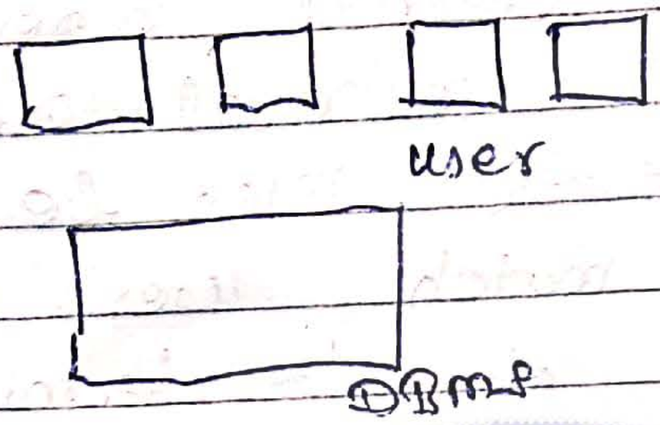
WWW.GRADESETTER.COM

Date : / /
Page No :

RDBMS :-



Users can't interact with database directly they interact through DBMS.

Schema - design of database

Instance - Data in database at particular moment of time.

3 - Schema architecture of DBMS -

external level 1 | external level 2 ---- > differ on based on user.

↓ external | conceptual mapping

conceptual level - view seen by database administrator.

↓ conceptual | Internal mapping.

Internal level

→ which data structures are used to store data.

→ How data resides physically on memory.

A DBMS must transform a request specified on an external schema into a request against conceptual schema and then into

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

entity type, the Primary key of this relation will be the combination of Primary key of Participating entity types.

* Mapping of multivalued attribute:-
For each multivalued attribute a, create a new relation R, this relation will include an attribute corresponding to a plus the Primary key attribute of entity E, to which A belongs.

A → R
A U PU(E)

The Primary key of E, will act as foreign key in relation R and the Primary key of relation R, will be Combination of A and Primary key of E.

• Location

D.NO	location
D1	Patiala
D1	Chandi
D2	Amritsar

* Employee:-

SupVid	Vid	Fname	mname	Lname	salary	Address	DOB	Dept NO

Department:-

Depid	Depname	Dno	Dname	mgr-Vid	start-date

Project:-

PNo	name	location	Dept no

Dependents:-

Vid	Dname	mgrid	

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Dependents

EVid	Deptname	Dep-DOB	Relationship

Department

Dno	Dname

Works_on

emp_Vid	Pno	Hours

Dep_location -

Dept-no	Location

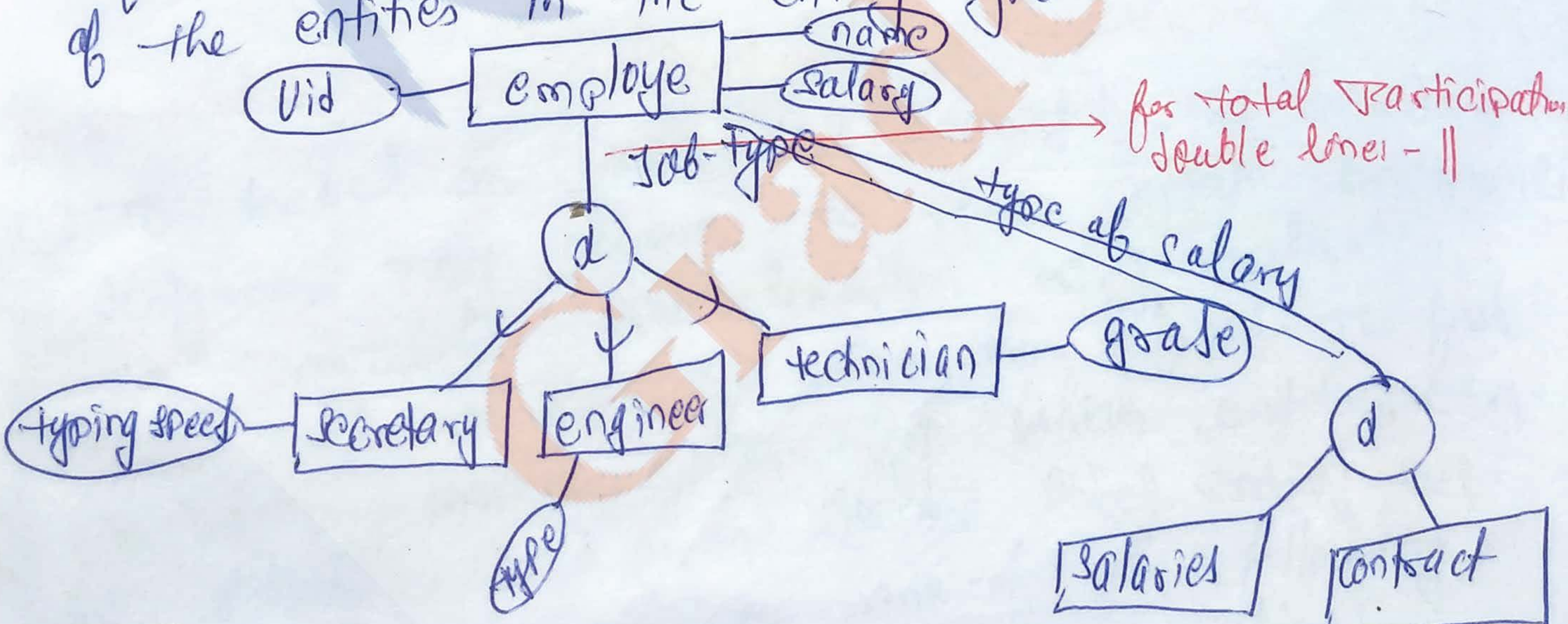
* EER model - (enhanced ER model)

The EER model consists of all the ~~consists~~ constraints of ER model and in addition of it consist of
- specialization
- generalization
- category or Union

— includes class, subclass and relation.

- Specialization :-

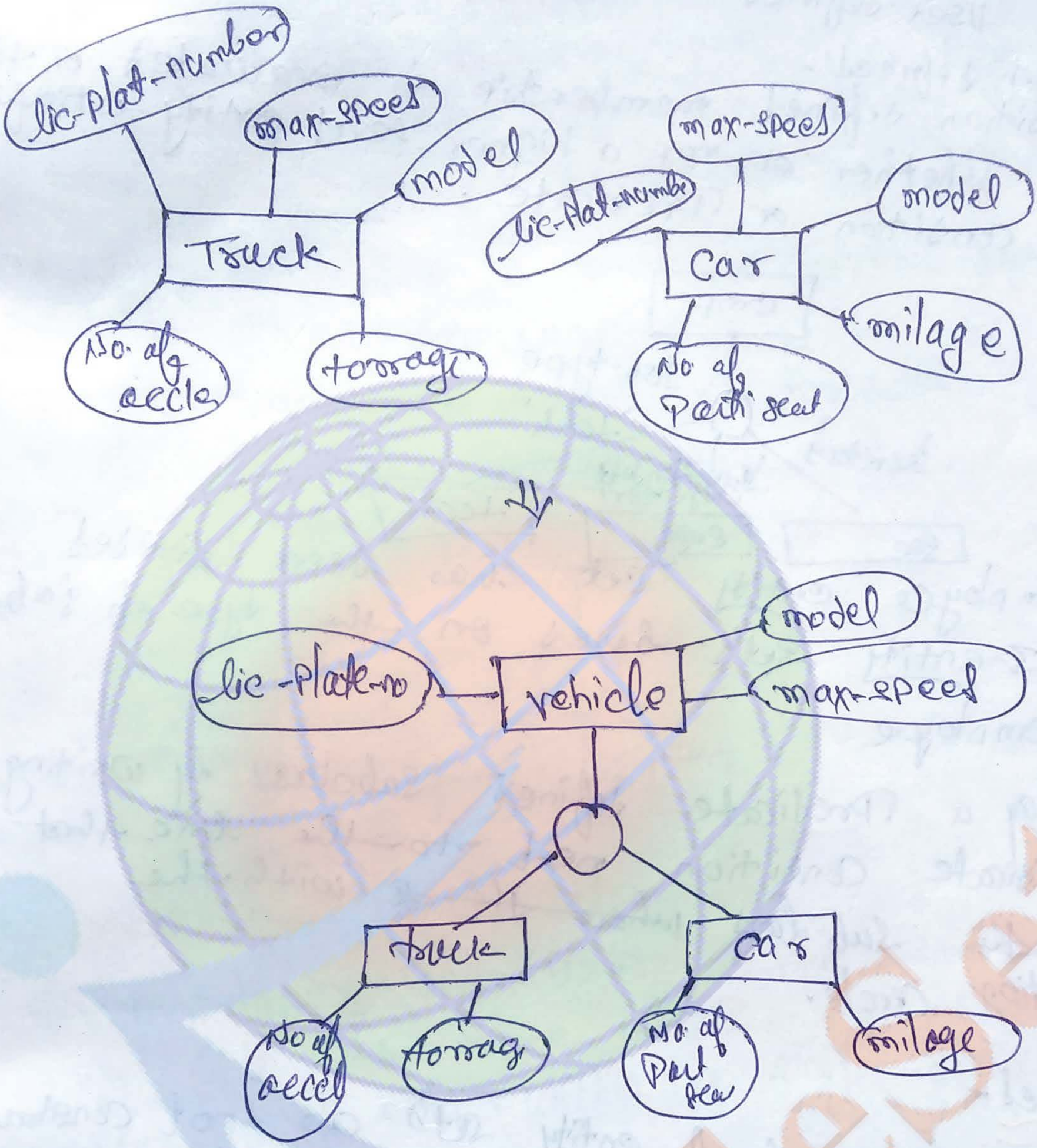
It is the process of defining a set of subclasses of an entity based on some distinguishing characteristics of the entities in the entity type.



WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

* Generalisation:-



Constraints on specialization & generalisation

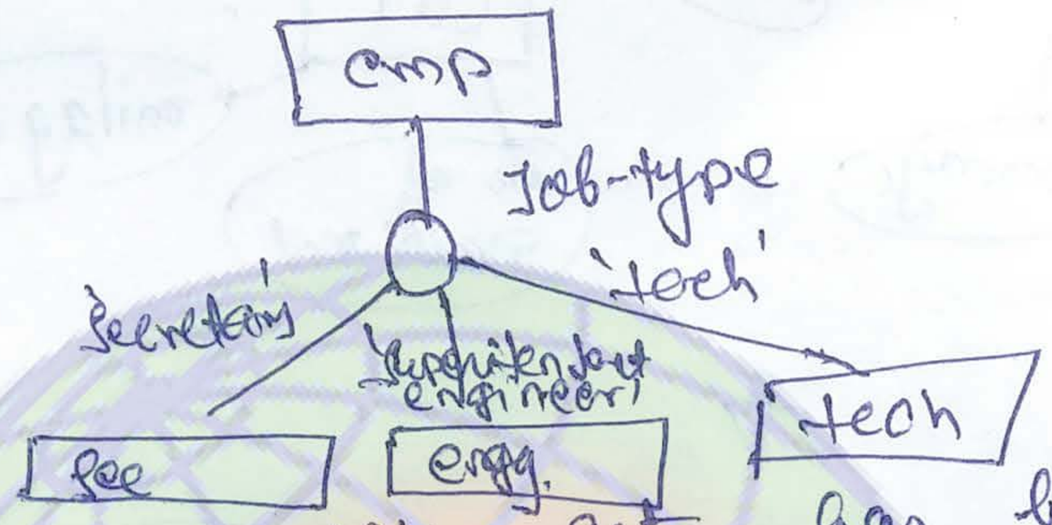
(1) To model an enterprise accurately, the database designer may choose to place certain constraints on particular specialization or generalization. Three such constraints can be defined. The first constraint involves determining which entities, can be members of a given lower level entity set. Such membership can be one of the following condition or predicate defined or user defined.

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

* a) Condition / Predicate
  b) User defined

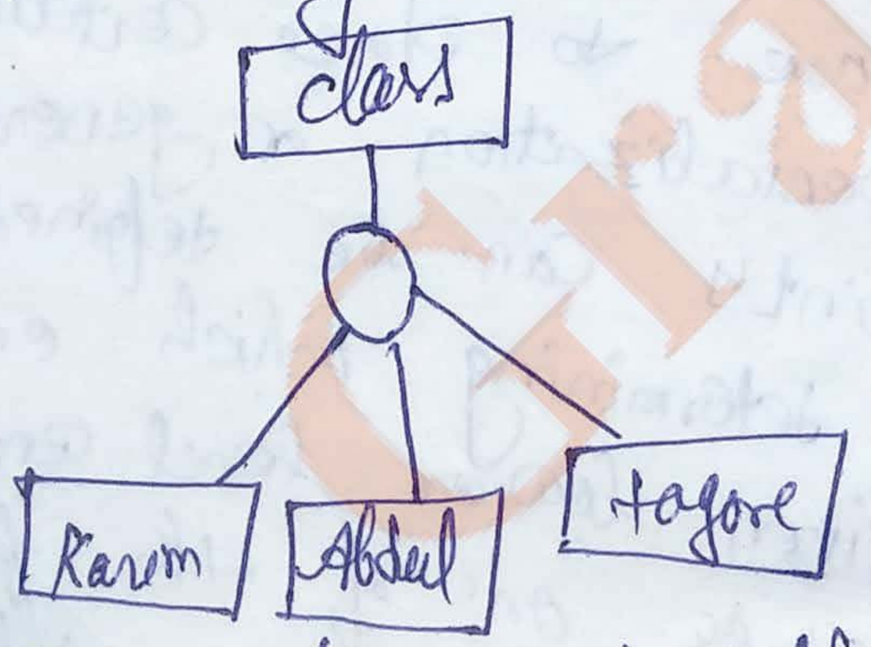
(a) Condition defined:-
In Condition defined membership is evaluated on the basis of whether or nor a higher level entity satisfies an explicit condition or Predicate.
for eg.



the employee entity set has been divided into 3-entity set's based on the type of job of an employe

we display a Predicate defined subclass by writing the Predicate Condition next to the line that Connects the sub-class where the ~~s~~ with the specialization circle.

(b) User defined:-
User defined lower level entity sets are not constraint by a membership Condition rather the database designer assigns the entity to a given entity set.



The 2nd type of constraints, whether or not a higher level entity may belong to more than one lower level entity sets

WWW.GRADESETTER.COM

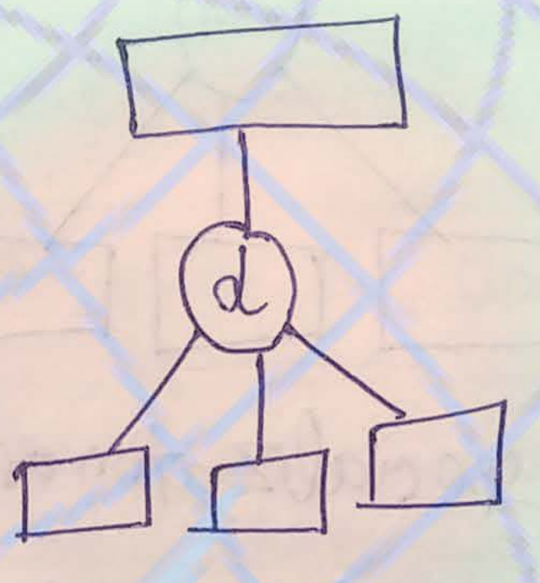
WWW.GRADESETTER.COM

*(i) Disjoint -

The disjointness constraints requires, that an entity belongs to no more than one lower level entity set's. that is an entity can be a member of atmost one subclass of specialization.

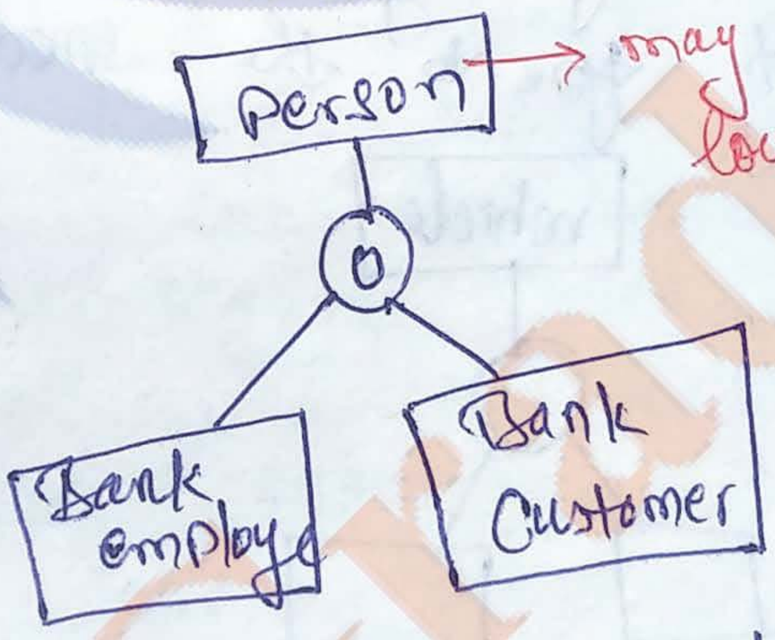
specialization that is predicate defined automatically employes disjointness constraint.

The disjointness constraint is indicated by placing a 'd' in the specialization circle.



(ii) Overlaping specialization:

In overlaping specialization the same entity may belong to more than one lower level entity sets. Within a single specialization



→ may belong to both the lower level entity sets

An overlaping is indicates by placing an 'o' in the specialization circle.

