1st class

Date 2/1/18
Page 1

Database managment System :-

Collection of inter related data & a set of application program to access and process the data.

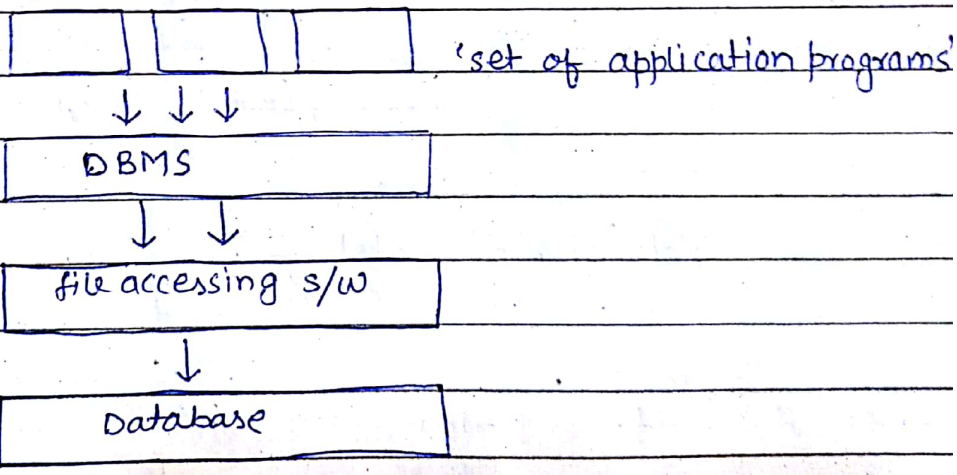
e.g. →

Banking, Railway reservation.

Problem with file system :-

- Security
- Redundancy
- Inconsistency
- Disorganised development

Role of DBMS :-



Data abstraction :-

1. Physical level (implementation)
2. Conceptual level (design level)

3. View level

4. Users level

Data Model :-

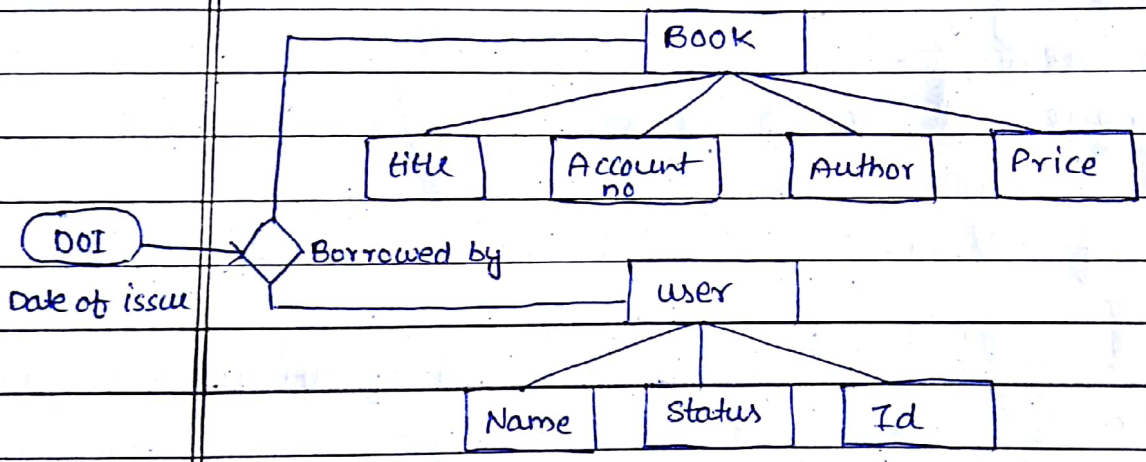
Tool to Conceptulise a database.

e.g. →

Iterative

Recursive.

1. E-R model (Entity relationship)



2. Relationship model :-

Entity are represented by table.

Book	Acc. No.	title	Author	Price

3. Object oriented Model :-

2nd class

Date 08/01/18
Page 3

E-R Model :-

Entity :- Any distinguised object is called a Entity.

