- **HIERARCHICAL MODEL (COVERSION,I,U,D,R):-**

Relational Model to Hierarchical Model Supplier-Part Database

The Supplier records

Sno 1	Name 2	Status 3	City 4
S1	Suneet	20	Qadian
S2	Ankit	10	Amritsar
S3	Amit	10	Amritsar

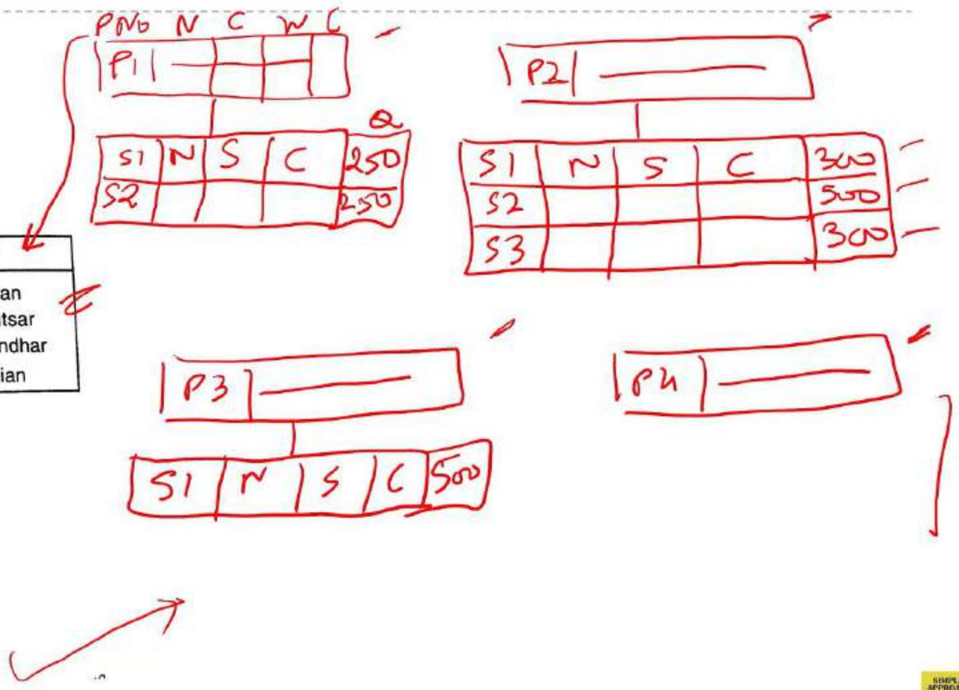
The Part records

Pno	Name	Color	Weight	City
P1	Nut	Red	12	Qadian
P2	Bolt	Green	17	Amritsar
P3	Screw	Blue	17	Jalandhar
P4	Screw	Red	14	Qadian

The Shipment records

R

Sno	Pno	Qty
S1	P1	250
S1	P2	300
S1	P3	500
S2	P1	250
S2	P2	500
S3	P2	300



Hierarchical Model: Supplier-Part Database

The Supplier records

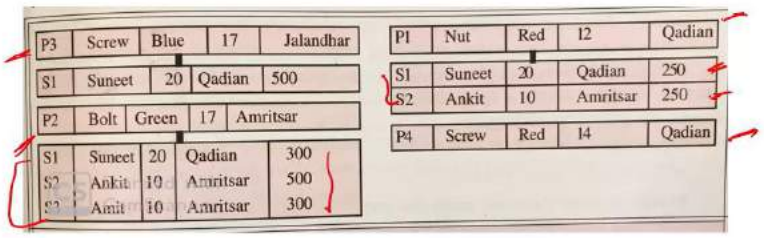
Sno	Name	Status	City
S1	Suneet	20	Qadian
S2	Ankit	10	Amritsar
S3	Amit	10	Amritsar

The Part records

Pno	Name	Color	Weight	City
P1	Nut	Red	12	Qadian
P2	Bolt	Green	17	Amritsar
P3	Screw	Blue	17	Jalandhar
P4	Screw	Red	14	Qadian

The Shipment records

Sno	Pno	Qty
S1	P1	250
S1	P2	300
S1	P3	500
S2	P1	250
S2	P2	500
S3	P2	300

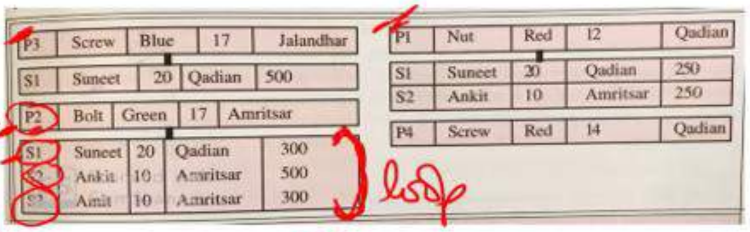


Retrieve Operation

- **Query1:** Find supplier numbers who supply part P2.