Completness Constraint:-

The completness constraint specifies, whether all the entities in the higher level entity

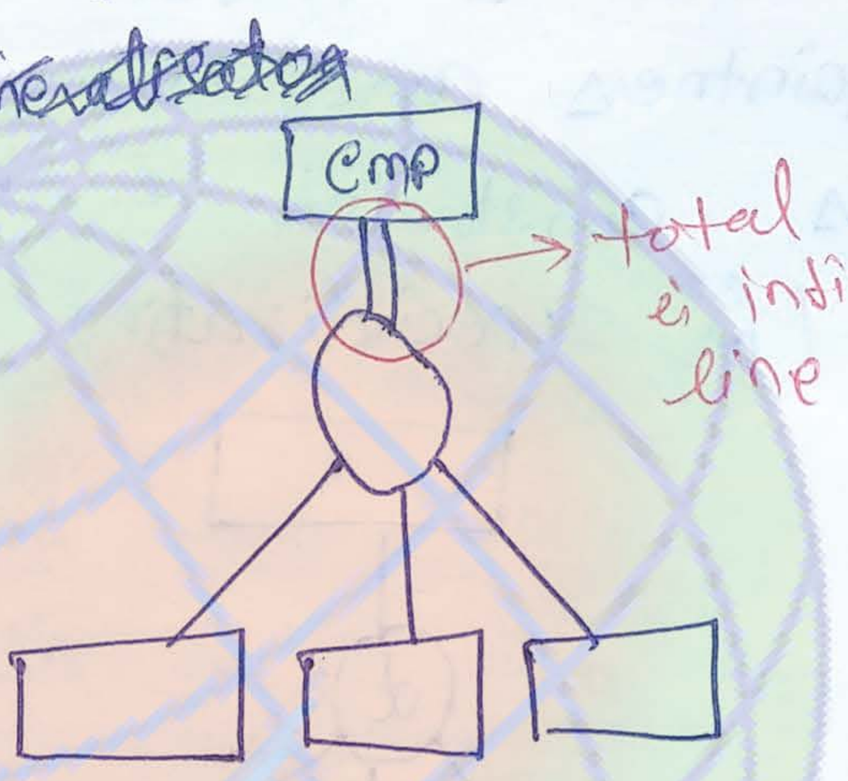
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

set belong to atleast one lower level entity set or not.
This constraint can be one of the following.

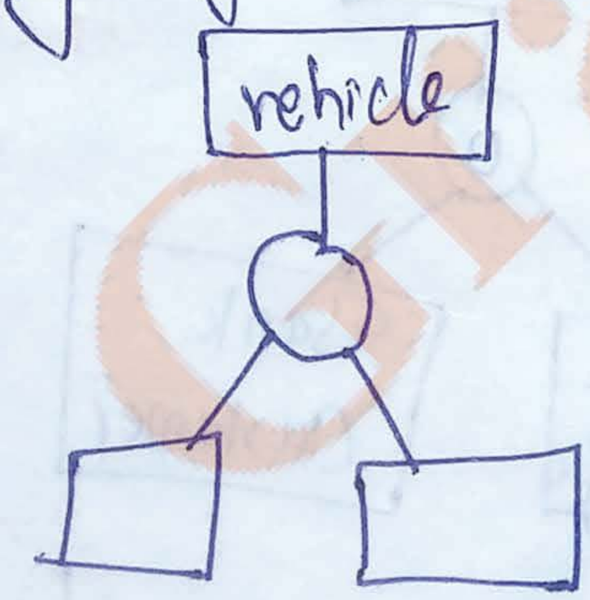
(a) total specialization or generalization
↳ here each higher level entity must belong to a lower level entity set.

eg: Generalization



(b) Partial specialization or generalization
Some higher level entities may not belong to any of the lower level entity types. defin in the specialization.

The ei Indicates by single line '|' from the super class entity type to the specialization circle



WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date: / /
Page No:

Differes b/w a latice and shares subclas.

• Diffb b/w shares subclass and union type

A shares subclass inherits the properties of all it's superclasses.
the entities of a shares subclass, are the entities that we get on intersection of all the super classes.

whereas the category or Union type is the subset of Union of it's superclasses.

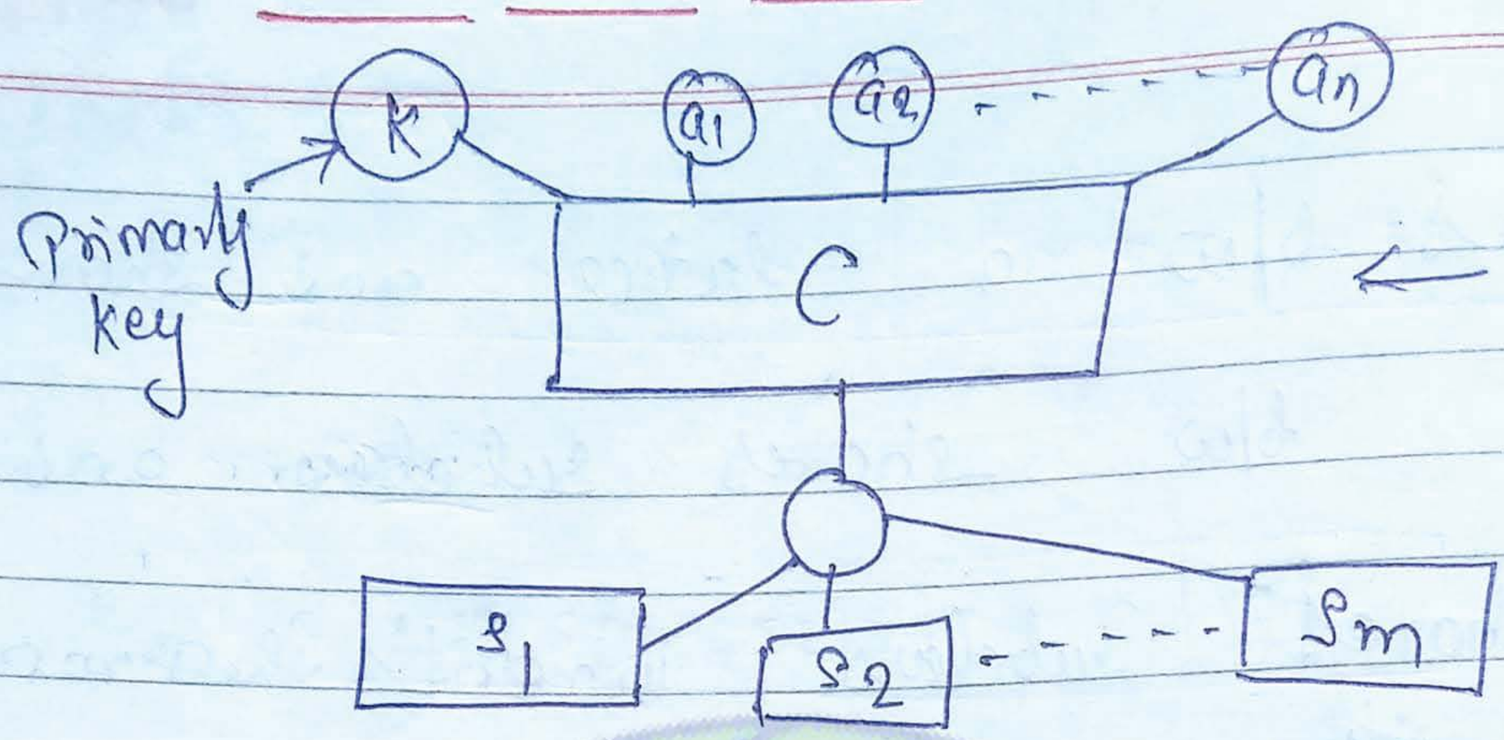
Attribute inheritance works more selectively in case of categories
for eg.
each owner entity inherits the attributes of a person or a bank or a company depending on the superclass to which it belongs.

A category can be total or partial. A total category holds the Union of all the entities in it's superclasses, whereas a partial category holds a subset of the Union.

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

EER to Relational:



← exam मे बताएँ कि क्या है, K, C, S1, ~ कि क्या है

We can convert the specialization into corres. relation using one of the four methods:-

(1) multiple relations - (superclass and subclass)

create a relation 'i' corresponding to super class C, the attributes of 'L', will be $K \cup \{a_1, a_2, \ldots a_n\}$

$L \rightarrow C$

attrs $(L) \rightarrow K \cup (a_1, a_2, \ldots a_n)$

And the primary key of relation 'i' will be 'k'.

P.K $(L) \rightarrow K$

- create a relation L_i corresponding to each subclass s_i

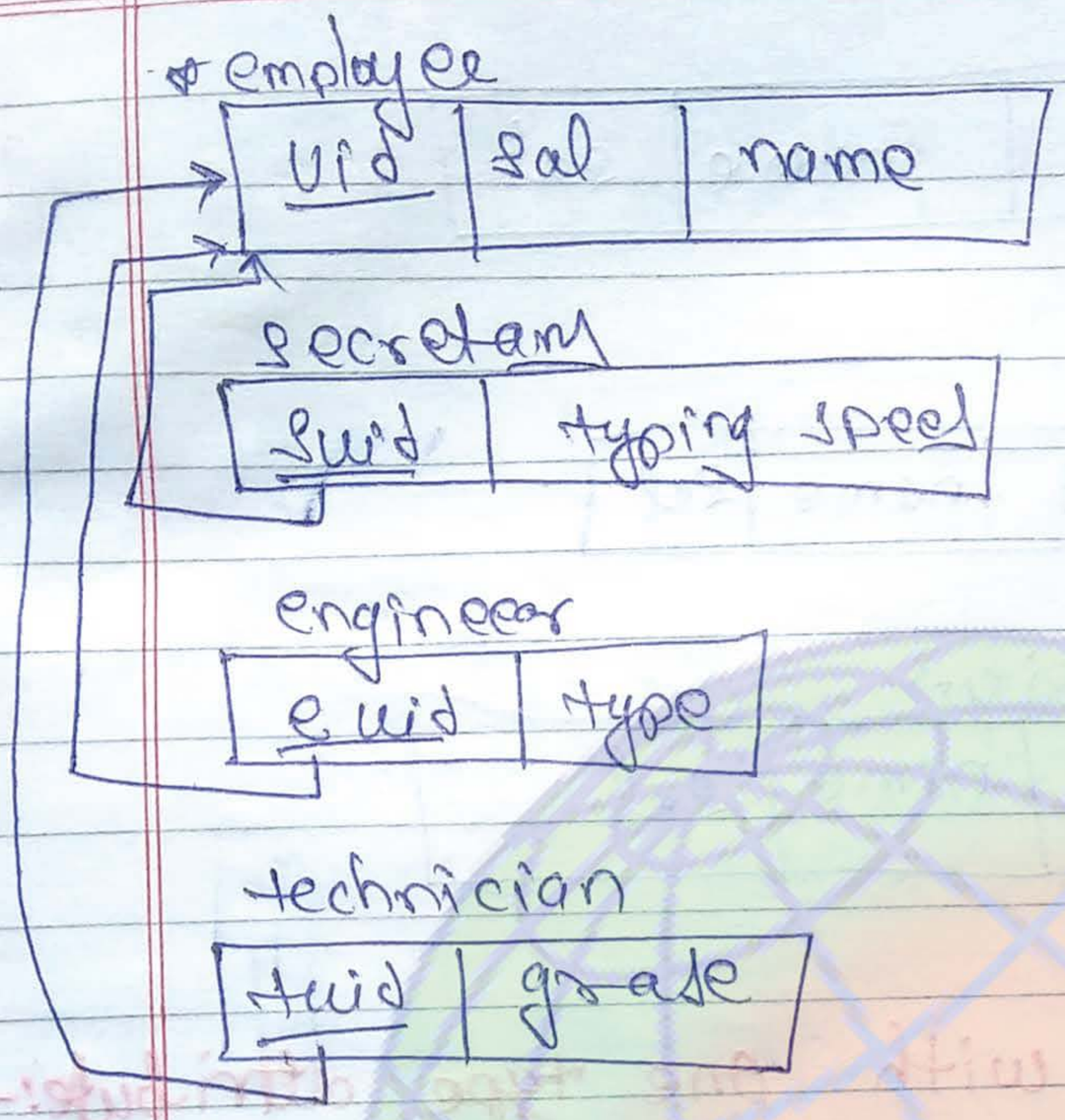
$L_i \rightarrow s_i$ where $1 \leq i \leq m$

where i varies from 1 to m and the attribute of L_i will be K U attrs (s_i)

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date: / /
Page No:



This method can be used for any type of relation whether it is total, whether it is partial

(2) multiple relations - Subclass relations only

~~For this~~

create a relation 'Li' corresponding to each of the subclasses Si

$$Li \longrightarrow Si$$

The attributes of 'Li' will be ~~KU{a1, a2 ... an}~~

$$KU \{a_1, a_2 \dots a_n\} \cup Attr(Si)$$

where

$$1 \leq i \leq m$$

Note:

This method is useful when the specialization is total and disjoint.

If the employee fall only in these three category,

then the specialization can be represented using this method.

SARAA

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

engineer

Vid	eng-type	name	sal

secretary

Vid	typing_speed	name	sal

~~technical~~ technician

vid	grade	name	sal

(iii) Single relation with one type attribute:-

Create a single relation 'L', to represent the specialization the attributes of 'L' will be K U attrs of super class C union the ~~attributes~~ the type of attributes union attributes of all the Subclass.

Attrs(L) →

K U Attrs(C) U type attribute V attrs(S_1) V ---- V Attrs(S_m)

employee

Vid	name	sal	Job-type	eng-type	T-grade	typing speed

↓ Tech

engg

↕ comp

Note: This method is useful when the specialization is partial.

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date :
Page No :

a request on to a Internal schema. Ifo request is for data retrival. Data extracted from database must be reformatted to match user's external view. The procey of transforming results b/w levels is called mapping.

* Data Independence :-

Logical data Independence
Physical data Independence

* Data model -

ER model - is used for purpose of designing database.
ER model is based on presiption of real worlds that consists of entities and it also defines relationship among them.

entity

attribute

Relation

Total participation

A Company is organised into departments each department has a unique name, a unique no, & particular employe who manages department. we also keep track of staff data when that department. A department may

SARAA

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

(IV) Single Relation with multiple type attributes:

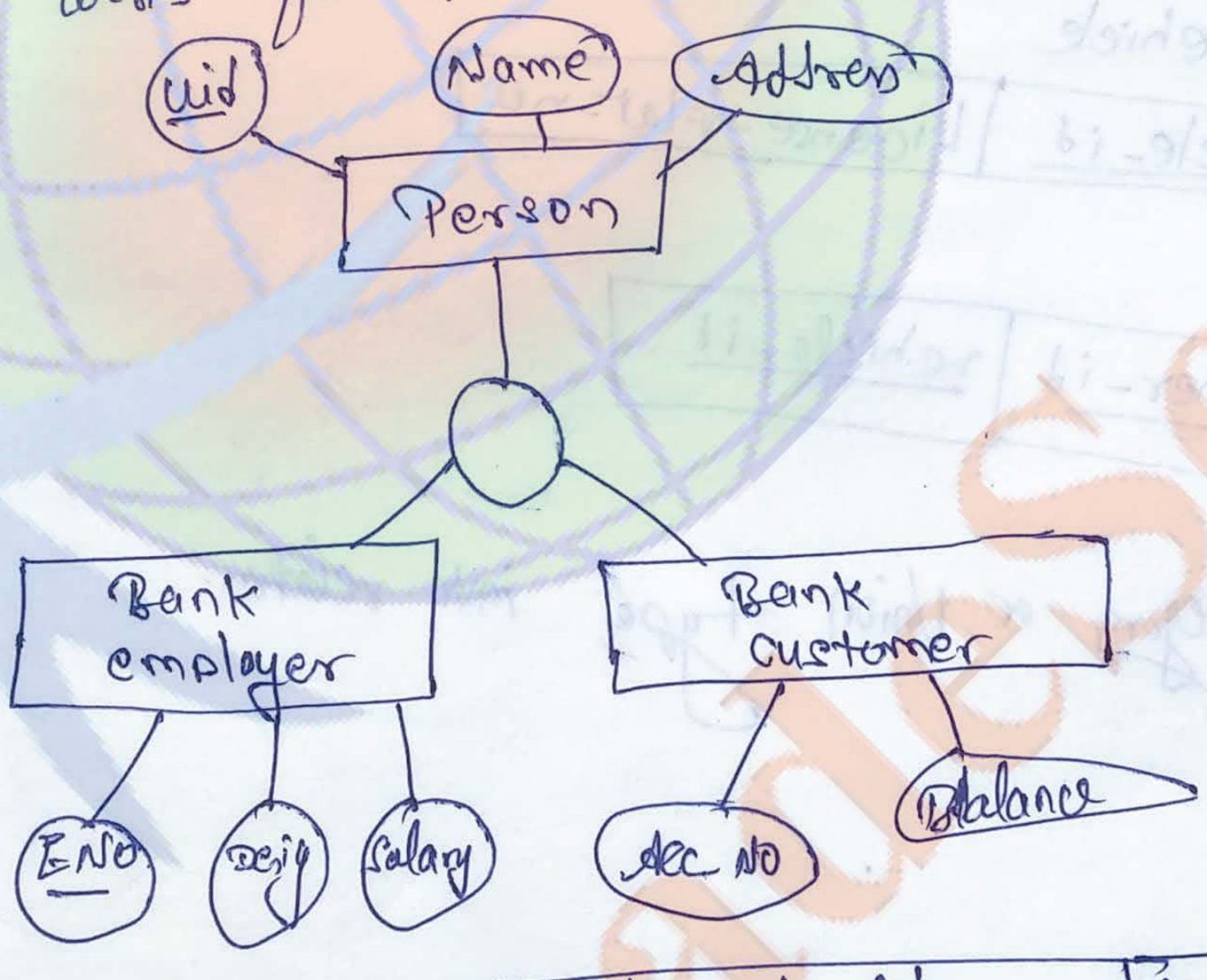
Create a relation 'L' where attributes of 'L', will be attribute of the super class C, Union $\{t_1, t_2 \ldots t_m\}$ type attribute, Union attribute of all the subclasses.

$$Attr(L) = Attr(C) \cup \{t_1, t_2 \ldots t_m\} \cup Attr(S_1) \cup Attr(S_2) \cup \ldots \ldots \cup Attr(S_m)$$

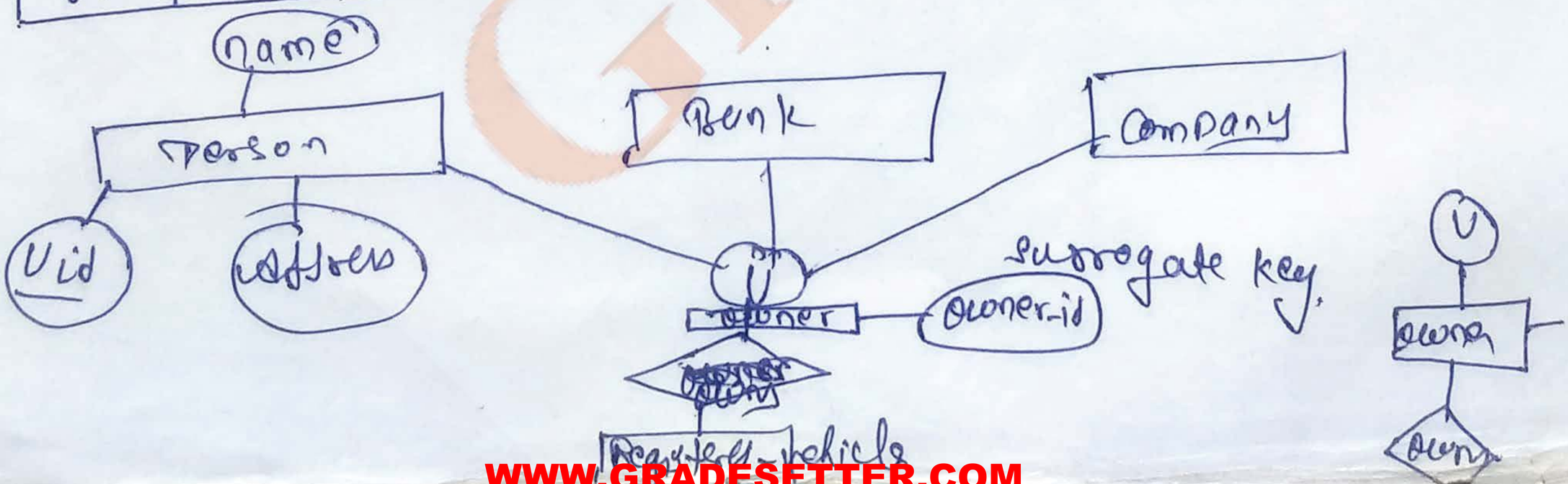
Each type attribute t_i is a boolean indicating, weather a touple belongs to subclass S_i or not

$$t_i \rightarrow S_i$$

only if a touple belongs to S_i, then the corresponding attributes will be filled otherwise, they will have null values. this option works for specialization, whose subclasses are overlaping.