e.g. →

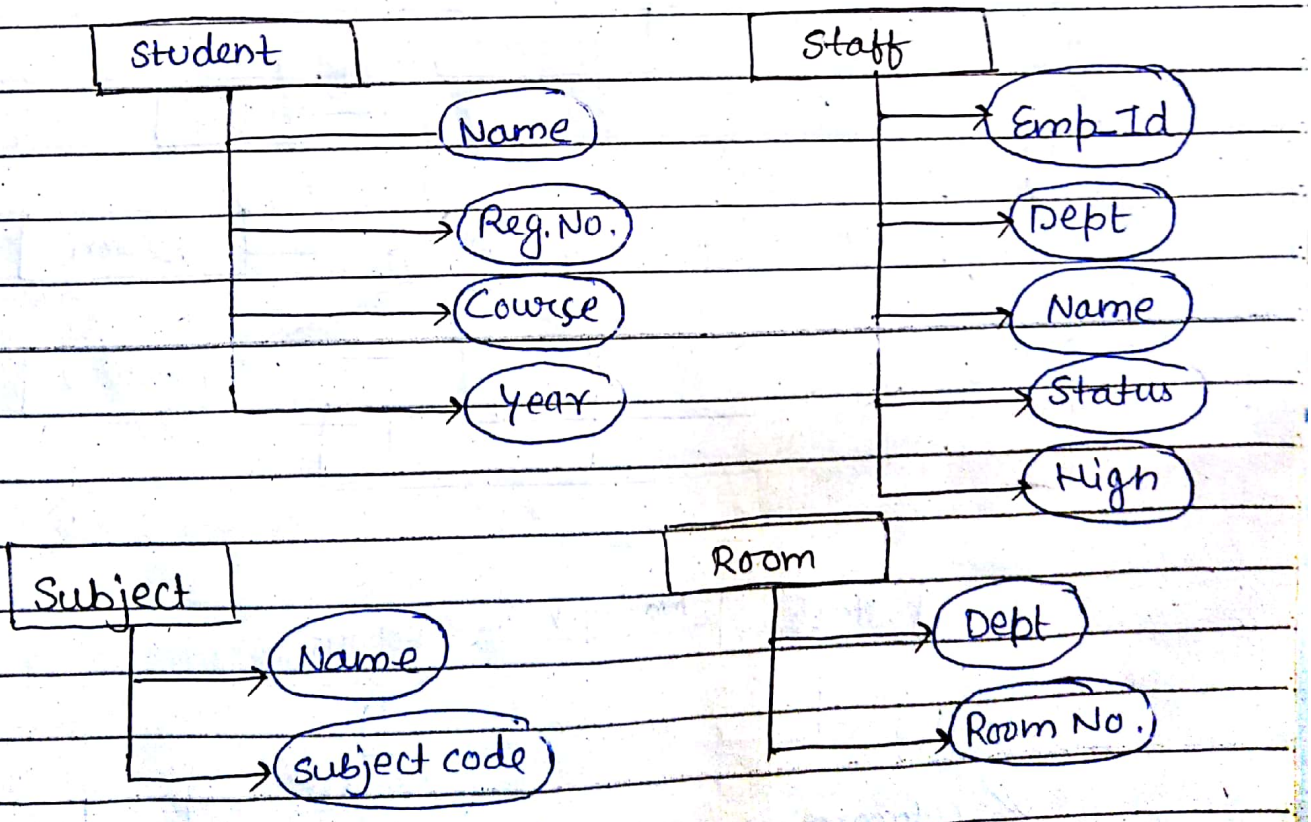
Book details

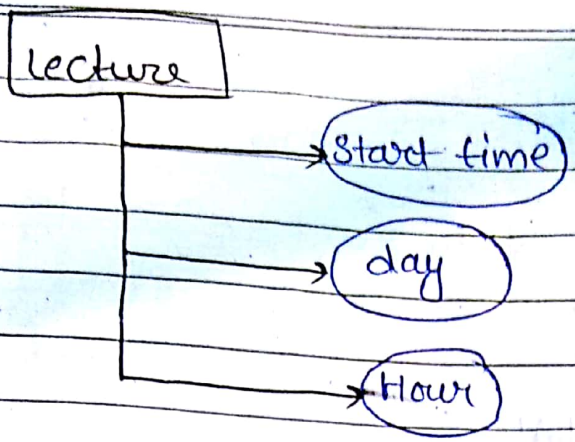
Acc. no.	Book Name	Author	Publication
232356	Computer design	Aho ulman sethi	2017, Pearson India

"Entity of same type form an entity set."

Time table Entity set :-

Name of class | Date of Exams | faculty held Exam | List of Student

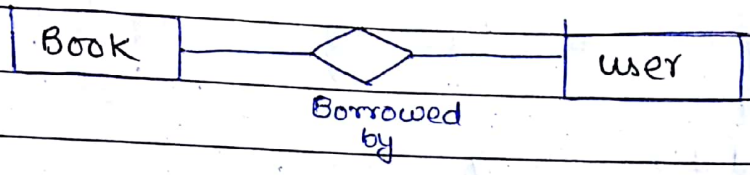


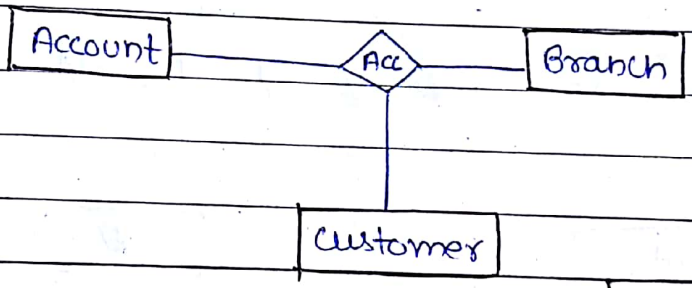


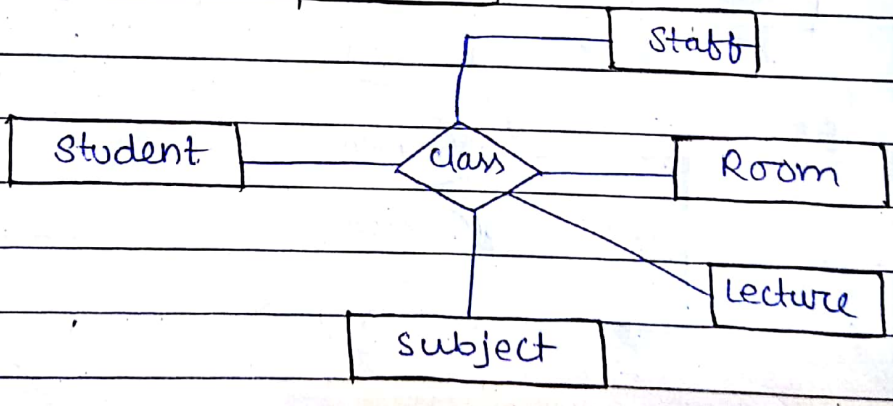
Relationship :—

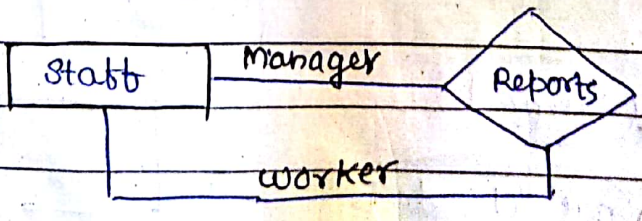
Association between the entity.

e.g.