Algorithm

```
get [next] part where PNO=P2;
do until no more shipments under this part;
    get next supplier under this part;
    print SNO;
end;
```



P3	Screw	Blue	17	Jalandhar
S1	Suneet	20	Qadian	500
P2	Bolt	Green	17	Amritsar
S1	Suneet	20	Qadian	300
S2	Ankit	10	Amritsar	500
S3	Amit	10	Amritsar	300
P1	Nut	Red	12	Qadian
S1	Suneet	20	Qadian	250
S2	Ankit	10	Amritsar	250
P4	Screw	Red	14	Qadian

loop

✓2

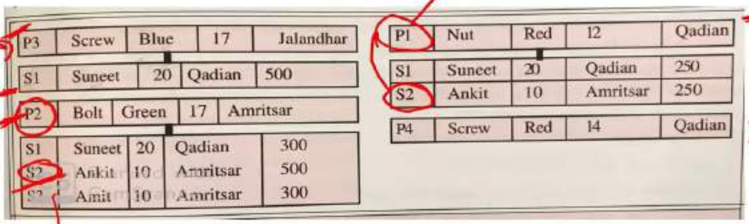
Retrieve Operation

- **Query1:** Find part numbers for parts supplied by supplier S2.

Algorithm

```
do until no more parts;
    get next part;
    get [next] supplier under this part where SNO=S2;
    if found then print PNO;
end;
```