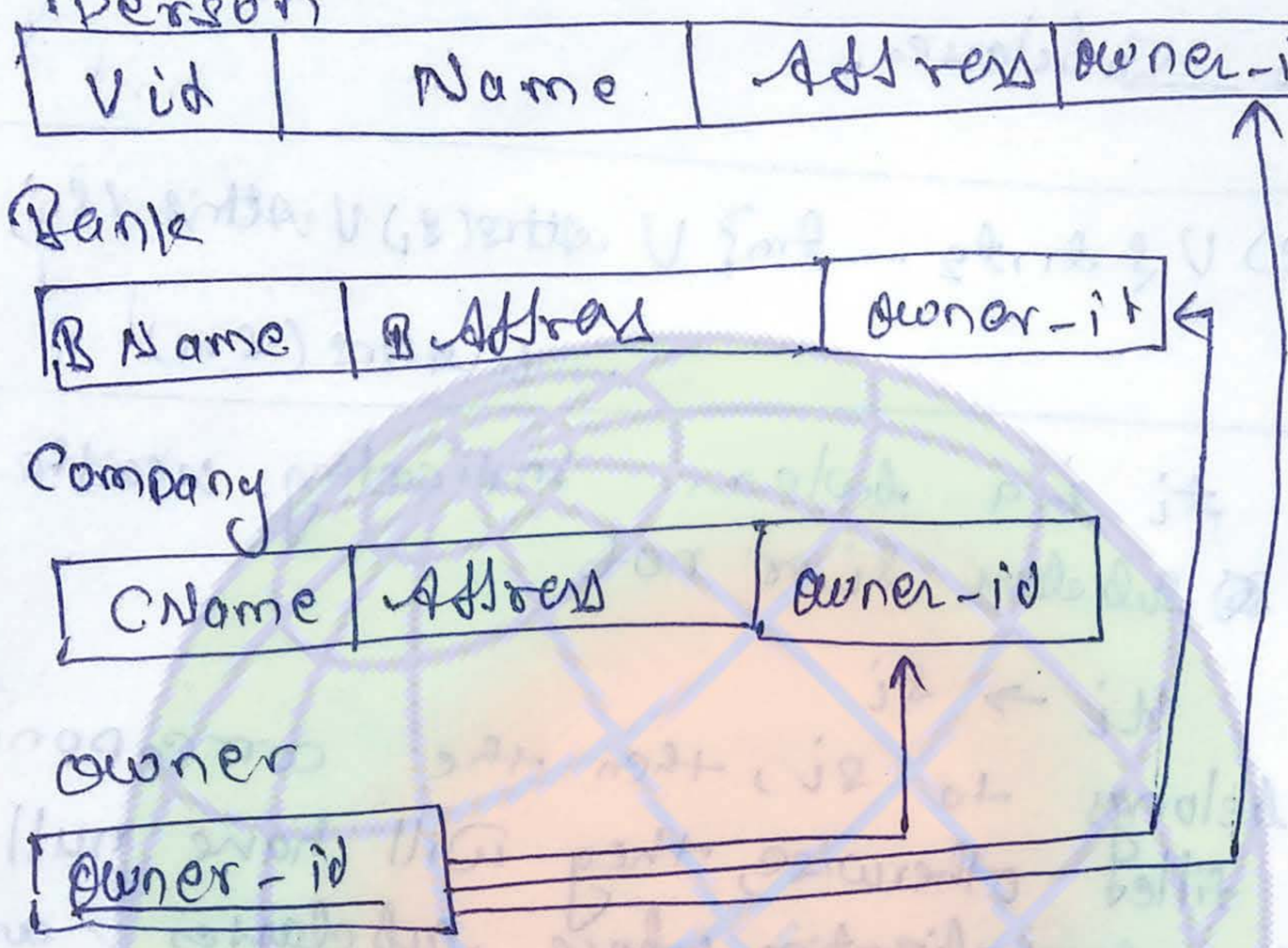
Vid	Name	Addres	D_Emp	ENO	Desig	sal	D_Cus	ANO	Bal



WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

For mapping a category, whose defining superclass have different keys, we create a new key attribute called surrogate key.



Note: Category or Union type into relation.

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

* Advantages of Concurrency:

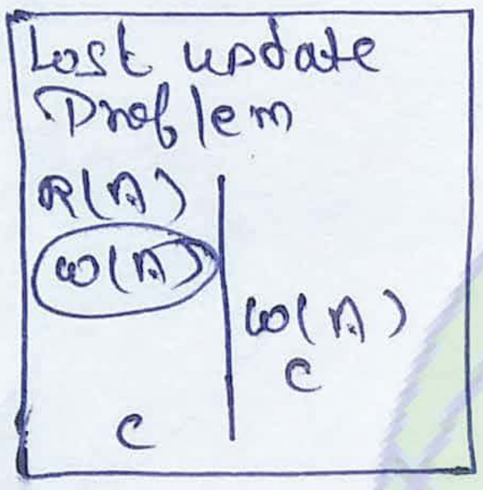
(a) Waiting time (↓)

(b) Response time (↓), Response time will also decrease.

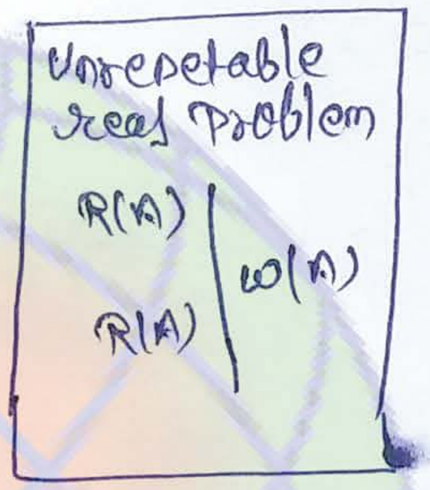
(c) ~~Utilisation~~ Resource utilization (↑)

(d) efficiency (↑)

* Lost update Problem







WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

SQL - Declarative language

PL/SQL → Procedural language Extension of SQL.

Date:
Page No:

- It is an extension of SQL.
- Not case sensitive
- Literals
  - character literals 'A'
  - Numeric literals 29
    36.72
    -82.7
  - String 'Hello'
  - True, False
- Comments
  - single line comments
    - —
  - multi line comments
    /*
    */

A-Z
0-9
Special Symbols

var_name data-type;
a number(3,2)

e.g.: constant number(3,2) := 3.14;

*

emp

Vid	Name	Addres

% type attribute -

% type
emp.Name % type;
aff emp.addrey % type;

*

declare
    Declaration section.
begin
    executable statements.
    SQL statements.

SARAA

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

exception :

end ;

All the above sections are optional.

declare, begin, exceptions, end

~~declare~~

declare

a number; | a a number := 20;
b number; | b number := 35;
c number; | c number := a+b; a = 20
↳ comparision operater

a := 20
↳ assignment operater

begin

a := 20;
b := 35;
c := a+b;

dbms_output.put_line ('Sum of ' || a || ' and ' || b || ' is ' || c);

end ;

If (a > b)
then
d.o.p.l ('a is greater');
end if;

<< label >>

goto label;

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

* If condition1
then
action1
elsif condition2
then
action2
elsif
:
:
else
default action;
end if;

* loops:-

loop — — — — — — end loop;	loop a:= — exit when condition; end loop;

* while condition
loop
a := a+2;
— —
— —
end loop;

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

(*)

(1) Write a PL/SQL block to find greater of three numbers.

2. Write a PL/SQL block to calculate the Price of Petrol in a given State.

Price of Petrol = Price of refined oil + excise duty + Commission of seller + VAT

↓ ↓ ↓ ↓

₹32/l ₹21.54/L 2.63/l Pun 20% & 16% del 2%

Soln

```
declare
  a number;
  b number;
  c number;
begin
  a := 10;
  b := 20;
  c := 30;
if ( a > b and a > c )
then
dbms_output.put_line ('A is greater');
elsif ( b > a and b > c )
then
dbms_output.put_line ('B is greater');
else
dbms_output.put_line ('c is greater');
end if;
end;
/
```

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

* → Declare

⇒ Price number(4,2),
initial number(4,2),
refined number(2) := 32,
exide number(4,2) := 21.54;
Comin number(3,2) default 2.67;
~~rate~~ rate varchar2(8);
rat number(2);

→ Begin
rate := 'star'
if rate = 'star'
then
rat := 30;
elsif rate = 'star'
then
rat := 16;
elsif rate = 'Del'
then
rat := 25;
end if
initid := refined + exceed + Comm;
Price := initial + initid;
dbms_output.put_line('Price is Petrol');
in' || state || 'is' || Price;

→ end

// Case variable
when con1 then statement 1;
when con2 then statement 2;

space वाली आएगा

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

else
default statement
end case;

Case state
when 'Pun'
then
rat := 30;

rat := Case State
when 'Pun' then 30
when 'Har' then 16
~~Ha~~ ~~Har~~ 16
when 'Del' then 25
End;

Q) Write a PL/SQL block to calculate grade of student given marks in different subjects.

Sol)

~~Dec~~ ~~declare~~
marks1 Number(2) := 54;
marks2 Number(2) := 65;
marks3 Number(2) := 81;
marks4 Number(2) := 84;
marks5 Number(2) := 82;
Total number(3);
Per Number(2);
Grade varchar2(3);

Begin
Total := marks1 + marks2 + marks3 + marks4 + marks5;

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

Per: (Total/500)*100

If per >80 AND per <85 then
Grade := 'A'

else if Per >85 AND Per <90 then
Grade := 'A+'

else if per >90 then
Grade := 'O'

else
Grade := 'B+'

endif;
end;

loop

end loop;

Declare
a number(2) := 1;

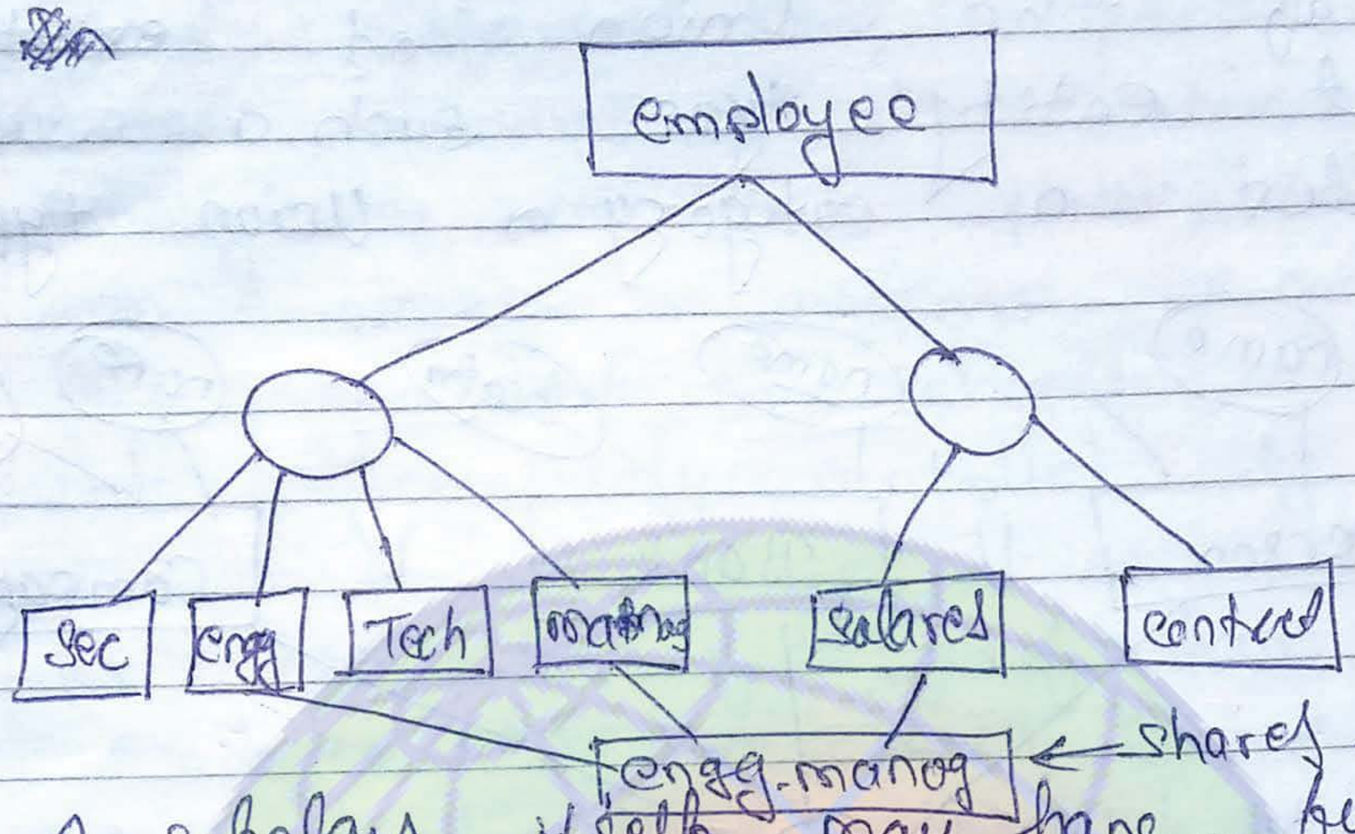
Begin
loop
dbms_output.Put_line('a =' || a)
a := a+1;
exit when a> 10;
end loop;
end;

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date: / /
Page No:

Hierarchy / Lattice:-



← shared subclass.

A subclass itself may have further subclasses specified on it forming a hierarchy or a lattice.

A specialization hierarchy or lattice, that every subclass, participates as a subclass in only one class/subclass relationship i.e. each subclass has only one parent this will result in a tree like structure.

A subclass with more than one superclass is called a shared subclass. and the existance of atleast one shared subclass leads to the concept of lattice.

Category (Union type)

Some times need may arise for modeling a class subclass relationship with more than one superclasses where the superclasses represents different entity types. In such a case, the subclass will represent a collection of objects. that is

SARAA

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

while loop

```
Declare
  a number (2) := 1';

Begin
  while (a <10)
  loop
dbms_output.put_line ('a = '||a)
    a := a+1';
  end loop;
  end;
```

For loop

```
Declare
  a number (2) := 1';
  Begin
    for a in 1..10
  loop
dbms_output.put_line ('a = '||
  end loop;
  end
```

```
Declare
  a number;
Begin
  for a in reverse 1..10
  loop
dbms_output.put_line
  ('a = '||a);
  end loop;
end;
```

SARAA

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

Ques. write a PL/SQL block to find sum of first natural variables.

Ques. write a PL/SQL block to find sum of first 100 odd numbers.

Ques. PL/SQL to find wheather given No. is Prime or not.

```
Declare
    a number (2) := 1;
    sum number (5) := 0;
Begin
    for a in 1..100
        sum = sum + a
    end loop
dbms_output.put_line('sum=' || sum)
    end;
Declare
    a number(2) := 1;
    sum number(2) := 0;
Begin
    for a in 1..100
        if a mod 2 <> 0 then
        sum = sum + a;
    end if;
end loop
dbms_output.put_line ('sum=' || sum);
end;
```

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

A-1 Declare
state varchar2;
a number(2):=7;
b number(2):=1;
Begin
for b in 2..a-1
loop
if a mod b =0 then
state:='T';
end loop;
if state ='T' then
dbms-output.Put_line("No is not Prime");
else
dbms-output.Put_line("No is Prime");
end if;
end;
Delare
R info.RollNO%type;
N Info.Name%type;
C info.class%type;
Begin
R:=101;
N:='Harjeet singh';
C:='B.tech III';
Insert into Info(RollNO, Name, class) value(R,N,C);
end;
Delare
R info.RollNo%type;
N info.Name%type;
C info.class%type;
Begin
Select RollNo, name, class into R,N,C from info where
RollNo=101;
dbms-Output.Put_line('RollNo, name and
class are '||R||N||C);
end;

WWW.GRADESETTER.COM

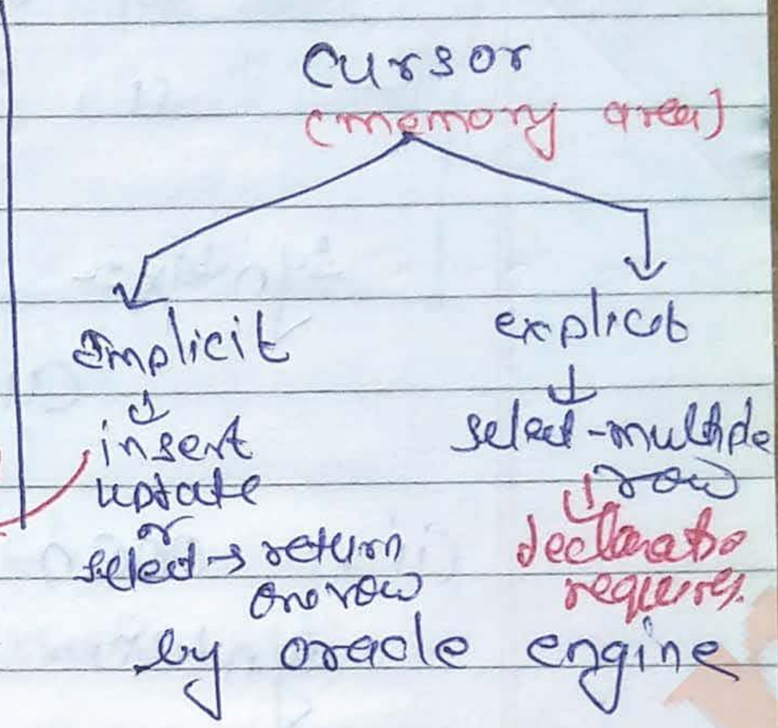
WWW.GRADESETTER.COM

CURSOR

Date : / /
Page No :

A cursor represents a work area or section of memory in which an SQL statement is being processed in an oracle server.