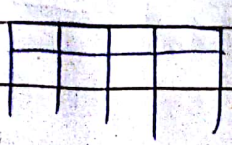


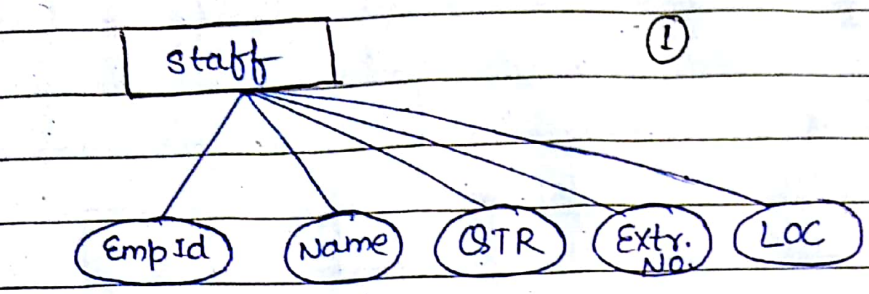




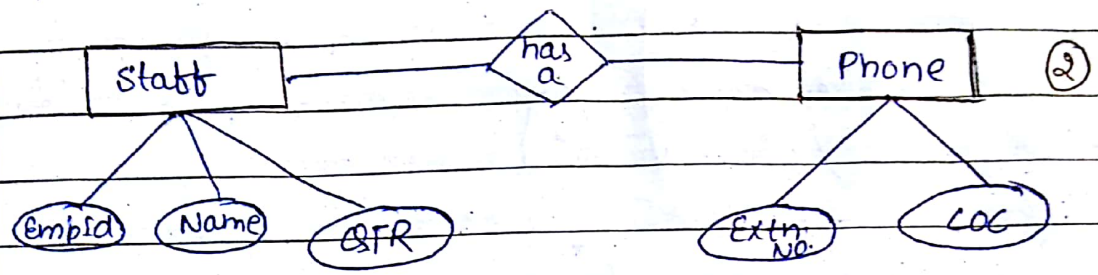


↓ colomn → attribute





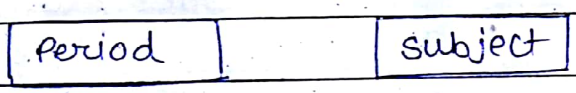
①



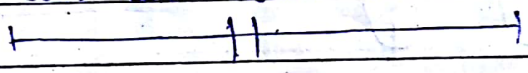
②

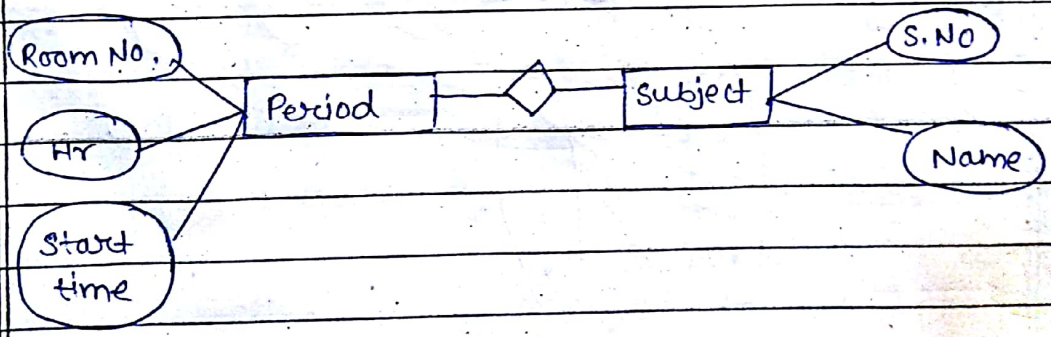
✓ '2' is better than '1'