P1
P2

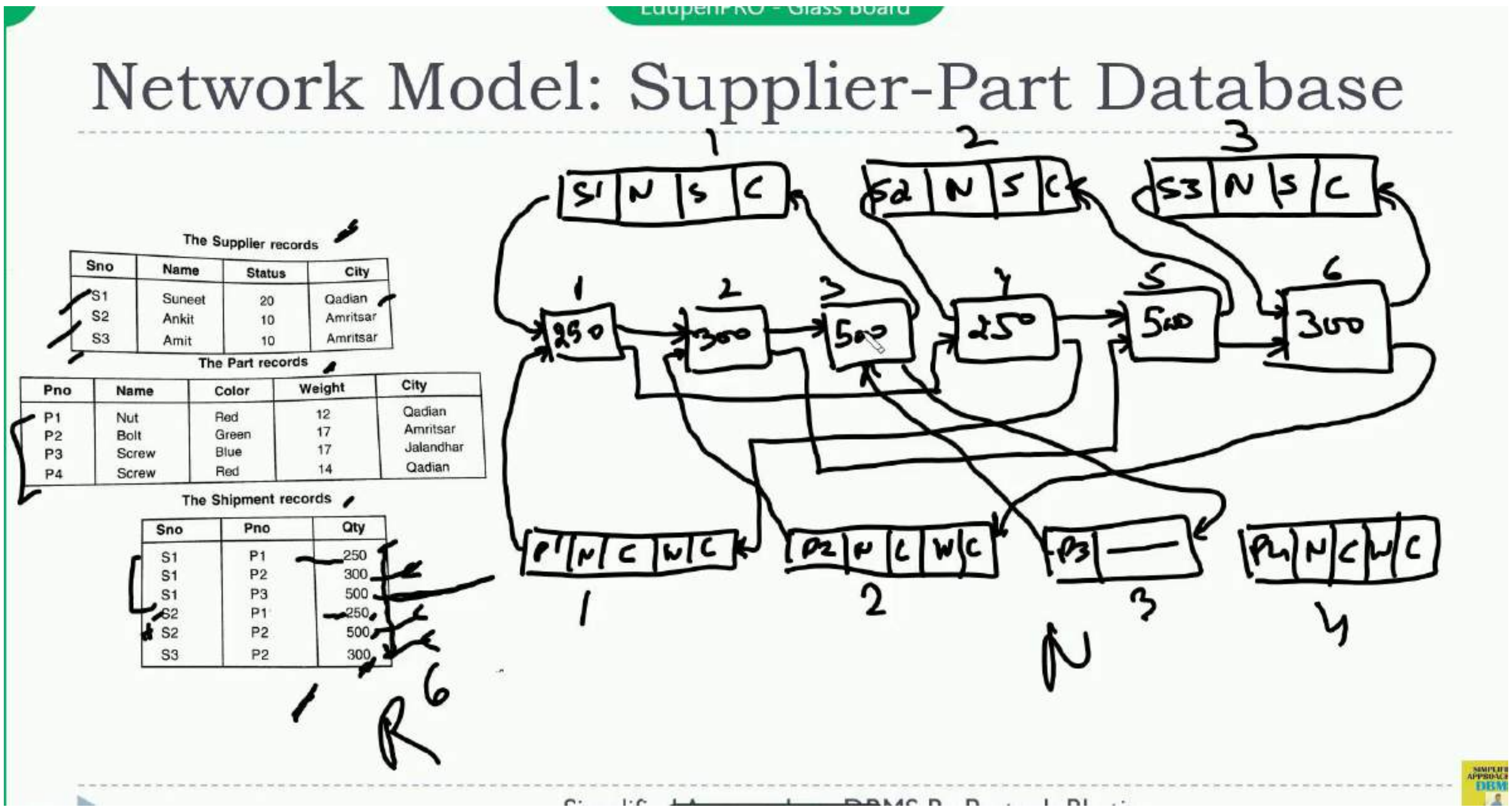


P3	Screw	Blue	17	Jalandhar
S1	Suneet	20	Qadian	500
P2	Bolt	Green	17	Amritsar
S1	Suneet	20	Qadian	300
S2	Ankit	10	Amritsar	500
S3	Amit	10	Amritsar	300
P1	Nut	Red	12	Qadian
S1	Suneet	20	Qadian	250
S2	Ankit	10	Amritsar	250
P4	Screw	Red	14	Qadian

Anomalies of Hierarchical Model

- Insert ✓
  - Child data cannot inserted without parent
- Update
  - Child record need multiple update operations which is equal to number of parents it has.
- Delete
  - Deletion of parent results into deletion of all corresponding child records which is worst for a child who has only one parent.
- Retrieval
  - Retrieval operations are asymmetric.

- **NETWORK MODEL (COVERSION,I,U,D,R):-**



Network Model: Supplier-Part Database

The Supplier records

Sno	Name	Status	City
S1	Suneet	20	Qadian
S2	Ankit	10	Amritsar
S3	Amit	10	Amritsar

The Part records

Pno	Name	Color	Weight	City
P1	Nut	Red	12	Qadian
P2	Bolt	Green	17	Amritsar
P3	Screw	Blue	17	Jalandhar
P4	Screw	Red	14	Qadian

The Shipment records

Sno	Pno	Qty
S1	P1	250
S1	P2	300
S1	P3	500
S2	P1	250
S2	P2	500
S3	P2	300

Network Model: Supplier-Part Database

The Supplier records

Sno	Name	Status	City
S1	Suneet	20	Qadian
S2	Ankit	10	Amritsar
S3	Amit	10	Amritsar

The Part records

Pno	Name	Color	Weight	City
P1	Nut	Red	12	Qadian
P2	Bolt	Green	17	Amritsar
P3	Screw	Blue	17	Jalandhar
P4	Screw	Red	14	Qadian

The Shipment records

Sno	Pno	Qty
S1	P1	250
S1	P2	300
S1	P3	500
S2	P1	250
S2	P2	500
S3	P2	300

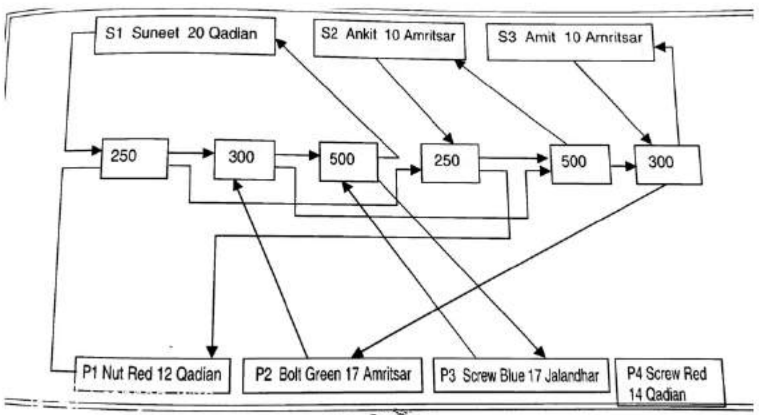


Figure 3.6.