Cursors can be of two types:-
(i) Implicit cursor
(ii) Explicit cursor.



(1) Implicit
Implicit cursors are created by oracle engine itself.
PL/SQL creates an implicit cursor every time a DML statement such as insert, update, or select that returns a single row is executed.

~~(x) Explicit cursor~~ (i) Declare
(ii) Open
(iii) Fetch
(iv) Close

(ii) explicit
explicit cursors are used when you need to retrive more than one row of data at a time through a select statement.

The row's that are returned by the select statement are stored in an explicit cursor. We can then use this cursor to fetch these rows one at a time into the PL/SQL Block.

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

- The steps involved in declearing and manipulating a cursor are

(i) Declaring a cursor

Syntax:-

Cursor Cursor-Name is Select statement

(ii) Open

Syntax:-

Open Cursor-Name;

(iii) Fetch

Syntax:-

Fetch Cur-Name into PL/SQL variables

(iv) Close

close Cur_Name;

(a) % is open - It returns true, if the cursor has been open succesfully false other wise.

(b) % found - It returns true, if the records has been fetch succesfully.

(c) % notfound - It returns true, if the records not fetch succesfully.

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

(d) %. rowcount - It returns the no. of records processed from cursor

c1

Roll	Name	class
101	A	B.Tech III

* SQL %. rowcount

* Begin
Update Info
set class = 'B.Tech III C.E'
where class = 'B.Tech III e.s.E';
dbms_output.Put_line (SQL %. rowcount);
end;

Declare
R info.Rollno %. Type;
N info. Name%. Type
C info.class %. Type;
cursor c1 is select Rollno, Name, class, from Info
Begin
open c1;
for C in 1 .. 5
loop
Fetch c1 into R, N, C;

dbms_output.Put_line (R || ' ' || N || ' ' || C);
end loop;
close c1;
end;

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

→ Begin execute

marks

RNO	m1	m2	m3	total
101	85	75	69	
102	76	81	73	

marks

RNo. m1

① declare
R marks.RNO%type;
m4 marks.m1%type;
m5 marks.m2%type;
m6 marks.total%type;

cursor c1 is select m1, m2, m3 from marks;
Begin
open c1;
Fetch c1 into R, m4, m5, m6;
while (c1%found)
loop
T:= m4+m5+m6;
update marks set total = T
where Fetch c1 into R, m4, m5, m6;
Rno=R;
end loop;
close c1;
end;

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

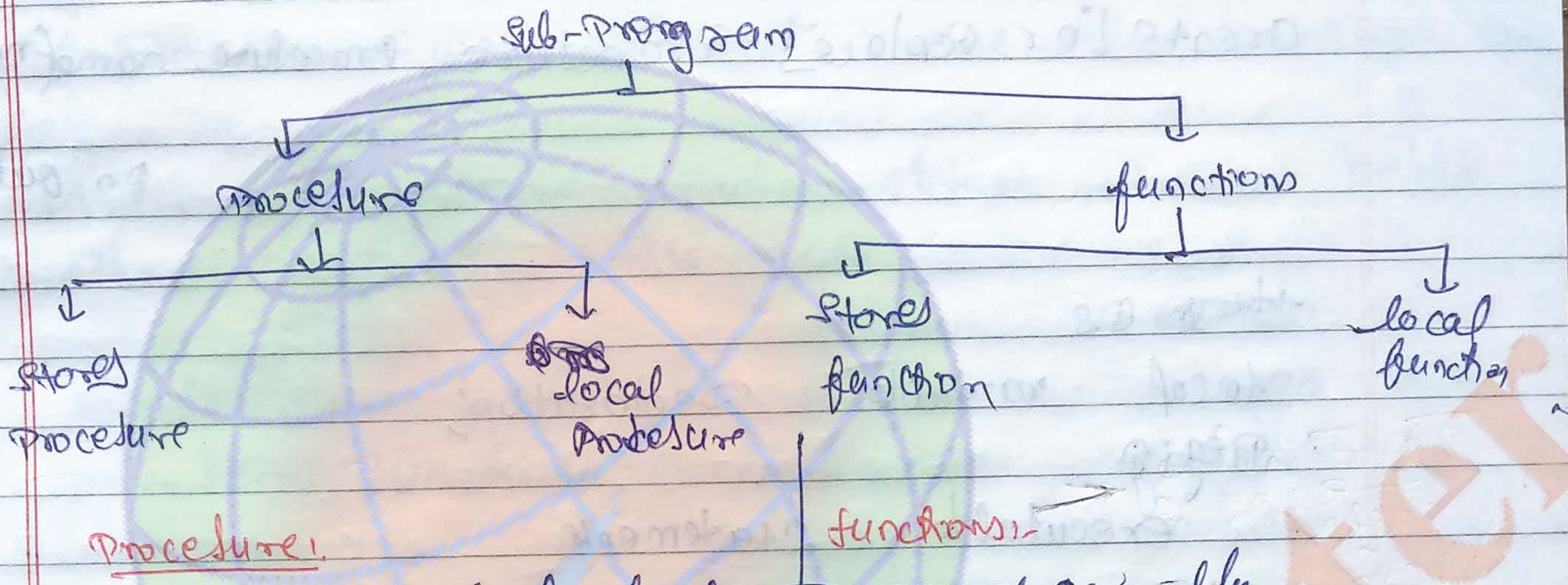
Sub-Programs:

Sub Programs are named ~~PL/SQL~~ PL/SQL blocks

They can be of two type -

(1) Procedure

(11) functions



Procedure:

- ~~local procedure / local function~~
- Used to perform some task.
- do not return a value
- Procedure may take 0 to n input Param
- ~~procedure~~ function can be called from ~~function~~ procedure

functions:-

- is part of generally
- Calculate & return a value
- return only single value
- function take one input Parameter
- ~~function can be~~ procedure can't be called from function
- is a logically group

A procedure or a function is a group of SQL and PL/SQL statements that perform a specific task.

A ~~stored~~ stores procedure or function is a named PL/SQL code that has been compiled and stored in Oracles system tables.

A function must return a value back to the ~~caller~~ caller. and it can return only one valu

SARFAZ

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

By defining multiple out parameters in the procedure, multiple values can be passed to the calling environment.
The syntax for defining the stored procedure is -

Create [or Replace] Procedure procedure_name(name of parameter mode in, out, inout) datatype

DS

local variable Declaration;
Begin
executable statements
end;

create or Replace Procedure A, (a in number,
b in number, c out number)
d out number,

Is
Begin
c := a + b;
d := a - b;
end;

Declare
a number;
y number;

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Packages

Date: 1 1
Page No:

- is similar to directer folder in windows.
- It is an oracle object uses to store Procedure, functions.

Package specification — Package body — —

Parameter mode (in/out) & Type

For access :- we will have to use the Package name

A Package is an oracle object, which holds other object/s within it.

objects commonly held within a Package are Procedures, functions, cursors etc.

A Package has two Parts :-
(i) Package specification and (Public objects)
(ii) Package body or definition. (Private objects)

The syntax for Package specification is

Create or replace Package Package_name
is
function declaration;
~~function~~ ~~Proc~~
~~Procedure~~ Procedure declaration;
end;

So,

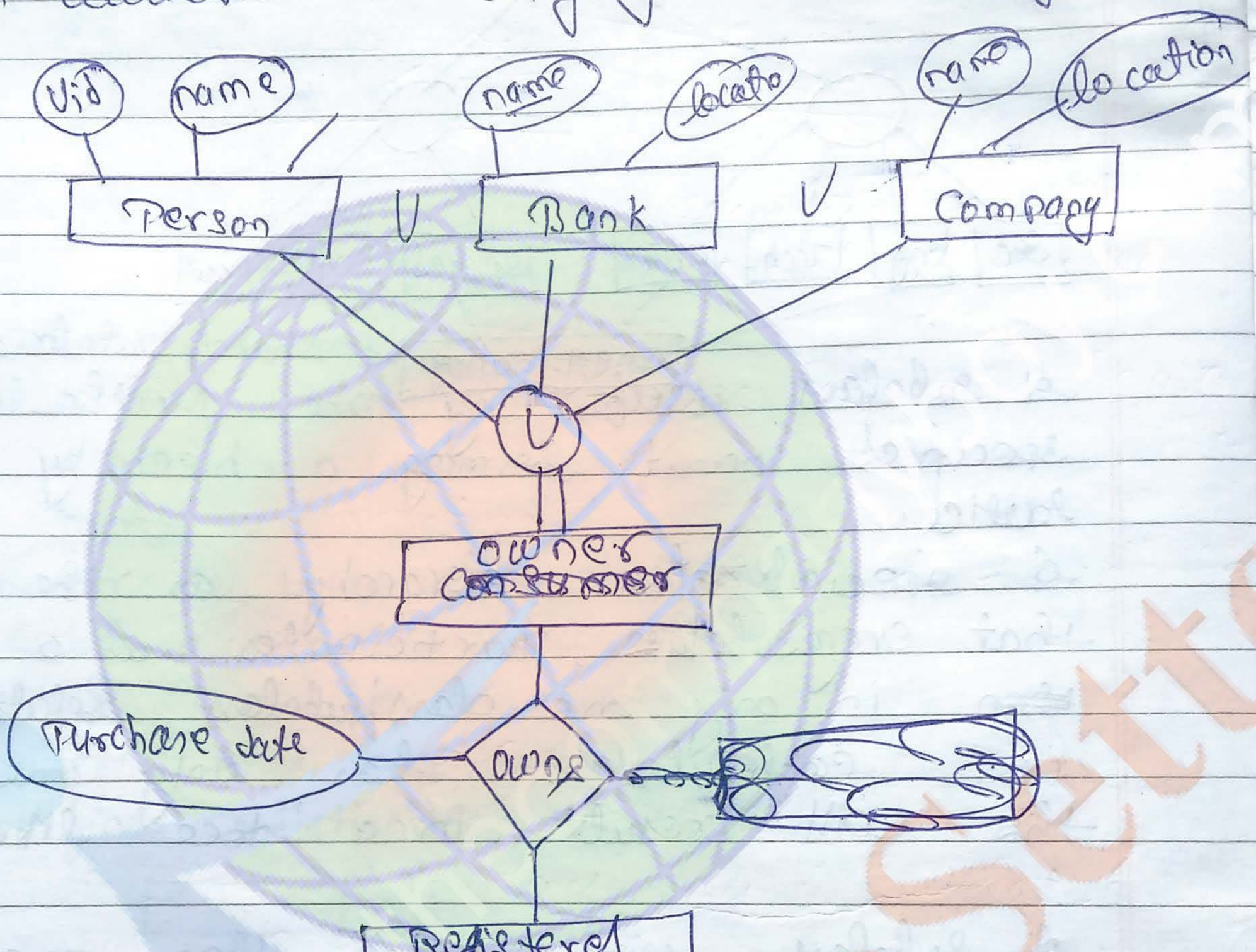
For body of package :-

Create or Replace Package Body Package-name
is

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Subset of the Union of ~~existing~~ the distinct entity type. Such a subclass is called a category or Union type.



A category has two or more superclass that may represent distinct entity types whereas a specialization hierarchy a subclass participates as a subclass in only one class, subclass relationship.

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

function definition;
Procedure definition;
end;

* Employee
Vid Ename Addres salary Age

Q Create a package name 'emp', in that package create a function name 'Tax' which takes employe ~~number~~ vid as argument and returns the tax liability of an employee.

a) Create a procedure 'taxation' that displays vid, name, Addres, salary, age and tax liability of all the employes.

Note:
- 2.5
- 2.5 - 5 — 10%
- 5 - 10 — 20%
- 710 — 30%

Note: function can be excecuted from select statement

Soln

Create or Replace Package Body emp
is
Note:
Select Emp.Tax(101) from dual;
Begin
Emp.Taxation;
End;

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

Create or Replace Package Body is
create or Replace function Tax (id in number)
return number
is
Tax number;
sala number;
Begin
select salary into sal from employee where Oid = id
if (s < 2.5)
then
Tax = 0;
elsif (s > 2.5 and s < 5) then
Tax = (10 / 100) * sal;
elsif (s > 5 and s < 10) then
Tax = (20 / 100) * sal;
elsif (s > 10) then
Tax = (30 / 100) * sal;
return (Tax);

Create or Replace package emp is
create or Replace function Tax

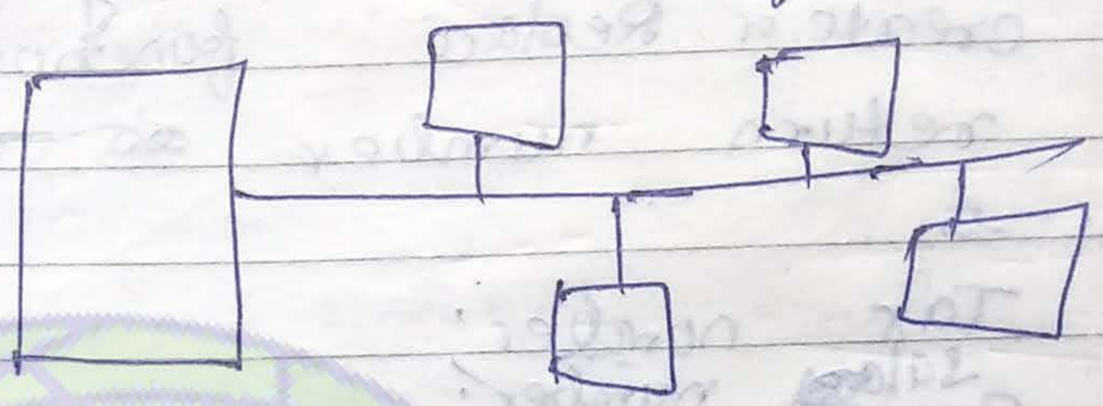
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Client - Server - Architecture

Date: / /
Page No:

* 2 - Tier client server architecture for DBMS -

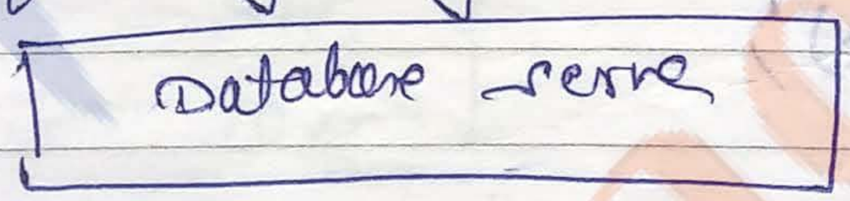


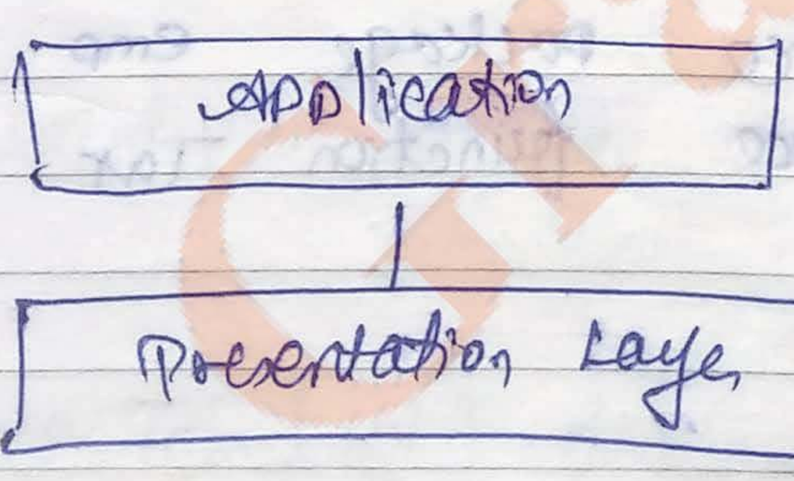
In this type of architecture the database is stored on the server machine, the query and transaction functionality is provided by the server. The User interface programs can run on the client side when DBMS access is required the program establishes a connection with the server.