E-R diagram for time-table :—



can we club this Entity

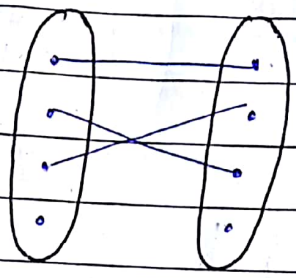




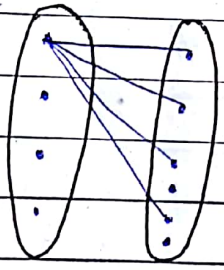
we donot club both of entities to one

Mapping Constraints :-

1. One - One
2. One - many
3. many - one
4. many - many

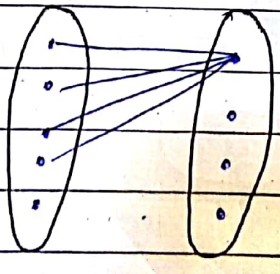


One - One



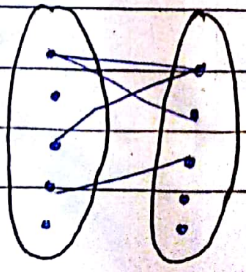
one - many

A B



Many - one

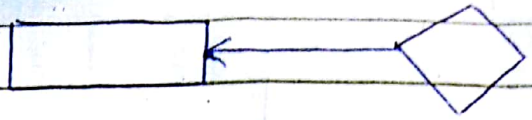
A B



Many - Many

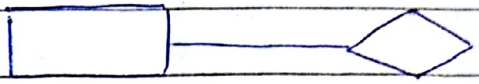
Date
Page 7

*



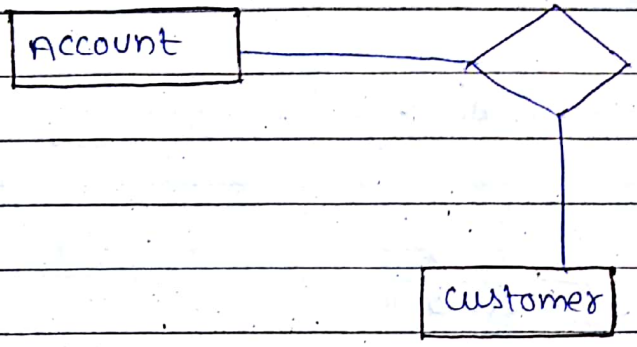
one-one / one-many relationship

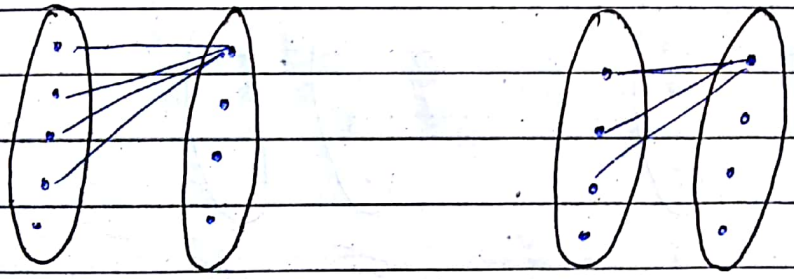
*



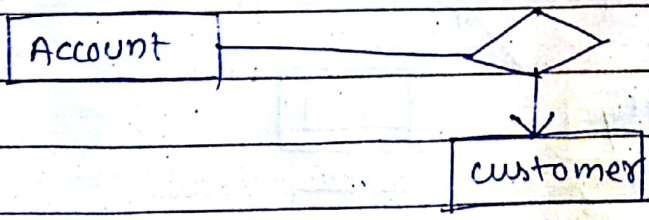
Many-one / many-many

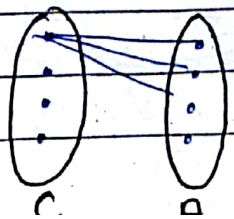
#



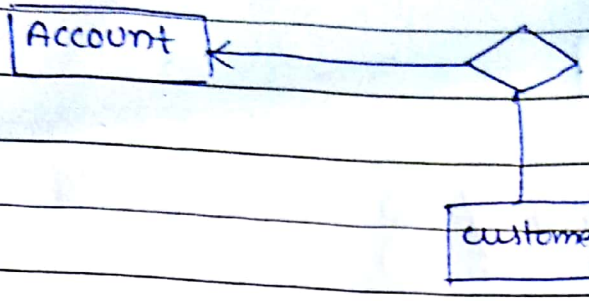


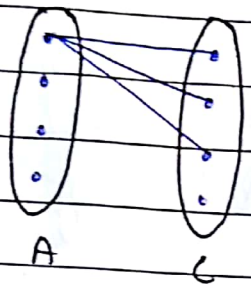
#

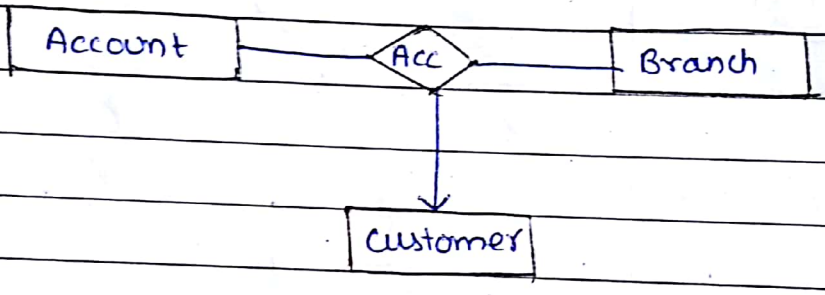


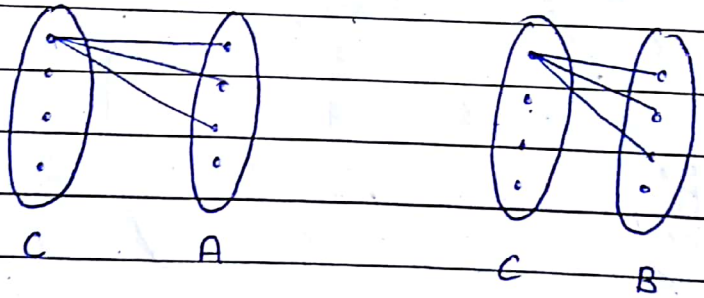


#



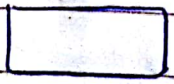




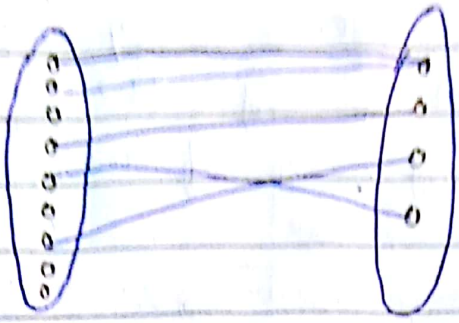


Existential dependency.

weak entity 

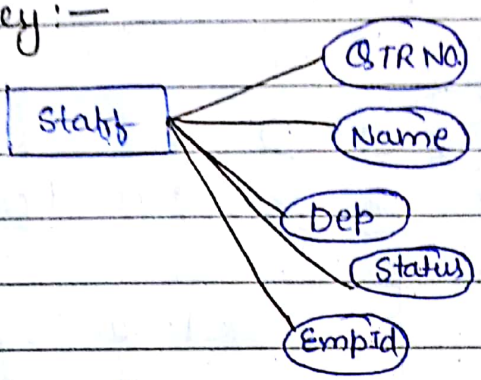
Strong entity 

3rd class



Key :—

Super key :—



{ QTR NO, Name, Dep }
{ QTR NO., Name }
{ Name, Dep, Status }
{ Name, Dep }
{ Empid }