Insert

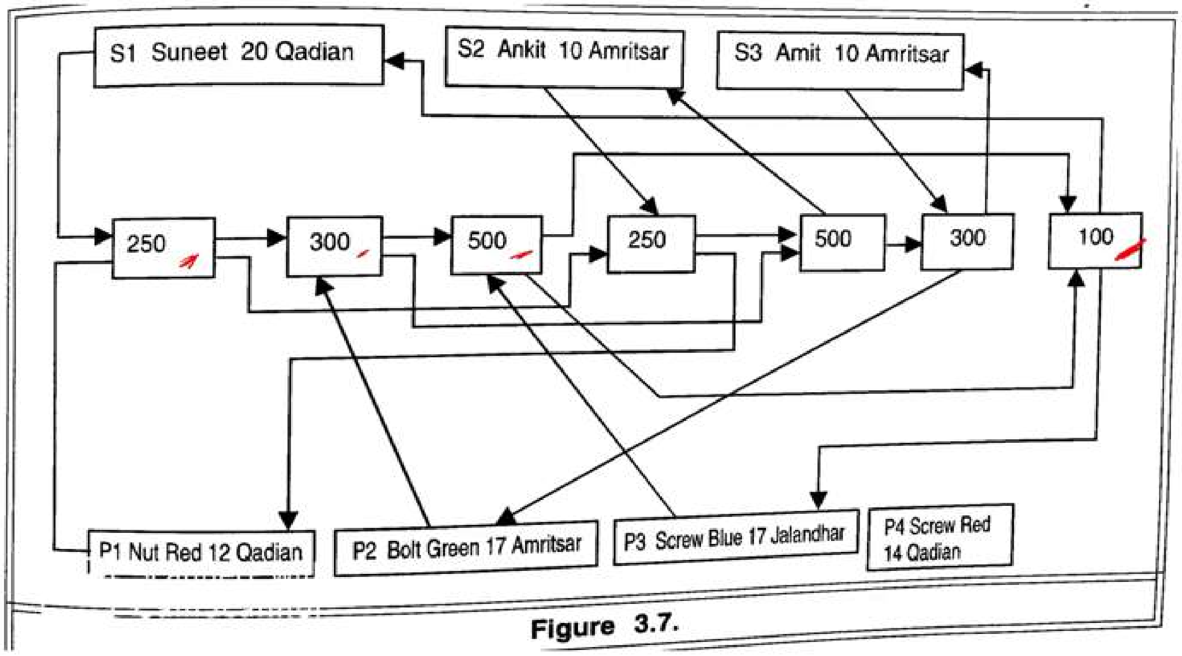


Figure 3.7.