(11)

* 3 - Tier client server architecture -

The 3-tier client serva architecture consists of the following layers





(1) The Presentation layer provide the user interface and interacts with user. The

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

and forms to the client in order to interfa
this log use, input, and output by accepting
user ~~com~~ and then the ~~re~~ f

(II) Application entry logs - Programs the application logic
here queries can be formulated based on
the ~~d~~ ~~progr~~
user input from the client.
Here queries can be formed based in
user input from the client

* additional application functionality such as -
(i) security check
(ii) Identity verification

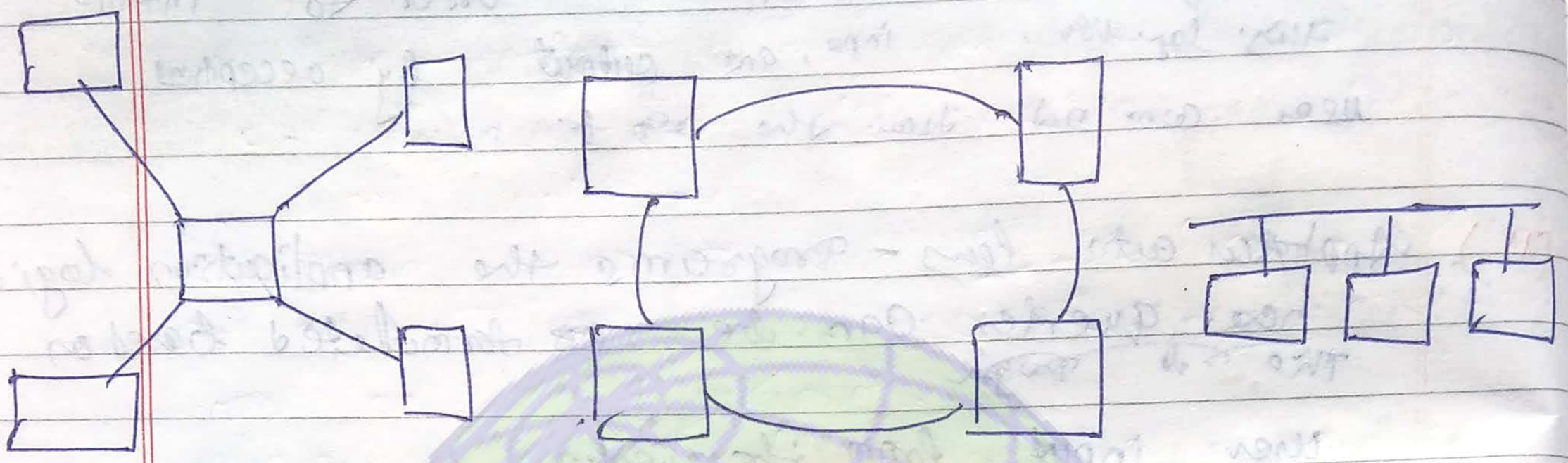
(III) * Database server
This layer handle ~~re~~ query of
update requests from the app. layer,
Process these request and sends the results back
to the application layer.

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Distributed DBMS

Date : 1 1
Page No :



A distributed DBMS or a distributed database is a collection of multiple logically interrelated database distributed over a computer network.

A distributed DBMS is a software system that manages a distributed database, while making that distribution transparent to the user.

Structure of a distributed Database

* Additional tasks of a distributed database

(i) Distributed directory management
A directory containing information about data in the database has to be maintaines.
A track of data distribution both for data fragmentation and replication has to be kept in this catalog.

(2) Distributes query Processing
The distributed database should have

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

the ability to access remote site, and transmit queries and data among various sites via communication network.

(3) Distributed transaction managements -
It should have ability to device execution strategies for transactions that accers data from more than one site and to synchronise the access to distributed data and to maintain overall integrity of the data.

(4) Replicated Data management -
The distributed DBMS should have the ability to decide which copy of of replicates data item to accers and to maintain consistency among the different copies of replicated data.

(5) Distributed database recovery
It should have the ability to recover from individual site crases and from failures such as failure of a communication link

* Types of Distributed databases systems (DBMS)

Distributed database may be of different types depending on the degree of homoginity of the DBMS software. If all the servers use identical software, then the distributed DBMS is homoginious, otherwise it is heteroginious

WWW.GRADESETTER.COM

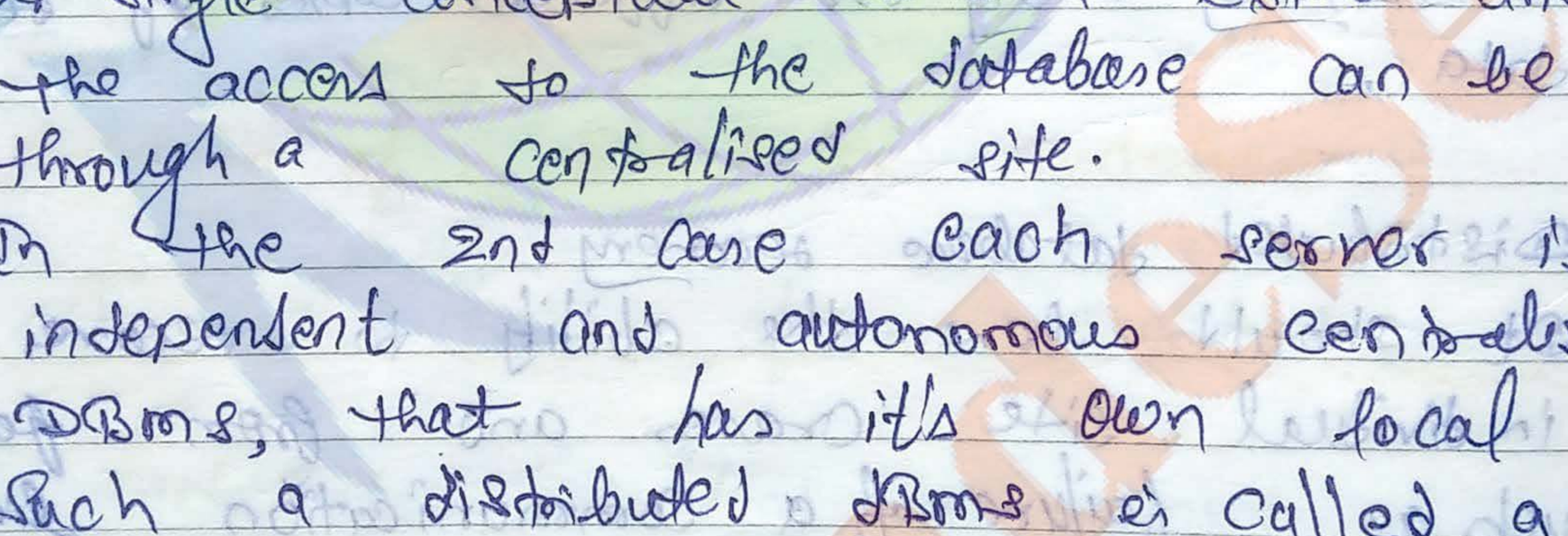
WWW.GRADESETTER.COM

Another factor which can be used to distinguish distributed DBMS is the degree of local autonomy.
If there is no provision for a local site to act as a stand alone DBMS, then the system has no local autonomy.
whereas if a local transaction is permited to have direct
then the system has some degree of local autonomy.

In the first case we have a distributed DBMS, that looks like a centralised DBMS to the user.

A single conceptual schema exist and all the access to the database can be obtained through a centralised site.
In the 2nd case each server is an independent and autonomous centralised DBMS, that has it's own local users such a distributed DBMS is called a federated distributed DBMS.

19
21



SARAA

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

* Advantage of distributed database -
(i) ↑ Reliability → (Increase
(ii) ↑ availability
(iii) ↑ speed
(iv) ease of expansion

* Disadvantage of distributed database.
(i) Difficult to develop software
(ii) costly (software development cost)
(iii) bugs (greater potential for bugs.

★ Design of a distributed database -

- Replication - when you store same copy at different
- Fragmentation

(1) Replication -
The system maintains several identical copies of a relation and each copy is stored at a different site.

Advantage - ↑ availability, ↑ speed

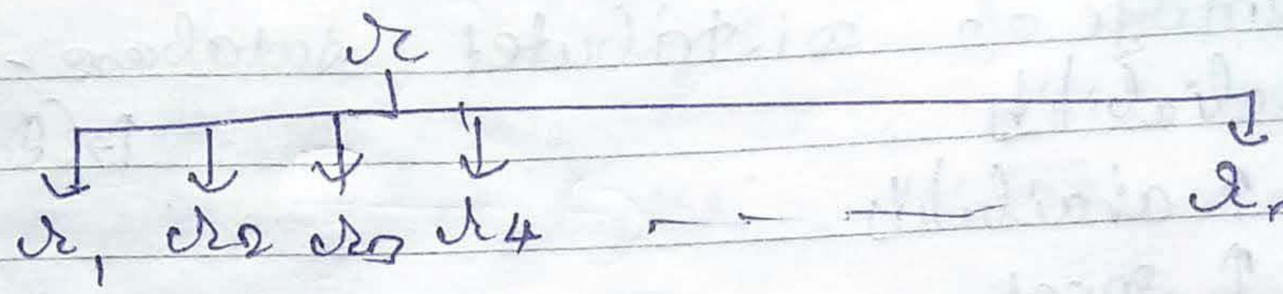
Disadvantage - Increase overheads on update

(2) Fragmentation -
In fragmentation a relation is divided into a number of relations or a number of sub-relation and each sub-relation is stored on different sites.

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :



Type of fragmentation

Horizontal fragmentation (divide on the basis of rows)

vertical fragmentation (divides on the basis of columns)

Student

RNo	Name	FName	Add	class	TFee	Fee_dept	m1	m2	m3

B. Tech-1
B. Tech-2
B. Tech-3
B. Tech-4
} Horizontal fragmentation

The Relation R is partiones into a number of subsets $R_1, R_2 \dots R_n$ each subset consists of a number of touples or rows of relation R. each touple of relation is must belong to one of the fragments. so that the original relation can

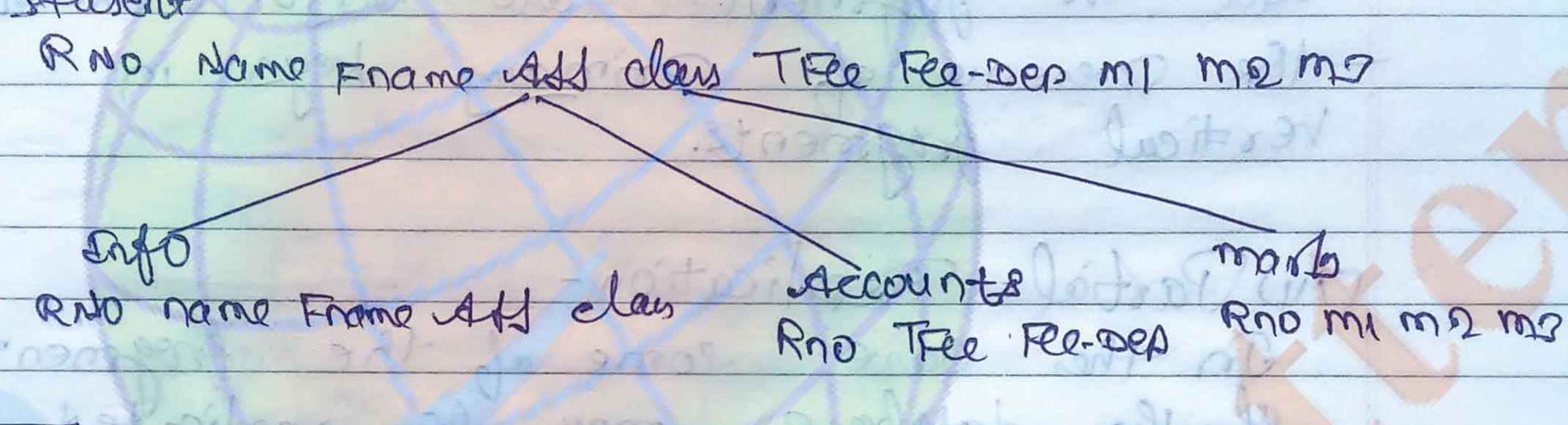
SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

be re-constructed if required.

Vertical Fragm -
It divides a relation vertically by columns or attributes. a vertical fragment of a relation will contain only certain attributes of the original relation.



m1	m2	m3

Data Allocation -
Each fragment or each copy of a fragment must be assigned to a particular site in a distributed database system
This process is called data distribution or data allocation.

It can be of following three types -
(i) Complete replication
we will not divide the relation into fragments. we keep copies of the different sites.

WWW.GRADESETTER.COM

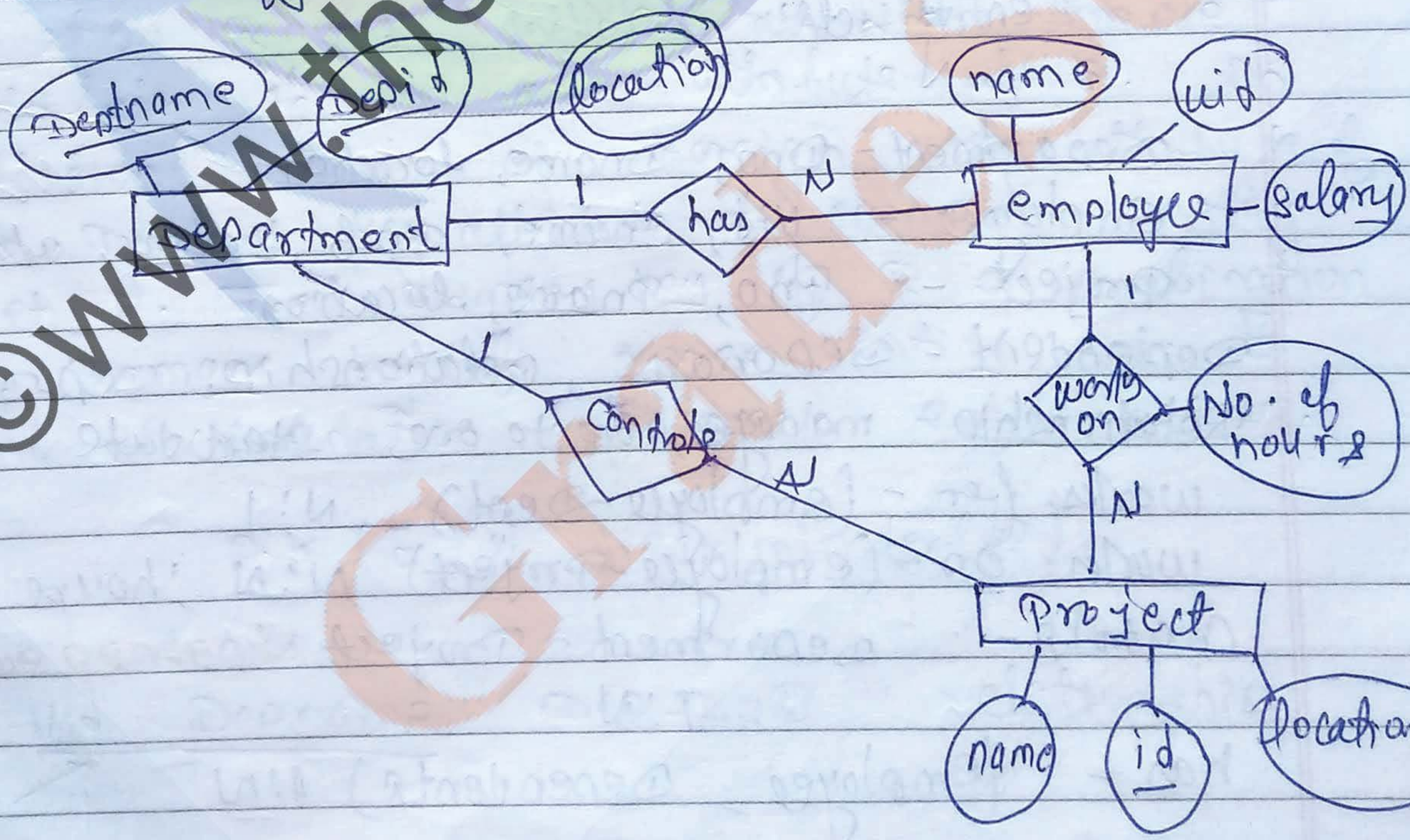
WWW.GRADESETTER.COM

Date : / /
Page No :

A department may have several locations. A dept control different project which have unique name, id and single location, we store each employee name, id, address, salary and job. An employee is assigned to one department but may work on several projects which are not necessarily controlled by single department we keep track of no. of hours per week, employee works on each project. we also keep track of direct supervisor of each employee. we each employee. we store department name, DOJ, relationship with diagram for above mentioned organization.

- Identify entities
- Identify P.K
- Identify relationship

*



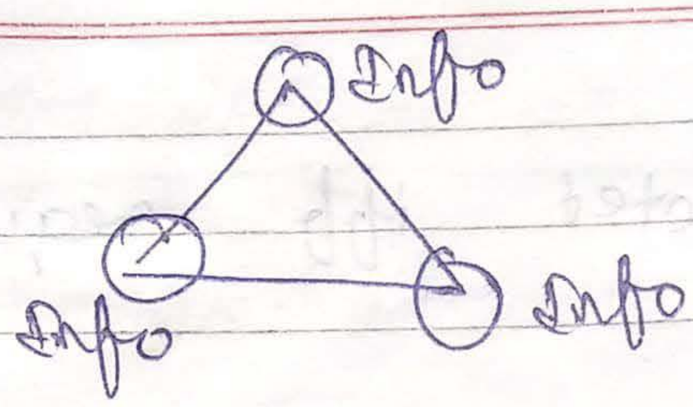
©www.thecompanyboy.com

SARAA

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

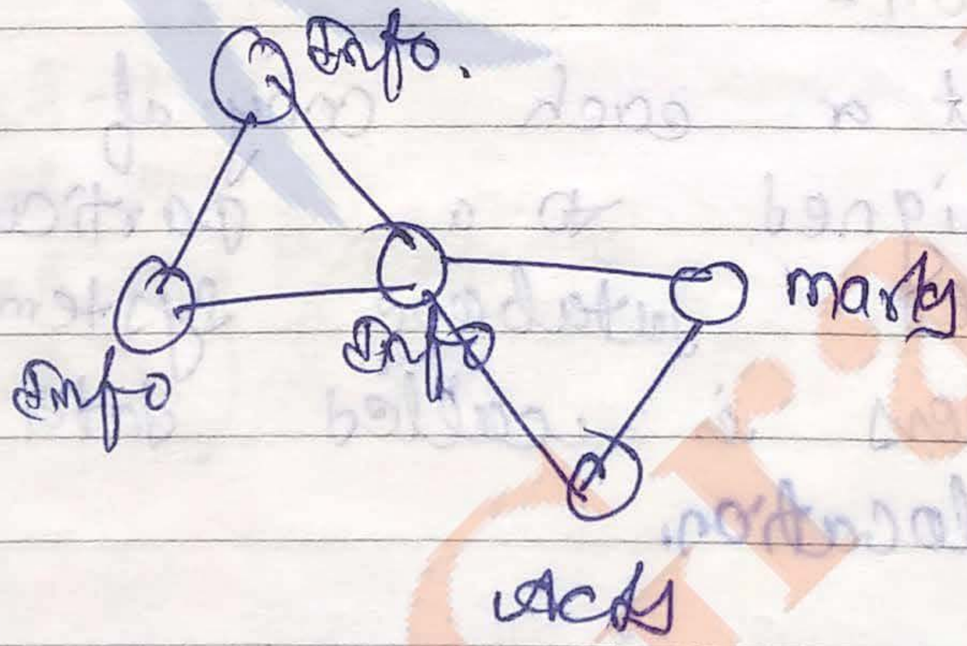
Date : / /
Page No :



(b) Complete frogmentation or no replication

In this case the database is divides into different fragments and each fragment ~~is~~ stored at exactly one site, all the fragments are disjoint except the presence of primary keys in vertical fragments.

