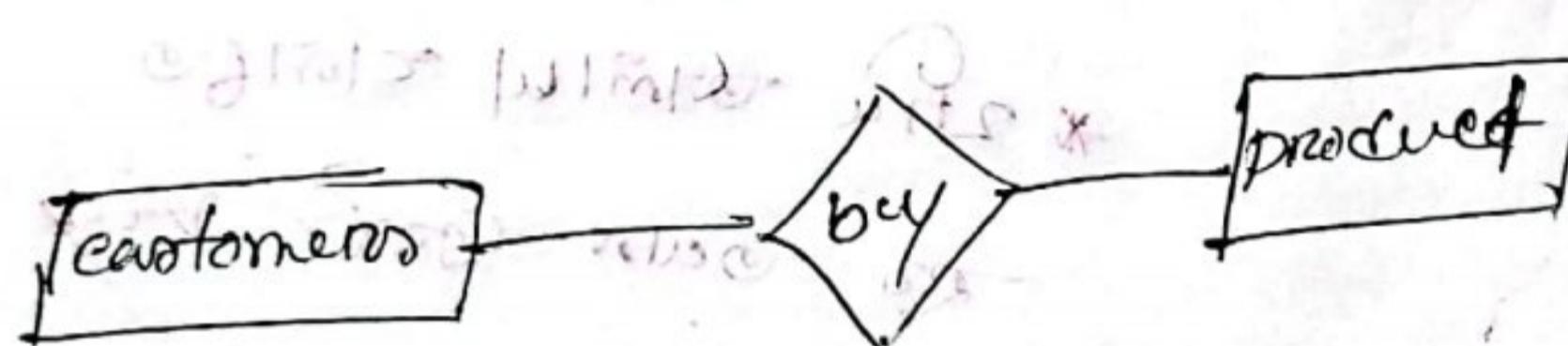
Overall design of database called Schema

composite rule into schema



ERD to Schema

Previous example to Schema



Schema

Customer (cid, DOB, first name, last name, mobile 1, mobile 2)

product (pid, Pname, unit price)

Buy (cid, pid, quantity, buy date)

Primary key relation to Schema

Key rule primary word

underline foreign key

Now ERD to Data Table

* Important

table : customers

CID	DOB	J.N	L.N	M ₁	M ₂



table : customers

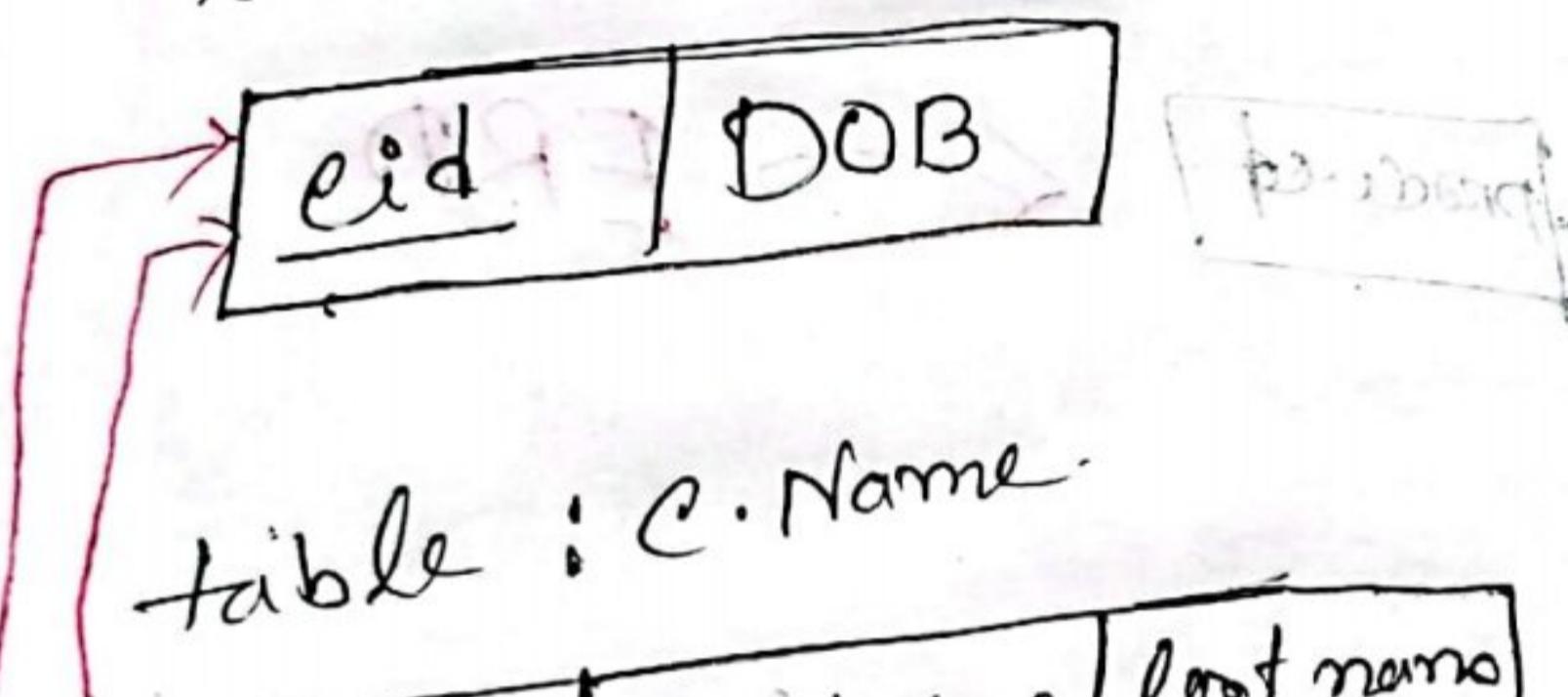


table : c_name

cid	first name	last name

table : mobile

eid	mobiles	mobile

* 2nd primary key
- Q1 own entry key

common track history

Like CID all 3rd

Q1(Q2 Q3) -
primary key
foreign key

~~fair use~~ (not
common, big) having

3rd

common primary

3rd

3rd

3rd

3rd

3rd

3rd

table : Product

Pid	P.Name	unit price
-----	--------	------------

Buy table :