Delete Operation

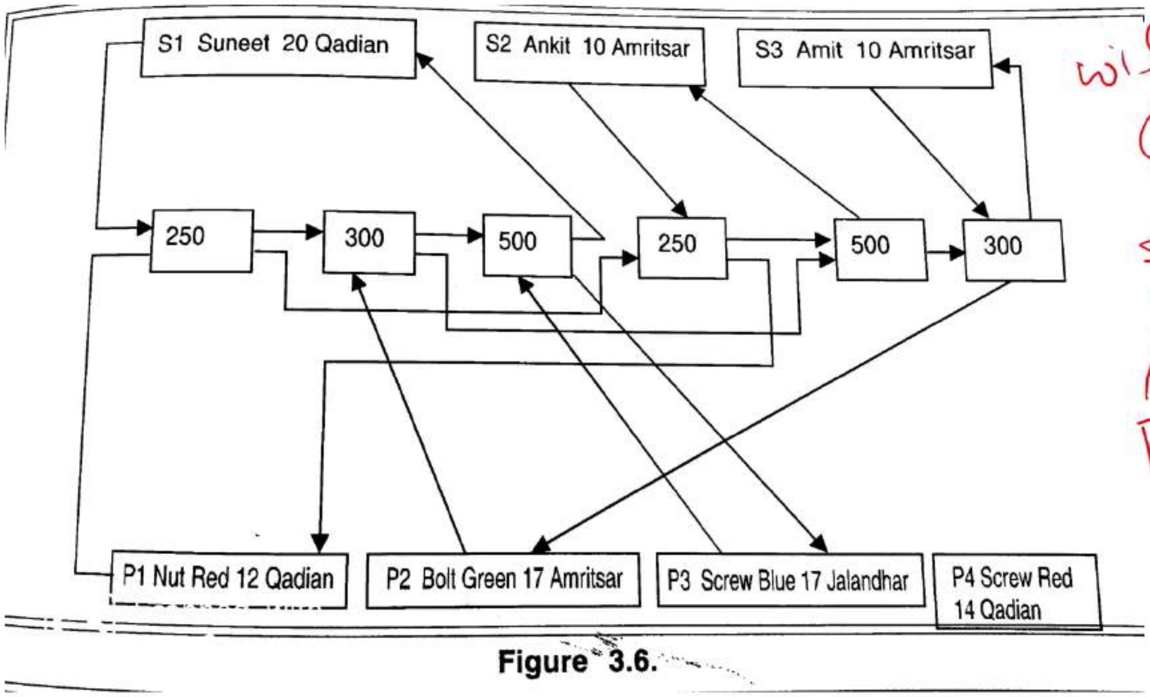


Figure 3.6.

will not be affected

5

shipments

P1

Retrieve Operation

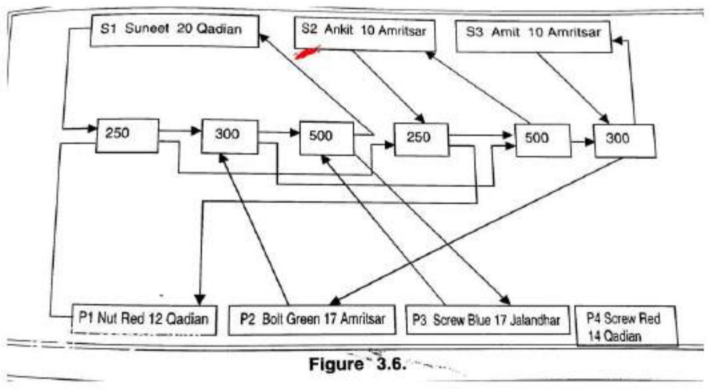


Figure 3.6.

Find part number for parts supplied by supplier S2.

Algorithm

```
get [next] supplier where SNO=S2;
do until no more connectors under this supplier;
        get next connector under this supplier;
        get part over this connector;
        print PNO;
end;
```

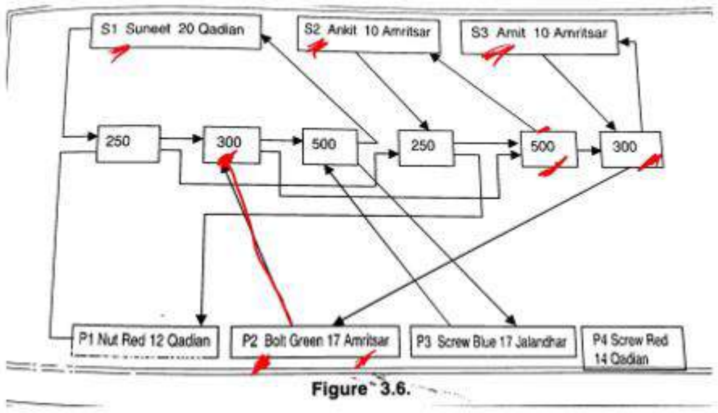


Figure 3.6.

Find supplier number for suppliers who supply part P2.