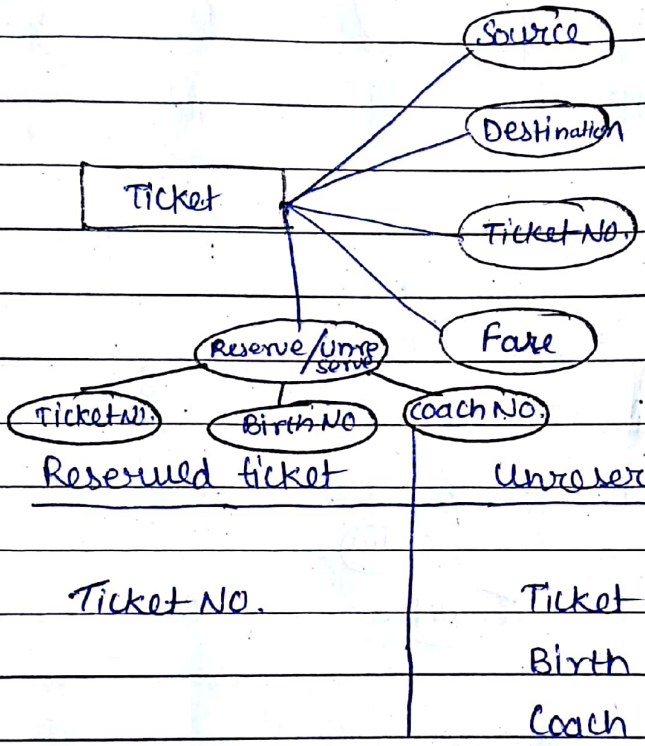
Candidate key :—

Primary key :—

Exestential Dependancy :—

weak entity set :—

weaks entity key do not have a Primary key. It is identifeed by the primary key of the strong entity set.