(iv) Partial Replication-
In this case some of the fragments of the database may be replicated while the others may not be replicated.



WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

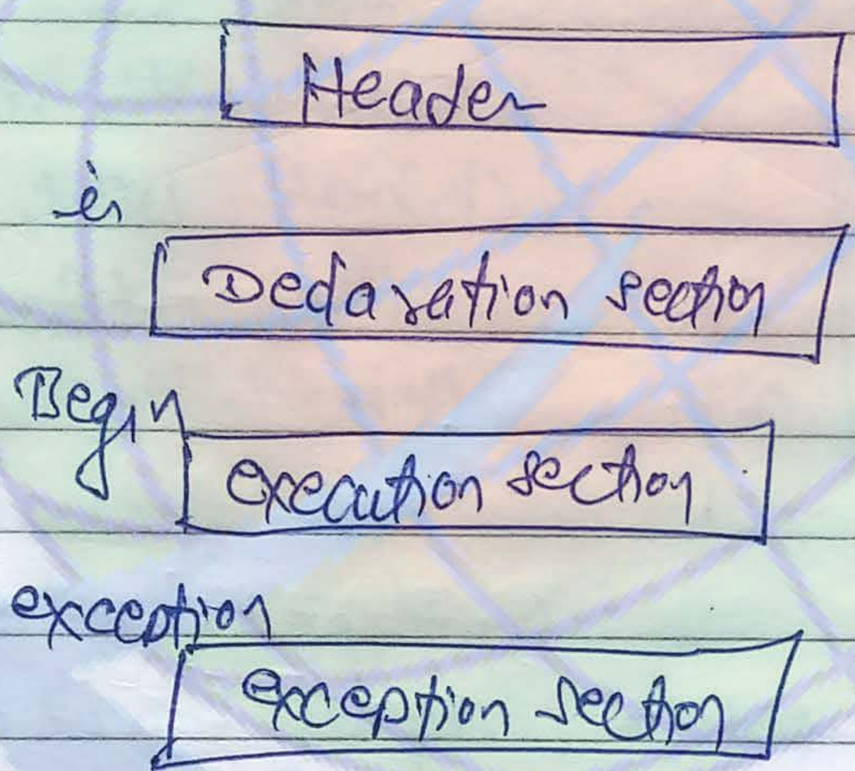
(1) %TYPE -

%type is used to declare a "field" with the same type as that of a specified table's column

%ROWTYPE

%ROWTYPE is used to declare a "record" with the same type as found in the specified database table, view or record

(2) PL/SQL block structure



WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Transactions:

Date : / /
Page No :

A transaction is a logical unit of database process.
It may contain number of read or write operations.
eg.

withdraw money - single transaction.

```
read_item(x)
write_item(x)
x := x - N;
```
} single transaction.

=

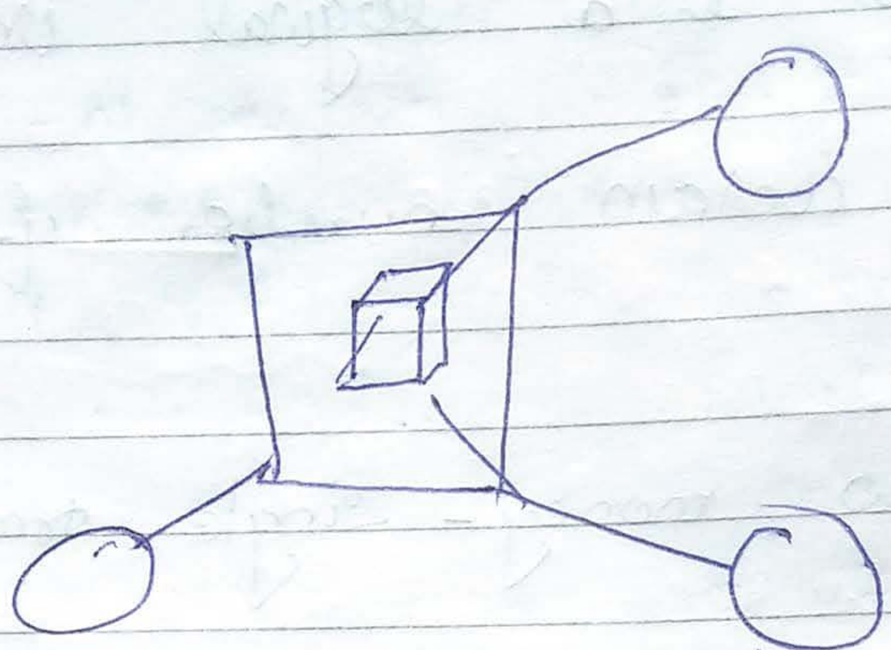
```
read_item(x)
X := X - N;
write_item(X);
read_item(y);
Y := X + N;
write_item(Y);
```
} Balance transfer. single transaction

- Database system =
  (i) Single user system
    ↳ Of only one user is able to access database
  (ii) Multi. User database system:
    Of multiple user are able to access the database at the same time then it is multiuser database system
    "multiuser database system gives rise to concurrency".

- The Problem of
  Concurrency occurs when two or more user's try to access the same database object at the same-time.

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM



Following are the problem, that may be caused by concurrency

(1) Lost - Update Problem -

T_1	T_2
read-item (x) x=80 N=10 (10)	
x := x - N;	
	read-item (x) 80
	x := x + m; m=20 (100)
write-item(x); (70)	
Read-item (x);	
y := y + N;	
write-item(x);	write-item(x); (100)

the update have been lost

The lost update problem occurs when two transactions, that access the same database item have their operations internally in away that makes the value of that database items incorrect

SARAA

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

(ii) Temporary Update (Dirty Read)

This problem occurs, when one ~~transaction~~ Update a database item and then the transaction fails. In the mean time the updated item is accessed by another transaction before the changes made by the first transaction are undone.

T_1	T_2
read-item(x)	
x := x - N	
write-item(x),	
	read-item(x),
transaction fails { abort T1;	x := x + m;
	write-item(x);

(iii) Incorrect summary Problem.

If one transaction is calculating an aggregate summary function on a number of records, while another transaction is updating some of these records,

The aggregate function may calculate, some values before ~~they~~ are updated and others after they are updated. this may lead to incorrect summary.

T_1	T_2
	sum := 0',
	read: item(A')
read-item(x)	

SARAA

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date :
Page No :

x := x - N
write_item(x);

read_item(x);
sum := sum + x;
read + item(4);
sum := sum(4

ACID Properties :-

There are certain Properties that a transaction should possess, they are known as acid Properties of a transaction.

(i) Atomicity - A transaction is an atomic unit of processing, it is either performed in its entirety or not performed at all.

"दूसरे कोटा नहीं हो सकता"
"या तो हुआ या नहीं हुआ"
- Partial not allowed.

(ii) Consistency Preservation - A transaction is consistency preserving if its complete execution takes the database from one consistent state to another.

(iii) Isolation - A transaction should appear as though it is being executed in isolation from other transactions. i.e. other transaction should not have an affect on the execution of the transaction.

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

(iv) Durability / Permanence.

The changes applied to the database by committed transaction must persist in the database.

☆ Schedule:-

When transaction are executing concurrently in a interleaved fasion, then the order of execution of operation from the various transaction is known as a schedule.

↑ schedule of n-transaction

S

$T_1, T_2, T_3, \dots\dots T_n$

is an ordering of operations of the transaction subject to the constitute that for each transaction 'Ti' that participates in 'S'. the operation of Pi in S must appear. In the same order in which they occur in Ti!

However the operation from other ban' can be interleave with the operations of Pi'.

S

$T_1, T_2, \dots\dots T_n$

Ti

S: $R_1(x)$; $R_2(x)$; $W_1(x)$; $R_1(y)$; $W_2(x)$; $W_1(y)$

conflicting operations

Conflicting operation:-

Two operations in a schedule are said to conflict if they satisfy the following three conditions -

SARAA

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

(a) They belong to different transaction
(b) They access the same database item.
(c) Atleast one of these operations is a write operation.

* Serial schedule-
A schedule 's' is serial if for every transaction 'Ti' participating in 's' all operations of 'Ti' are executed consequitively in a schedule. otherwise the schedule is said to be non serial.
Here,
In a serial schedule, only one transaction is active at a given time.
The commit point of an active transaction initiates the execution of next transaction.

No interleaving occurs in a serial schedule.

* Serializable schedule:-
A schedule 's' of n-transaction is said to be serializable if it is equivalent to some serial schedule of same n-transaction.
For the two schedule to be equivalent, the operation apply to

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

their data item affected by the schedule. should be

in the both the

Two definitions of equivalence are view and conflict equivalence, blue equivalence

☆ View equivalence.

Two schedules S and S' said to be view equivalent if the following three conditions are maid-

(i) The same set of transactions participate in S and S' and S and S' include the same operations of those transaction.

(ii) For any operation $r_i(x)$ of T_i if the either r or w

value of x, read by the operation has been written by an operation $w_j(x)$ in schedule S, the same condition must hold for value of x, read by operation $r_i(x)$ in schedule S'.

$r_i(x)$ of $T_i \rightarrow w_j(x)$

S'

S	S'
$w_j(x)$	
$r_i(x)$	

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

(iii) If the operation $w_k(y)$ is the last operation to write item y, in schedule "s". then $w_k(y)$ must be the last operation to write item y in s' also.

example:

A

T_1	T_2
read_item(x)	
x := x-n;	
write_item(x);	
read_item(y);	
y := y+n;	
write item(y);	
	read_item(x)
	x := x+m;
	write_item(x);

serial scheduling

B

T_1	T_2
read_item(x);	
x := x-n;	
write_item(x);	
read_item(y);	
y := y+n;	
write_item(y);	
	read_item(x);
	x := x+m;
	write_item(x);

serial scheduling

C

T_1	T_2
read_item(x);	
x := x+n;	
write_item(x);	
	read_item(x);
	x := x+m;
	write_item(x);
read_item(y);	
y := y+n;	
write_item(y);	

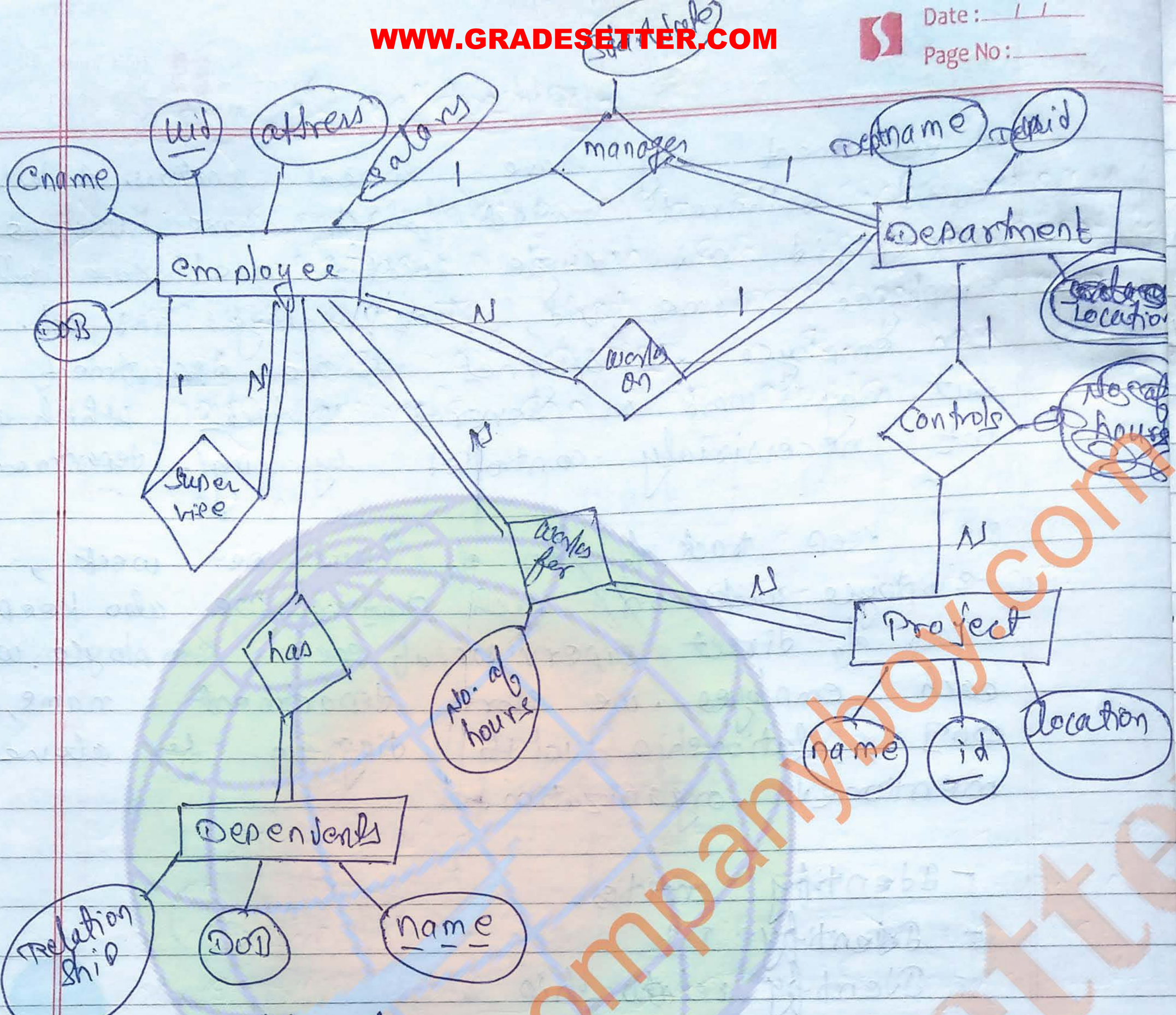
D

T_1	T_2
read_item(x);	
x := x-n;	
	read_item(x);
	x := x+m;
write_item(x);	
read_item(y);	
	write_item(x);
y := y+n;	
write_item(y);	

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :



entity sets :-

Department - Dno, Dname, location
employee - Ssn, Ename, DOB, salary, address
Project - Pno, Pname, location
Dependent - Depname, Relationship, Dep-DOB
Relationship - manages, one to one, start date
works for - (employee - Dept) N : 1
works on - (employee - Project) N : N ; hours
controls - department - Project
1 : N
has - (employee, Dependents) 1 : N

SARAA

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

* Data Dictionary -

A data dictionary is a file or a set of files that contains a database's metadata.

like - data ownership
data relationship
numbe of records in each file
name and type of each field

A: $r_1(x)$, $w_1(x)$, $r_1(y)$, $w_1(y)$, $r_2(x)$, $w_2(x)$;

B: $r_2(x)$, $w_2(x)$, $r_1(x)$, $w_1(x)$, $r_1(y)$, $w_1(y)$;

(equivalent serializable schedule)

C: $r_1(x)$, $w_1(x)$, $r_2(x)$, $w_2(x)$, $r_1(y)$, $w_1(y)$;

D: $r_1(x)$, $r_2(x)$, $w_1(x)$, $r_1(y)$, $w_2(x)$, $w_1(y)$;

Here,

- schedule C is equivalent to serial schedule A
- so, D is not serializable

Conflict equivalence

Two schedules are said to be conflict equivalent if the order of any two conflicting operations is same in both the schedules.
(we can check it by making graph)
The serializability of the schedule can be checked by making a serializability

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

graph, which is a directed graph

$$G = (N, E)$$

that consists of a set of nodes 'N' and set of edges E

$$N = \{T_1, \dots T_n\}$$

$$E = \{e_1, e_2, \dots e_n\}$$

The graph can be made as follow

(i) For each transaction T_i, participating in schedule S, create a node labelled T_i in the precedence graph or serializability graph.

(ii) For each case in S, where T_j executes a read item 'x' after T_i executes a write item x, create an edge from node

$$T_i \rightarrow T_j$$

$(T_i$ to $T_j)$

(iii) for each case in S, where T_j executes a write item x; after T_i executes a write item, create

(iv) For each case in S, where T_j executes a write item x after T_i executes a read item x,

Create an edge from T_i to T_j.

The schedule S is serializable if only if the serializability graph does not contain any cycle.

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

if and only if the serialized graph do not contain any cycles.

☆ Mendatory Access Control:-

In mendatory access control, each data object is labled with a certain classification level and users are given certain clearance level.
A given data object can be accessed only by users with appropriate clearance level.
Typical security classes can be:-

(i) Top secret
(ii) Secret
(iii) confidential
(iv) Unclassifies

where top secret is the highest level of security.

Now we classify each object and each subject into one of the given security classification.
Two restrictions are enforces on data access based on object-subject classification.

(1) A subject S is not allowed to read access on object O

$S \rightarrow O$

Until classification of S is greater than

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

or equal to classification of O.

$$class(s) \geq class(o)$$

This is known as Simple security Property.

Top secret	R_1, R_2	U_{10}
secret	R_5	U_6, U_7
confident	R_3	U_4, U_5, U_8
Unclassified	R_4	U_1, U_2, U_7

(ii) A subject S is not allowed to write an object 'O' ~~unless~~ untill classification of 's' is less than or equal to classification of O.

$S \rightarrow O$

$class(s) \leq class(o)$

This is known as star security property.

Ans

It Prevents the follow of information from top downwards.

* The first restriction is to enforce the rule that no subject can read an object, whose classification is higher than the subject security clearance.