Algorithm

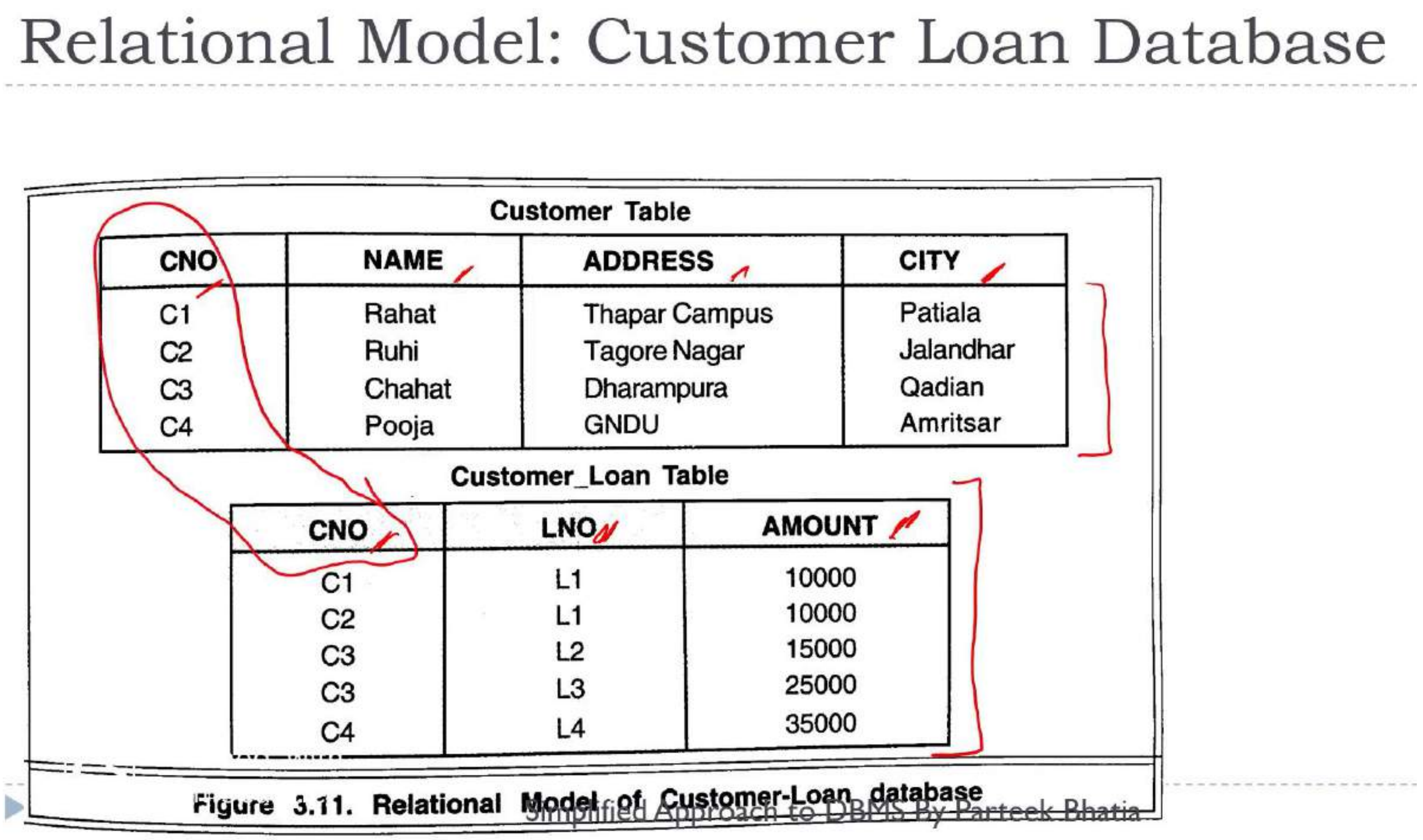
```
get [next] part where PNO=P2;
do until no more connectors under this part;
        get next connector under this part;
        get supplier over this connector;
        print SNO;
end;
```

$\boxed{S1, S2, S3}$

Operations over Network Model

- Insert
  - There is no anomaly.
- Update
  - There is no anomaly.
- Delete
  - There is no anomaly.
- Retrieve
  - There is no anomaly, retrieval operations are symmetric.
- Limitation of Network Model
  - The only limitation of network model is its complexity.