Cid	Pid	Quantity	buy date
-----	-----	----------	----------

✓

~~Relational Algebra (RA)~~

Schema

product (Pid, p name, unit-price)

Rel → buy (Pid, cid, quantity, buy date)

customers (cid, c name, address)

Q1: Find the product details. (RA figure 20)

SQL
Select *
from products
where shop name = "Shob Gaurav".

RA

Select *
from products
where shop name = "Shob Gaurav".

RA (σ shop
name = "Shob Gaurav")

[*] where shop name = "Shob Gaurav" [σ shop name = "Shob Gaurav"]

Q2 find the products details having unit price greater than 10.

$\Rightarrow \underline{\text{SQL}}$ select * from products where unit_price > 10.

[6^{25} or 30 or apply
generalized projection rule]

RA $\{ \text{unit_price} > 10 \text{ (product)} \}$ / $\{ \text{no condition} \}$
 $\{ \text{fact_format } 2(0) \}$
where
6 first

Q3: find the product details having unit price.

\rightarrow improve fact

SQL Select unit_price +2 / RA $\{ \text{unit_price} +2 \text{ (products)} \}$
from products

Q8: find the produced details, that have bought
on '17-01-2023' (Ex: table add (8))

sol

Select pid, Pname, unit-price
from product P, buy b
where p.pid = b.pid and buy date = '17-1-2023'

(RA)

$\pi_{pid, Pname, unit_price}$ and (8)

$\wedge_{buy date = '17-1-23'}$

$P.pid = b.pid$

where (8)

$(P(produced) \times P(buy))$

rename (8)

Schema to RA

EMPLOYER (person-name, street, city)

WORKER (person-name, company-name, salary)

COMPANY (company-name, city)

MANAGER (person-name, manager-name)

① find the names of all employees who work for
first bank corporation.

⇒ SQL

⇒ select person-name

from WORKER

where company-name = "first bank corporation"

RA

Π person-name

(company-name = "first bank corporation") / WORKER

② Find the names and cities of residence of all employees who work for first bank corporation.

Our SQL query for common table (first bank)

Ans
SQl Select works, person-name, city.

From works natural join employees
where company-name = "first bank corporation"
→ [our table add ex. 2012]

RA
 Π_{works} person-name, city
(Company-name = "first bank Corporation")
(works \bowtie employees)
natural join \Rightarrow

* এখন অধিক কমন হ্যান্ড

এখন natural join ব্যবহার করো।

* Natural join 1 - by কমন হ্যান্ড

এই পদ্ধতি অসুবিধা, কৃত হওয়ার জন্য সুবিধা।

natural join কী?

Select first-name, e.department-id, department-name
from employees \leftarrow left outer join departments ^{alias}
on (e.department-id = d.department-id)

d on (e.department-id = d.department-id)

[left outer join Left side go value left \rightarrow is
group Right go common value.]

#A π e.person-name (e.department-id = d.department-id) department
(ρ employee ID)

Select employee-id, first-name, department-id.

from employees

where department-id = (select department-id

from employees

where first-name = 'Lex')

② 21(04) department id Lex go department id Soman

group by department

where first-name = Lex thru Lex - user

group by department

• Find the names of all employees in this database
who live in the same city as the company

for which they work.

Allis

natural Allis foreign name
equijoin \Leftrightarrow allis must be large

>Select E.person_name

from employee as E, work as W, company as C

where E.person_name = W.person_name and

W.company_name = C.company_name

\rightarrow select where for table add $\exists \forall$

and $\exists \forall$ mistake in

Select person_name

natural join works natural join
from employee natural join work natural join
company

$\exists \forall$ $\exists \forall$ $\exists \forall$

BA
 $\exists \forall$ person_name (employee \bowtie work \bowtie company)

* $\Pi_S (G_P(E)) \rightarrow$ projected Selection.

- Outer join use \exists missing expression (good ⚡)
- Outer join use \exists left or right (error), \forall or matching (more
[left] left \exists error, \forall matching (more
[right] \forall right \exists error, \forall matching (more
[matching])
- $G_P(E_1 \times E_2)$

Equijoin

are employee on salary (needs)

Select sum(salary)

from works.

RA g sum(salary) (works)

(
Group function)

~~select city from employee where name = 'X'~~

कैसे हो?

SQL

• select person-name, city

from employee

where city = (select city

from employee

 $\begin{array}{|c|} \hline \text{IN} \\ \hline \end{array}$
(use value match compare)
value match compare)

where person-name = 'X')

RA

~~Π city (personname = 'X' (employee))~~

Π person-name, city ($\sigma_{city = \Pi city (person_name = 'X' (employee))}$)

(condition) (min 2)

प्रदर्शन