The second restriction Prohibits a subjects from writing an object at a lower security classification.

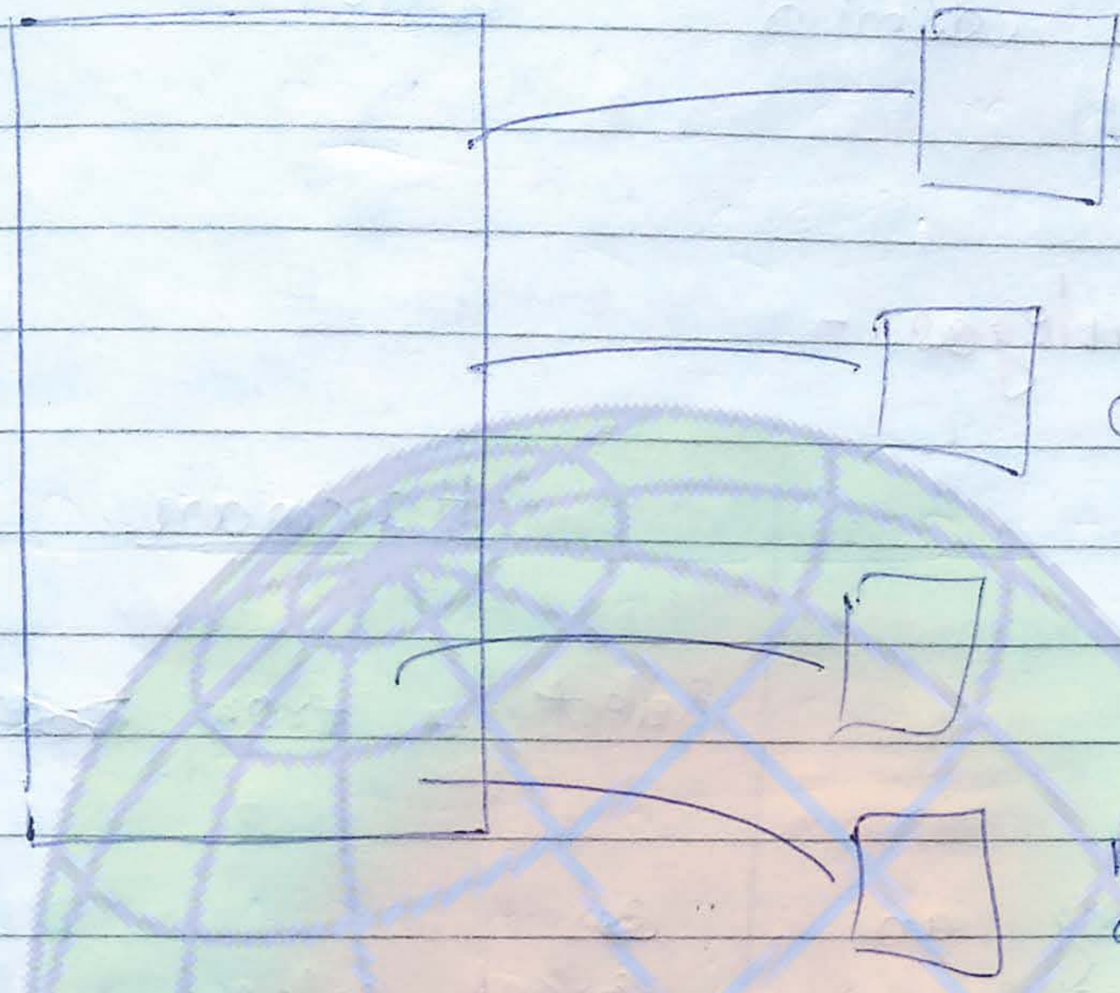
violation of this rule will allow information to flow from higher to lower classification

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

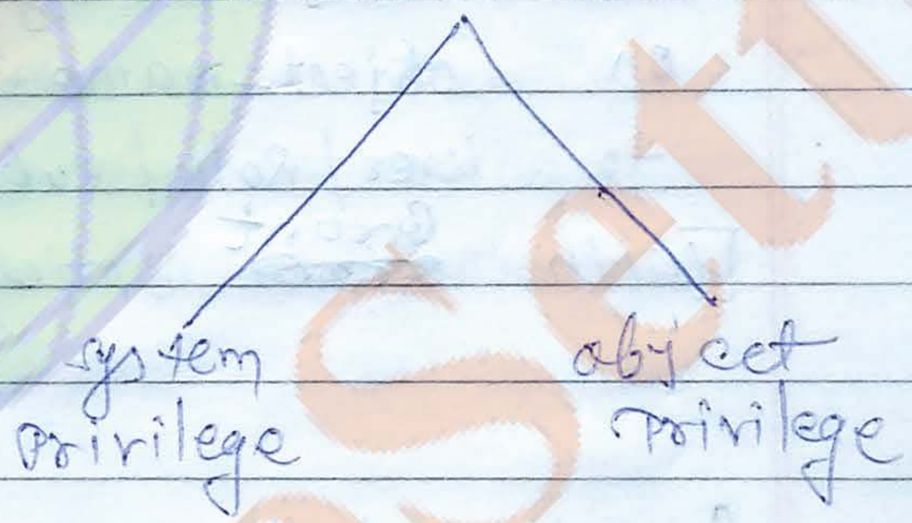
levels :



System

Create user user-name
identified by ****;
password expire;
Create user kanwal
identified by kl27;
Grant Create Table to
kanwal;
Alter user (user_name) kanwal
quota 5m on system;
unlimited.

Privilege



Create Table Info1
(Roll number(4)
Name varchar2(20));

Grant system Privilege to user-name
[with Admin option]



Grant select Any Table to kanwal;

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

Grant create Table to kanwal with admin option.

★ Object Privileges:-

Kanwal	Harman
Info	Select * from ~~Info~~ kanwal.Info
syntax for granting an object Privilege:- Grant object-Privilege / ALL on object_name to user / Role / Public [with Grant option]	Grant select on kanwal.Info to Harman;
Grant Select on Info to Harman;	
Grant Select update(Address) on Info to Harman;	
Grant Select update(cattrs) on Info to ~~to all user~~ Harman / Public / Role_name with admin option;	

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date: / /
Page No:

★ Role based Access control:-

In role based access control, a set of roles are created in the database. Authorization can be granted to individual user each data.

that gives authorises to perform.

#

Create Role Role_name;
Grant all on info to 'student';
Grant select on Fee to student;
Grant select on marks_student;

Grant Role_name to user_name;
Grant student to s1, s2, s3;
Drop user user_name

- Revoke:

Revoke system_Privilege from user_name;
Revoke object_Privilege on object_name from user_name
Drop role Role_name;

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Objects in Oracle

Date : / /
Page No :

```
create Type Type-name as object
(
  var-name | data-type1,
  :
  :
);
```

```
create Type person as object
(
  Id no      number,
  Fname      varchar2(10)
  Mname      varchar2(10),
  Lname      varchar2(10),
  Phone      number );
```

```
Declare
P1 Person;
Begin
P1 = Person(101, 'Hoopreet', 'Singh', 'Sidhu',
            9814828368);
dbms_Output.Put_line('Data of Object P1 is
  ||P1.idno||' '||P1.Fname||' '||P1.Mname||'
  ||P1.Lname||' '||P1.Phone);

End;
```

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

create table table-name of object-type;
Create Table PT of Person;

→ Insert into PT
values (101, 'Harpreet', 'Singh', 'Sidhu', 9814828761);

→ Insert into PT values
(Person(101, 'Harpreet', 'singh', 'sidhu', 98148287 6));

→ select * from PT;

:→ ~~select~~ declare
p1 Person;
Begin
select value(P) into p1
from PT (P) where P.idno = 101;
dbms_output.Put_line('Data of objects p1 e:-)

alias

- drop person Force;

* Alter type Person
add member procedure display cascade;
Alter type Person
Add function return_phone return numbe cascade;

* create or replace type body person
as
member procedure display
is
dbms_output.Put_line(Idno||' '||Fname||' '||Mname||' '

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

'' ||Lname|| '' ||phone);
end;

member function return-phone
return number
is
begin
return phone;
end;
end;

Using PL/SQL block create an object of Person.

Declare
P1 Person ; phn number(10);
begin
P1.display;
phn := P1.returnphone();
dbms_output.put_line('phn':||" '||phn);

~~Select P.return_phone() from~~
declare
P1 Person;

begin
select value(P) into P1 from pt P where
P.idno = 101';
P1.display;
end;

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date: / /
Page No:

* mapping of ER model into relations

- mapping regular entity sets E → R
- For each regular entity 'E' in ER model Create a relation R, that include all the simple attributes of E, include only the simple component attribute & choose one of the key attribute as primary key of relation R.

employee
Uid Fname Mname Lname salary addres

Department
Depid Deptname project id name location

mapping of weak entity set (W)
For each entity type W in ER schema with owner entity type E. Create a relation R and include all the simple attributes of W as attribute of R. In addition, include as foreign key attribute of R, primary key attribute of owner entity set E. The primary key of R is combination of primary key of owner entity set. E and discrimination of weak entity set W.

P.K (E) U discriminator (W)

* Dependents
Vid Depname Dep-DOB relationship

P.K

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date: / /
Page No:

Storing object as a column of a table

```
Create table contacts
(
Contact Person,
Addres varchar2(30));
```

```
Insert into contacts
values( Person(102, 'Harjeet', 'Singh', 'Sandhu', 9815026312),
                '#41, Urban Estate, Patiala');
```

```
declare
  P1 Person;
begin P1:=
insert into contacts values(P1,'---');
end;
```

Not correct (X)

Note

```
select * from contacts;
select c.contact, c.addres from contacts c;(X)
```

```
select c.contact.idno, c.address from contacts c; (✓)
```

- Begin declare
  select P1 Person;
  a varchar2(30);
  begin
  select

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

* Create an object type employe with attributes, Person, Dept and Joining date

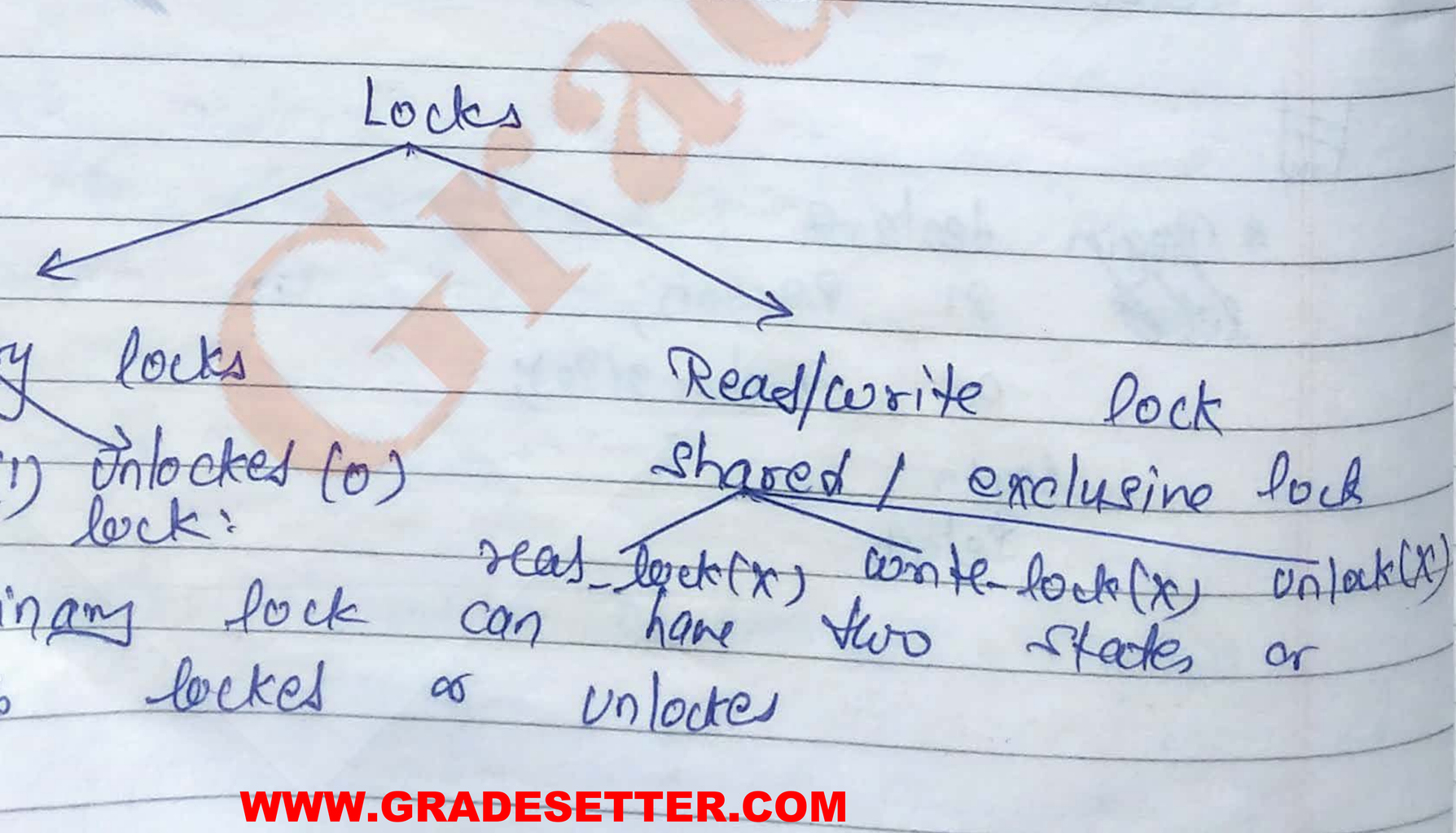
Create the corresponding table to store objects of type employe. also show how to insert an select value from this table.

* ## Locks

Locking data items prevent multiple transactions from accessing the data items concurrently.

A lock is a variable associated with a data item, that describes the states of the item with respect to various operations that can be applied to it.

Generally there is one lock for each data item in the database.

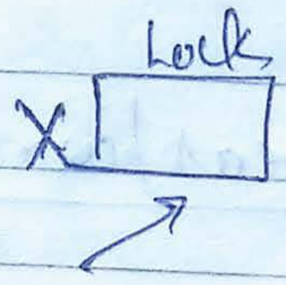


(1) Binary lock:
A binary lock can have two states or values locked or unlocked

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date: / /
Page No:



a distinct lock is associated with each database item 'X', If the value of the Lock on 'X' is '1' item 'X' can not be accessed by a transaction.

If the value of the lock on X is Zero(0), the item can be accessed by the transaction after it changes the value of lock from 0 to 1.

two operations:

lock-item

Unlock-item

are used in binary locking. A transaction requests access to an item 'X', by first issuing a lock item(X) operation.

If

Lock(X) = 1 , the transaction will have to wait

If

Lock(X) = 0 , then it is set to '1' and the transaction is allowes to access the database item X.

When the transaction is through using the item it issues

Unlock-item (X) → Lock (X) = 0,

Unlock-item(X) operation which sets lock(X) = 0 and it is available to other transaction to acces

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

(2) Shared/ exclusive or read/write lock :-

The binary locking system is too restrictive for database items because atmost one transaction can hold a lock on a given item. we should allow several transactions to access the same item 'x', If they all access 'x' for reading purpose only.

However,

If a transaction wants to acquire an item for writing purpose then it must have exclusive access on that data item.

The locks that use this mechanism are called read/write lock or shared exclusive lock.

There are three locking operations :-

(i) read_lock(x)
(ii) write_lock(x)
(iii) unlock(x)

A lock associates with an item 'x', can have three possible state

~~(i) read_lock(x)~~

(i) read-locked
(ii) write-locked
(iii) unlocked

A read-locked item is also called shared locked because, other transactions are also

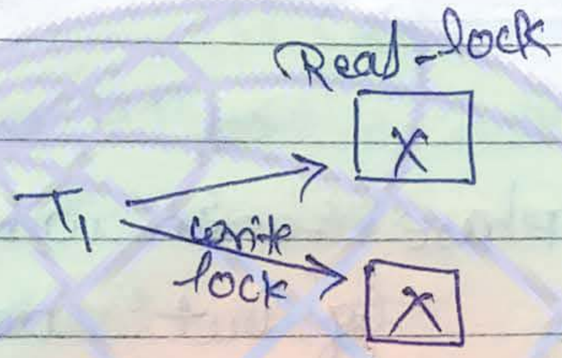
SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

allowed to read_lock and item at the same time, whereas a write lock item is exclusivly locked because only a single transaction can acquire a write lock at any given time.

Conversion of locks :-



A transaction that already holds a lock on item x is allowed under certain conditions to convert the lock from one lock state to another.

It is possible for a transaction T_i to acquire a Read_lock on item X, and then later on to upgrade the lock by issuing write_lock(X) operation, If no other transaction is holding a read lock on X, the lock can be upgrades, otherwise the transaction will have to wait.

It is also possible for a transaction to acquire a write_lock on X, and then to downgrade the lock by issuing read_lock(X) operation.

Two Phase locking :-

A transaction is said to follow, two phase locking protocol.

If all locking operations proceed the first unlocking operation.

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Such a transaction can be divided into two phases:-

(i) Growing phase:
During which new locks can be required, but no lock can be released.

(ii) Shrinking phase:- In which existing locks can be released but no new lock can be required.

If lock conversion is allowed, then upgrading of locks can be done during the growing phase.
and downgrading of locks can be done during shrinking phase.

*

(i) Conservative two phase locking
(ii) Strict " " "
(iii) Rigorous. " " "

*

(i) Conservative two phase locking:-
It requires a transaction to lock all the data item's it wants to access before the transaction begins execution by pre-declaring its read_set and write_set.
If any of the pre-declared item's needed can not be locked the transaction does not lock any item and waits until all the

SARAA

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

items are available for locking.

(II) strict two phase locking :-

In strict two phase locking a transaction does not release any of it's exclusive locks untill after it commits or aborts.
Hence no other transaction will be able to read or write an item that is written by transaction (T) unless transaction (T) has committed or aborted.

(III) Rigorous two phase locking :-

In Rigorous 2PL a transaction (T) does not release any of it's locks, untill it commits or aborts.