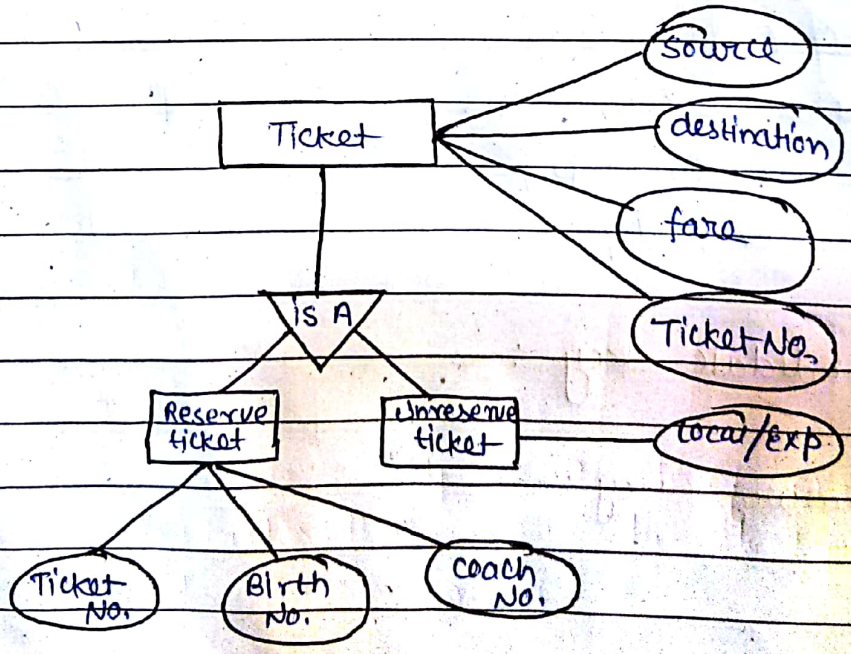


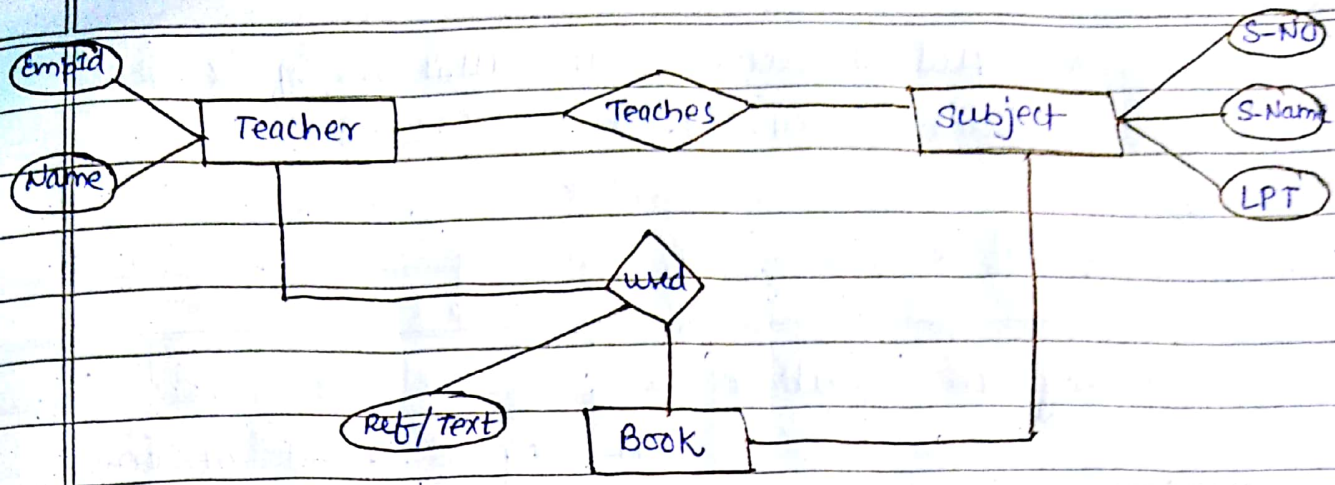
Reserved ticket

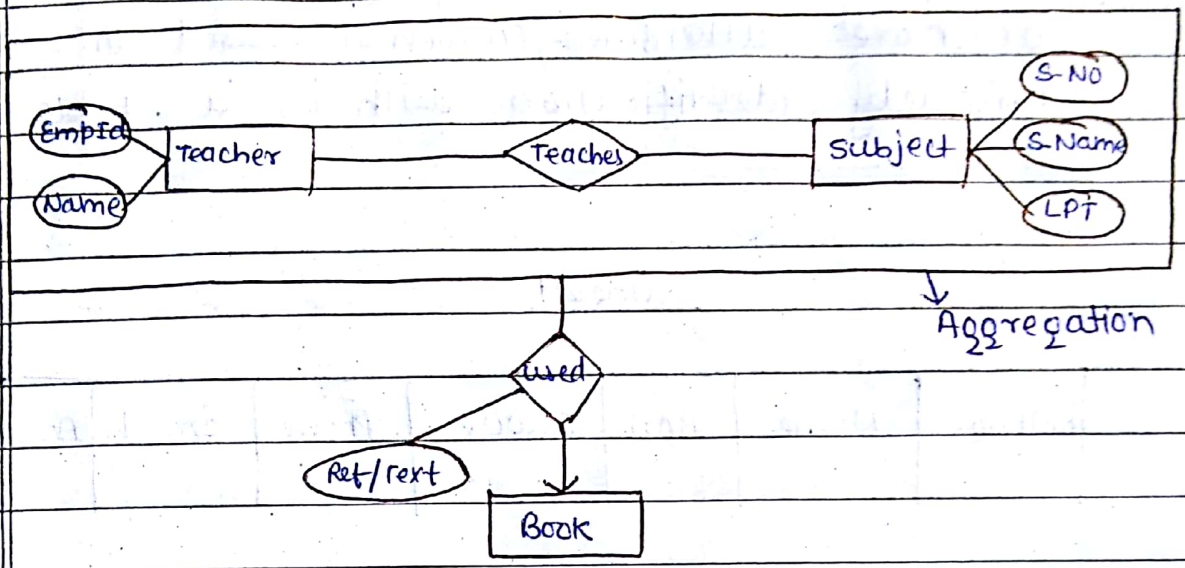
Ticket NO.

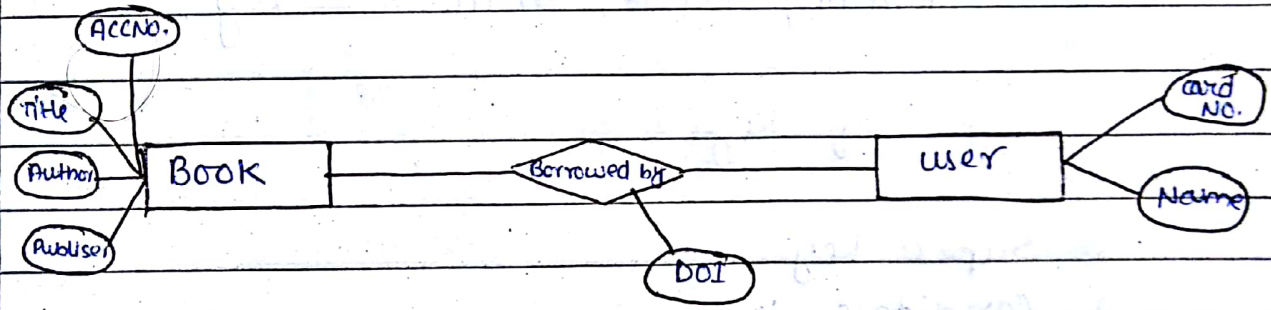
Unreserved ticket

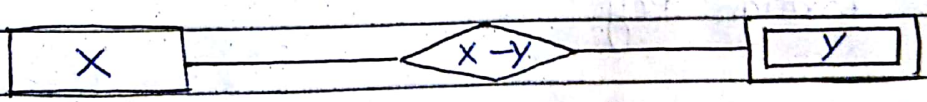
Ticket NO.
Birth NO.
Coach NO.











club or Not ?

Table of X-Y

Primary key of X	all the attributes of Y

Not club because All attributes of X Not associated with Y.

* self *

key in database :—

A key is a set of one or more attributes, which is used to uniquely identification with in a table.

e.g.

Student

Rollno.	Name	BOB	Course	Phone	Email	Address

Rollno, Phone, Email are key.

Types of Keys

1. Super key
2. Candidate key
3. Primary key
4. Foreign key.

Super key :—

Super key is a set of one or more attributes that uniquely identifies each records with in a table.

e.g.

Roll no

Rollno. + Name + course + Address

Rollno + Name.

Candidate key :—

Candidate key is a minimal super key, which contains no extra attributes.

it consists of maximum possible attributes, which is uniquely identifies.

e.g.

Roll no is candidate key

Roll no + Phone + Email is candidate key.

Primary key :—

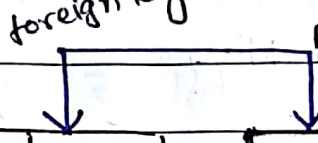
Primary key is an attributes, which is uniquely identifies each records with in a table.

e.g. — Roll no ✓

Foreign Foreign key :—

A foreign key are attributes in a table, whose value match as primary key in a another table.

- foreign key can contains duplicate value
- Also can contain NULL value.



Employee

E.No.	E.Name	Salary	D.NO
1	Pragati	50,000	D_1
2	Vipul	20,000	D_2
3	Anjelina	1,00000	
4	Abhishek	2,0000	D_1

Department

D.NO	D.Name	D.Loc
D_1	Sales	Delhi
D_2	Purchase	meruth

Date 22/01/18
Page 5

4th class (Monday (4-6))

Relational Algebra :— (procedural language)

Fundamental operation :—

1. Select
2. Project
3. Cartesion product
4. Union
5. set difference
6. Rename.

Some additional operation :—

1. Natural join
2. Theta join
3. Assignment
4. Intersection
5. Division / quotent

1. Select :—

"select those ~~constraint~~ that satisfy certain conditions" it partition our relation horizontally.