* ~~locks~~

T1	T2
read_lock (X)	
read_lock (Y)	
	read_lock (X)
	read_lock (Y)
write_lock (X)	
	write_lock (Y)

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

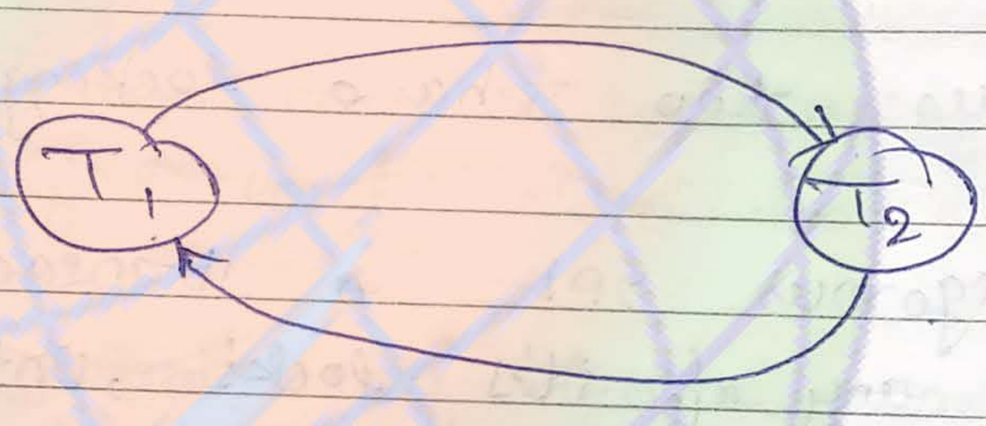
Date : / /
Page No :

Deadlock:-

Deadlock occurs when each transaction (T) in a set of two or more transactions is waiting for some item that is locked by some other transaction T' in the set

$T \longrightarrow T'$

Hence each transaction in the set is waiting for one of the other transactions in the set to release the lock on an item.



Way to deal with deadlock:-

(i) Deadlock Prevention
(ii) Deadlock detection and Recovery.

(i) Deadlock Prevention:-

We can use deadlock Prevention Protocol to ensure that a set of transactions do not enter the state of deadlock.

following Protocols can be used for deadlock Prevention

(a) The first deadlock prevention protocols requires that every transaction locks all the

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date: / /
Page No:

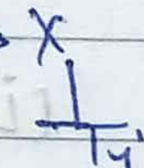
T)

items it needs in advance. If any of the database items can not be obtained then none of the items are locked, rather the transaction waits for some time period and then tries again to lock all the data items, it requires.

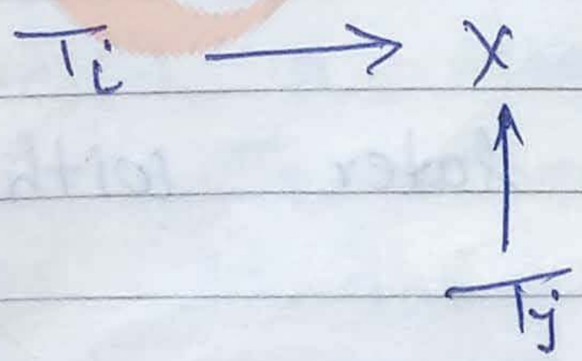
(b) No waiting and cautious waiting Protocols:-

In no waiting Protocol, if a transaction is unable to obtain a lock, it is immediately aborted and then re-started after a certain time delay without checking weather a deadlock will actually occur or not. In cautious waiting it was tries to reduce the number of needless aborts and restarts.

Suppose there is a transaction(T_i) that wants to acquire a lock on database item(X) which is lock by some other transaction T_j with a conflicting lock.



The cautious waiting Protocol states, if T_j is not blocked, then block T_i, else abort T_i



WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

(c) Time Stamp:

A time stamp is a unique identifier assigned to each transaction it is based on the order in which the transactions are started.

If T_1 starts before T_2, then time stamp of T_1 shall be less then T_2.

$$T_1 < T_2$$

$$TS(T_1) < TS(T_2)$$

i.e. the older transaction will have smaller value of time stamp.

Suppose Ti wants to acquire lock on database item X, which is currently acquires by Tj.

Then the rules followed by thei protocols are as follows:-

(i) wait-Die

If the $TS(T_i) < TS(T_j)$,

then

Ti is allowed to wait otherwise abort Ti.

and re-started later with the same time stamp.//

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

mapping of relationship

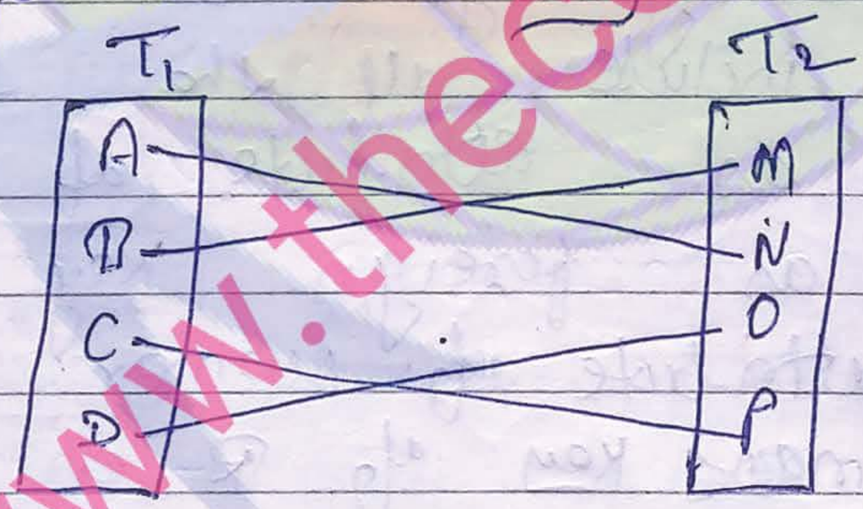
1.) **1:1 Binary relationship** - For each binary 1:1 relationship R, in ER schema, Identify relations S and T that correspond two entity types participating in R. then one following approache can be adopted.

(a) Foreign key approach - Choose one of relation - say S and include as foreign key in S, the primary key of relation T.

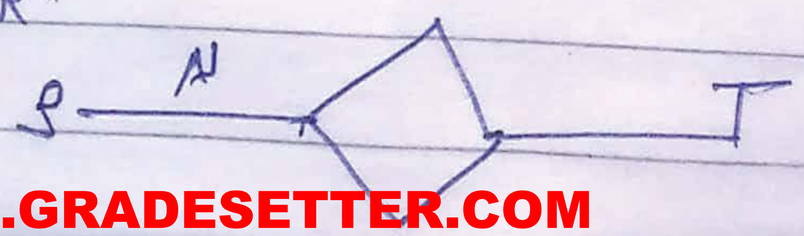
Department -

Dno Dname mgr-vid start-date

(b) merge Relation approach - This approach is used when participation of both entity types in relationship is total. In the approach we merge two entity types into engle Relation.



2.) **mapping of binary 1:N relationship** - for each binary 1:N relationship type R, Identify the relation S, that represents participating entity type at N-side of relationship, Include as foreign key in S, primary key of relation T that represents other entity type participating in R.



SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date: / /
Page No:

(ii) Wound_wait

In this if

$$TS(T_i) < TS(T_j)$$

then abort T_j. and re-started later with the same time stamp, other wise T_i is allowed to wait.

(11) Deadlock detection and Recovery:-

In this case, we allow the system to enter in state of deadlock, and then try to recover using a deadlock detection and recovery scheme.

Deadlock detection:-

Deadlocks can be described preisely in terms of a directed graph called wait-for graph. the graph is represented as

$$G = (V, E),$$

where V is the set of vertices and E is the set of edge.
the set of vertices consists of all the transactions in the system. when a transaction T_i, requests a dataitem currently held by transaction T_j, then an edge from T_i to T_j is inserted in the wait for graph.
the edge is removed only when the transaction T_j releases the lock on dataitem

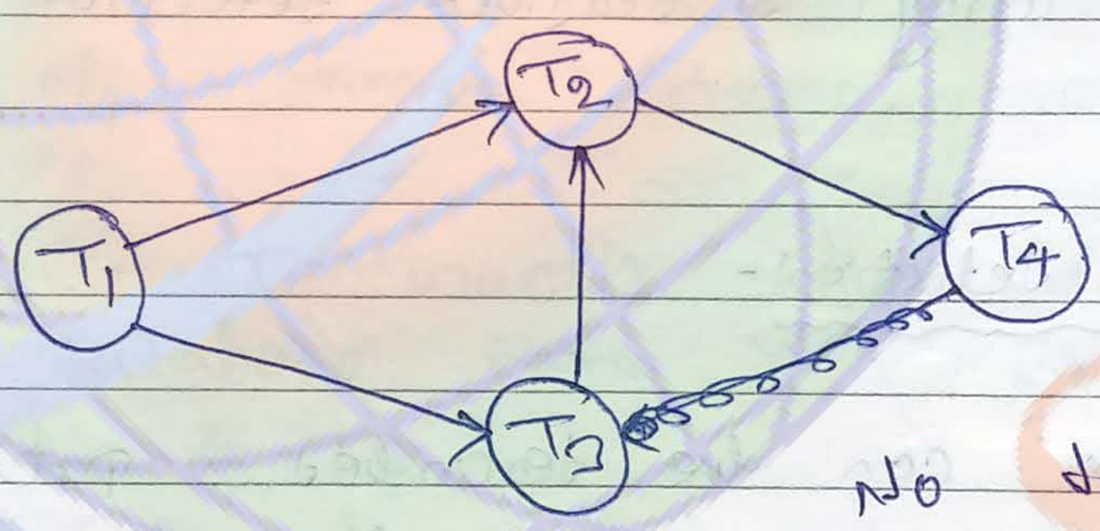
SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

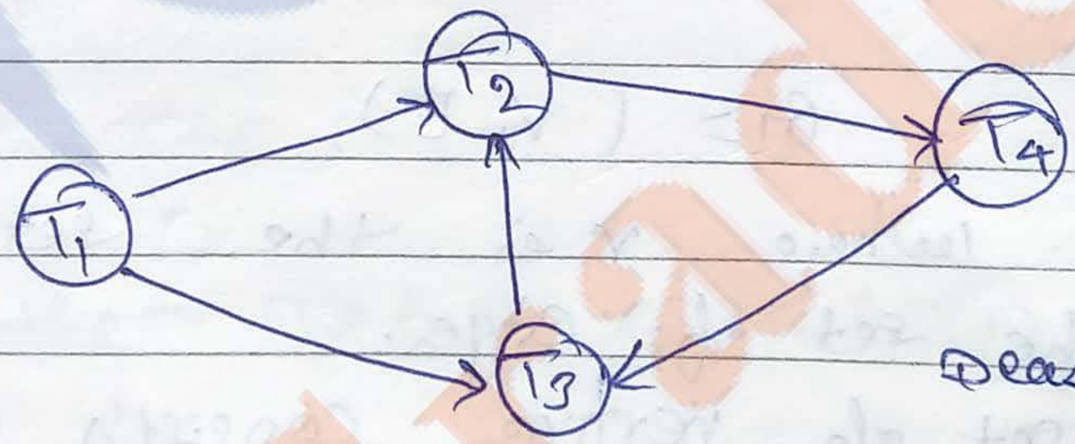
Date : / /
Page No :

i.e. required by Ti.
A deadlock exists in the system, if and only if the wait for graph contains a cycle.

Each transaction involved in the cycle is said to be deadlock, to detect deadlocks the system needs to main a wait-for graph and periodicly invoke an algorithm that searches for a cycle in the wait for graph.



No deadlock



Deadlock

Deadlock detection:
When detection algorithm determines that a deadlocks occurs, then system must recone

the most common action is to role back

SARAA
WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date: / /
Page No:

one or more transactions in order to break the deadlock.

(1) Selection of victim:-
Given a set of deadlock transaction, it has to be determines that which transaction will be aborted to break the deadlock.

Normally the system chooses that transaction, which will incur minimum cost.
many factor's may determine, the cost such as the transaction time the no. of database item's used, the no of database item's still required.

(2) Roll back :-

(3) Starvation:-

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

"Trigger"

A trigger is a code that executes automatically, when a particular database event occurs,

Database triggers are different from procedures and functions.

you have to invoke a procedure or a function explicitly but are database triggers implicit when a table "d" a database trigger is to a

Type of trigger
- statement trigger
- Row trigger
  - (i) before
  - (ii) after

Row
before
after
statement

(1) Trigger:

```
Create or Replace trigger trigger_name
{Before/After}
{ Delete, Insert, Update [of col] }
on Table_name
for each row [when condition];
[ Declare
  =
  =
  -
  Begin


  end;
```

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

Create Trigger message
After Insert
on Info
Begin
dbms_output.Put_line('OK Done')
End;

Create Trigger Upper
Before Insert
On Info
for each row
Begin
:new.name := Upper(:new.name)
end;

* alter Trigger Trigger_name Enable/Disable

* If Updating
then
op := 'Update';
end if;

* Create or Replace Trigger Day
before insert or Update or delete
On Info
begin
if rtrim(to_char(sysdate, 'day')) = 'sunday'
then
raise_application_error (-20000, 'can not Access DB on sunday');
end if;
end;

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

* to_number(to_char(sysdate, 'HH24')) > 17

write a trigger that does not allow user to access table info before 9 am and after 5 pm

(1) Types of security:-

(a) various legal and ethical issue
(b) policy issues.

(a) Right to access certain information
~~Public~~
└ various legal and ethical issue.

(b) Public information
└ Policy issue

(c) system level security
└ system related issue
(hardware level, O.S level, DBMS level)

(d) multiple security level
└ top secret, secrete, confidential, user based
└ The need to identify multiple security levels

(2) Control measures:-
✓(a) Access control
✓(b) Inference control -
(c) Flow control
(d) Data encryption

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

Date : / /
Page No :

3) Database security and the DBA:-

↓
"Database administrator"

Granting and revoking =

(a) Account creation — Acces control
(b) Privilege granting - } discretionary
(c) Privilege revocation -
(d) Security level assignment - mendatory authorization.

WWW.GRADESETTER.COM

WWW.GRADESETTER.COM

* 3. Mapping of binary m:n relationship:-

For each binary m:n relationship type are create a new relation S

$R \rightarrow S$

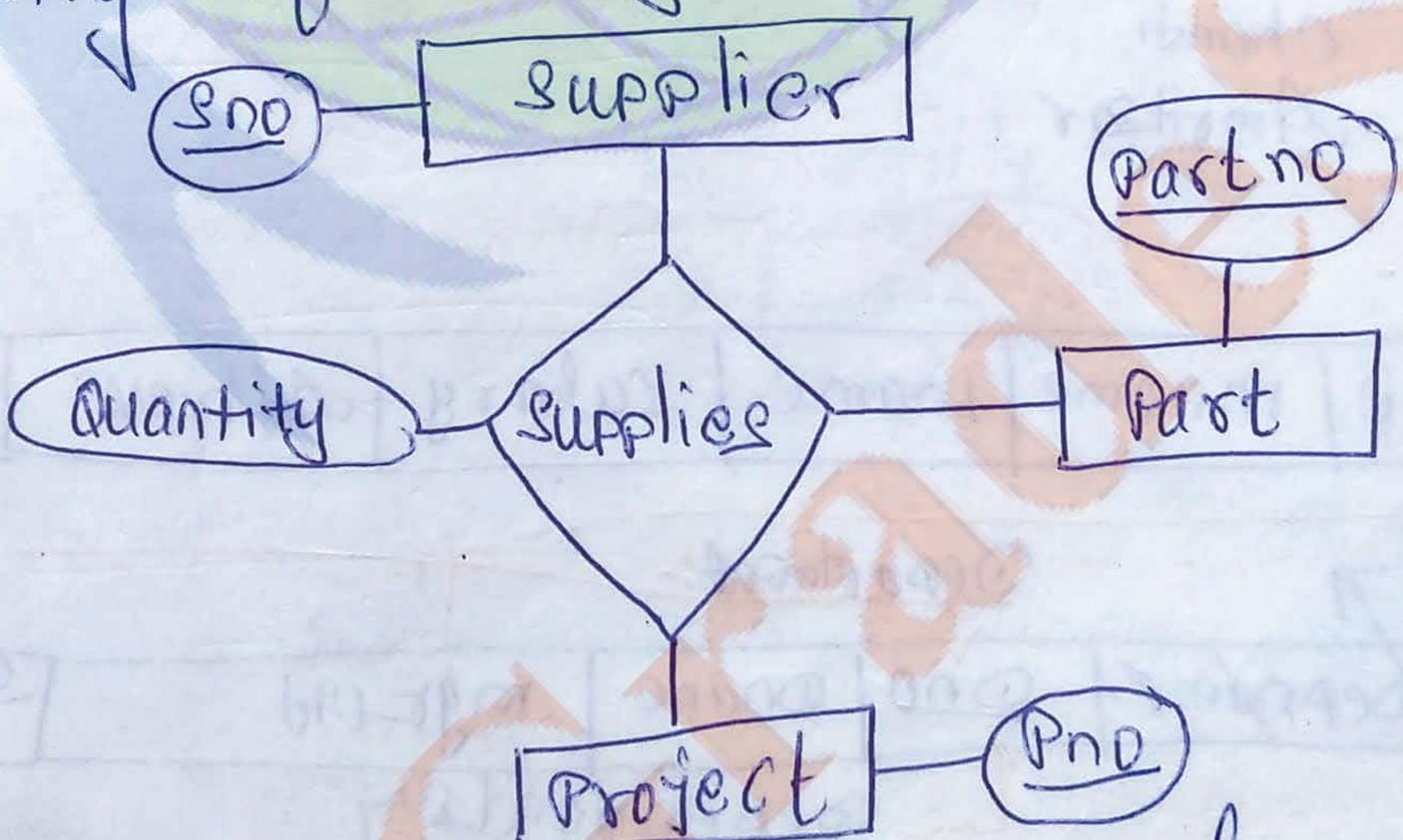
Includes as foreign key attribute in S, the Primary key of the relation, that represents the Participating entity type. there combination were form Primary key of S.

Also include in S, If there is any attribute of the relationship are.

works_on

Vid	Pno	Hours
101	P1	18

* Mapping of N-ary relationship-



For each n-ary relationship, where n is greater than 2, create a new relation to represent R. Include as foreign key attributes. The Primary key of the relation, that represents the Participating

WWW.GRADESETTER.COM