Symbol of the select is σ

e.g. book (Acc. No., title, Price, YR_PUB)

$\sigma_{Price > 1000}$ (book)

This is unary relationship.

e.g. $\sigma_{(price > 1000)}$ (book)

$\wedge (YR\text{-}PUB = 1992)$

2. Project :—

It portion our relation vertically. project operation can be represented by Π Symbol.

e.g.

$$\Pi_{ACC.NO,\ TITLE}\ (book)$$

Que- which price is > 1000 & only attribute TITLE and ACC.NO.

$$\Pi_{ACC.NO,\ TITLE}\left(\sigma_{Price > 1000}\ (book)\right)$$

3. Cartesion product operation :—

R =

A	B
a	b
c	d
e	f

S =

C	D
d	e
f	g

R x S =

A	B	C	D
a	b	d	e
a	b	f	g
c	d	d	e
c	d	f	g
e	f	d	e
e	f	f	g

R =

A	B
a	b
c	d
e	f

S =

B	C
d	e
f	g

R x S =

R.A	R.B	S.B	S.C
a	b	d	e
a	b	f	g
c	d	d	e
c	d	f	g
e	f	d	e
e	f	f	g

Que Find the names of all user who have the same address as Vijay.

user (card.no, bname, address)

USER x USER

$\Pi_{name}(\sigma_{add=Vijay}$ USER $)$

$\Pi_{bname}\left(\sigma_{(bname='VIJAY') \wedge (user.add = user.address)}(user \times user)\right)$

4. Rename operation :-

$\rho_{user1}(user)$

$\Pi_{user.b\_name}\left(\sigma_{(user1.bname='VIJAY') \wedge (user1.address = user.address)}\left(\rho_{user1}(user) \times user\right)\right)$

Book (ACC. NO, YR_PUB, Title)
borrow (ACC. NO, card NO, DOI)

Que Find the accession no. of all books which are available in the library.

Here we use set difference operation.

Set difference = Book − borrow.

$\Pi_{ACC.NO}(book) - \Pi_{ACC.NO}(borrow)$

Natural join operation :—

R =

A	B	C
a	b	c
d	e	f
g	h	i

S =

B	D
b	e
h	d

$r \bowtie S$ =

A	B	C	D
a	b	c	e
g	h	i	d

r =

A	B	C
a	b	c
d	e	f
g	h	i

S =

B	D	C
b	e	c
h	d	i
b	t	z

$r \bowtie S$ =

A	B	C	D
a	b	c	e
g	h	i	d

$r \bowtie S$ = $r \times S$

(If value of ~~attribut are~~ same)

Date 23/01/18
Page 20

Class 5th Tuesday (2-4)

Que Find out all the books which are either issued or have been supplied by supplier.

book (ACC·NO, YR_PUB, title)

sby (ACC., Sname, Price)

bby (ACC·NO, card NO, DOI)

$\Pi_{ACC\,NO}(bby) \cup \Pi_{ACC\,NO}(Sby)$

$R \cap S = R - (R - S)$

Exam-

R =

A	B
a	1
b	3

S =

A	B
a	1
b	2

$R \cap S$

A	B
a	1

Division / Quotient operation :—

R =

A	B
P	a

S =

B

$r/s = x - s$

A

$r =$

A	B
P	a
q	a
P	b
P	c
q	t
m	a
q	b

S

B
a
b

r/s

A
P
q

s में जो value होगी उसके combination r में होने चाहिए like a are present in s the its realy present Pa, Pb in table r

Que Sailor (sid, sname, rating, age)
Reserve (sid, bid, day)
Boat (bid, bname, color)

1. Find the names of sailors who have reserve boat no. 103.

$\pi_{sname}((\sigma_{bid=103}(\text{Reserves} \times \text{sailor})) \bowtie \text{sailor})$

Que Find the names of the sailors who have reserved a red boat.

$$\pi_{sname}\left(\sigma_{color='Red'}(boat) \bowtie (Reserve) \bowtie (sailor)\right)$$

Que Find the color of boats reserve by Lubber.

~~$\pi_{color}(boat) \bowtie \pi_{name='Lubber'}(reserve)$~~

$$\pi_{color}\left(\left(\sigma_{name='Lubber'}(sailor) \bowtie Reserve \bowtie boat\right)\right)$$

Que Find the names of sailors who have reserved at least one boat.

$$\pi_{sname}\left((Reserve) \bowtie (sailor)\right)$$

Que Find the names of sailors who have reserved a red or a green boat.

$$\pi_{sname}\left(\left(\sigma_{(color='red')}(boat) \bowtie Reserve \vee (color='green') \bowtie sailor\right)\right)$$

$\rho(tempbo, (\sigma_{color='Red'}(Boats)) \cup (\sigma_{color='green'}(Boat)))$

Tempbo = | bid | bname | color |

$\Pi_{sname}(tempbo \bowtie Reserve \bowtie sailor)$

Que Find the names of sailors who have reserved a red and green boat.

→

$\Pi_{sname}((\sigma_{color='Red'}(Boat)) \bowtie reserve \bowtie sailor) \cap$
$\Pi_{sname}((\sigma_{color='green'}(Boat)) \bowtie reserve \bowtie sailor)$

$\rho(tempRed, \Pi_{sid}(\sigma_{color='Red'}, Boat)) \bowtie Reserve$

$\rho(tempgreen, \Pi_{sid}(\sigma_{color='green'}, Boat)) \bowtie Reserve$

$\Pi_{sname}(tempRed \cap tempgreen) \bowtie sailor$

Que Find the names of sailors who have reserved atleast ~~one~~ two boat.

Date
Page 24

ρ (reser, $\Pi_{sname, sid, bid}$ (sailor $\bowtie$ Reserve))

reser =

sid	sname	bid
25	A.K	13
25	S.K	14

ρ (reserpair, (1-sid1, 2-sname1, 3-bid1, 4-sid2, 5-sname2, 6-bid2)

Reser X Reser

Π_{sname1} $\sigma_{(bid \neq bid2) \; (sid1 = sid2)}$ X (Reserpair)

Date 29/01/18
Page 25

Class 6th Monday (4-6)

Relational calculus:— (Non procedural language)

"Based on Notation of set theory."

- Explite Notation of set e.g. $S = \{1, 3, 5, \dots\}$
- Insplite Notation of set. e.g. $S = \{x \text{ is odd int} \mid x = 2y, y \in I\}$

Tuple relation formula:—

$$S = \{t \mid P(t)\}$$

t → tuple, P(t) → predicates

e.g. I want who books which are published in 1999

$$S = \{t \mid t \in BOOK \wedge t[YR\_PUB] = 1999\}$$

∴ t → all the attributes of Book

$$S = \{t \mid \exists u \in Book (t[title] = u[title] \wedge t[ACCNO] = u[ACCNO] \wedge u[YR\_PUB] = 1999)\}$$

Que Find the names of borrowers with there address who have issued a book on 14/8/95

user (card_NO, Name, add)
borrower (ACC_NO, card_NO, Date_of_issue)

$S = \{ t \mid \exists u \in user (t[name] = u[name] \wedge$
$t[add] = u[add] \wedge u \in borrower]$
$\exists b \in borrow (b[card\_NO] = u[card\_NO]$
$\wedge b[D.O.I] = 14|8|95) \}$

Que Find the names of borrowers who have issued a book supplied by Narosa or Allied

user(b_name, b_add, cardno)
supp(accno, s_name, price, DOS)
borrower(accno, card_no, DOI)

$S = \{ t \mid \exists u \in user (t[name] = u[name]) \wedge \exists b \in borrow$
$(b[card\ accno] = u[card\ accno] \wedge \exists c \in$
$supp\ borrower (c[accno] = b[acc] \wedge$
$(c[S\text{-}Name] = 'Narosa' \vee$
$c[S\text{-}Name] = 'Allied')) \}$

Que Find the names of borrowers who have issued a book supplied by Narosa but have not issued any book supplied by Allied.

$S = \{ t \mid \exists u \in user (t[b\_name] = u[b\_name])$
$\wedge \exists b \in borrower (b[cardno] = u[cardno]$
$\wedge \exists s \in supp (s[accno] = b[accno])$
$\wedge (s[s\_name] = 'Narosa' \wedge \neg$
$s[s\_name] = 'Allied') \}$

$\exists q \in supplier(q[acc\_no] = b[acc\_no] \wedge q[s\_name] = 'Allied'))\}$

Date- 30/01/18 Tuesday (2-4)

$\{t \mid \forall u \in Book(t[title] = u[title])\}$

means:-

t

title
compiler
DS
DBMS

u

ACCNO	title	YR
37	complier, DS, DBMS	
38	compiler, DS, DBMS	
39	compiler, DS, DBMS	

$\{t \mid \forall u \in Book(t[title] = u[title] \Rightarrow u[YR\_PUB] = '1991')\}$

Integrity Constraints :-

E-R model :-

1. Key constraints.
2. Cardinality constraints.
3. week entity and strong entity.

Schemes for checking integrity constraints :-

1. Domain constraints
2. Refrential constraints

3. Functional dependency.

Book (Acc_No, Title, YR_PUB)
Borrow (Acc_No, card_No, DOI)

✗ Acc No = 57 present in Borrow but not book
✓ Acc No = 57 present in Book but not Borrow

Foreign key :—

Let $r_1(R_1)$ & $r_2(R_2)$ be two relations. Suppose $\alpha \subseteq R_2$ is a foreign key referencing K (primary key of r_1 if $\Pi_\alpha(r_2) \subseteq \Pi_K(r_1)$.

∴ α → foreign key.

```
Create table Borrow( ACC_NO char(20),
                     card_NO char(20),
                     DOI Date
                     primary key (ACC NO),
foreign key ACC NO references Book);
```

$\Pi_\alpha(r_2) \subseteq \Pi_K(r_1)$

Insert : t_2 in r_2 $t_2(\alpha) \subseteq \Pi_K(r_1)$
t_1 in r_1 no check.

Delete :- t_2 from r_2 No check
t_1 from r_1 No corresponding Acc-No present in Borrower

Update :- delete followed by insert

t_1 in r_1
t_2 in r_2

Date - 06/02/18 Monday (4-6)

Employee

Fname	Minit	Lname	SSn	Add	DNo

Dept

Dname	DNumber

foreign key DNo References Dept.

Foreign key :-
A set of attributes FK in relational schema R_1 is a foreign key of R_1 that references relation R_2 if it satisfied :-

1. The attributes of FK must have the same domain as primary key of R_2. The attribute FK is said to refer to relation R_2.

2. A value of FK in a tuple t_1 of R_1 either occures as same value in tuple t_2 in R_2 or is null.

Ex-

Functional dependency :—

1. decide the attributes

e.g.

lib management :-

title, accNO, yr-Pub, Publisher, b-name, cardNo, Sname, supp-add.

It is a constrain b/w two attributes

$X : Y$

if $t_1[X] = t_2[X]$

then $t_1[Y] = t_2[Y]$

ACC NO → title ✓

ACC NO → Yr_Pub ✓

Title → Yr_Pub ✗

e.g. Database System → 1994

Concept

" → 2005

Title → ACC NO ✗

Compiler design . 59

" 49

Ex-

X	Y	Z	W
x_1	y_1	z_1	w_1
x_1	y_2	z_1	w_2
x_2	y_2	z_2	w_2
x_2	y_3	z_2	w_3
x_3	y_3	z_2	w_4

$x \to y$ x
$y \to z$ x
$x \to z$ ✓
$x \to w$ x
$y \to w$ x
$w \to y$ ✓

Dept → head
13 250231

head → phone
250231 231

Dep → Phone
13 231

closure :-
F of FD's + all the FD's inferred from F

denoted by closure is F^+

There are 6 inference rule first three are known as Armstrongs rule.

Inference Rule :-

1. Reflexive Rule :-
If $x \supseteq y \models x \to y$

e.g.
ACCNO → ACCNO

2. Augumentation rule :-

if $\{x \to y\} \models xz \to yz$

3. Transitive rule :-

if $\{x \to y, y \to z\} \models x \to z$

4. Decomposition rule :-

if $\{x \to yz\} \models x \to y, x \to z$

5. Union Rule :-

if $\{x \to y, x \to z\} \models x \to yz$

6. Pseudo transitive Rule :-

if $\{x \to y, wy \to z\}$

$\models wx \to z$