- **RDBMS[RELATIONAL] MODEL (COVERSION,I,U,D,R):-**



Relational Model: Customer Loan Database

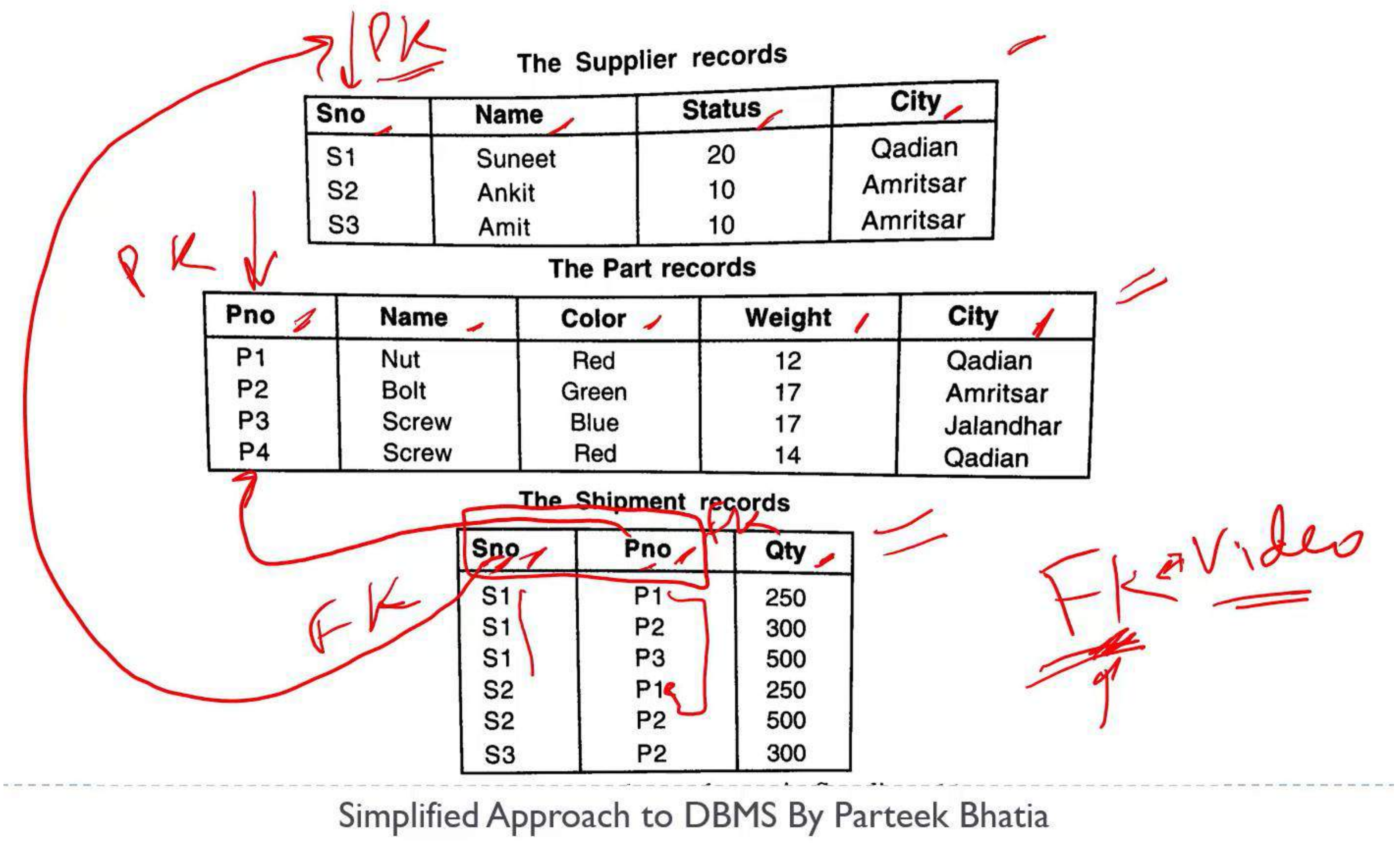
Customer Table

CNO	NAME	ADDRESS	CITY
C1	Rahat	Thapar Campus	Patiala
C2	Ruhi	Tagore Nagar	Jalandhar
C3	Chahat	Dharampura	Qadian
C4	Pooja	GNDU	Amritsar

Customer_Loan Table

CNO	LNO	AMOUNT
C1	L1	10000
C2	L1	10000
C3	L2	15000
C3	L3	25000
C4	L4	35000

Figure 3.11. Relational Model of Customer-Loan database



PK

The Supplier records

Sno	Name	Status	City
S1	Suneet	20	Qadian
S2	Ankit	10	Amritsar
S3	Amit	10	Amritsar

PK

The Part records

Pno	Name	Color	Weight	City
P1	Nut	Red	12	Qadian
P2	Bolt	Green	17	Amritsar
P3	Screw	Blue	17	Jalandhar
P4	Screw	Red	14	Qadian

The Shipment records

PK

Sno	Pno	Qty
S1	P1	250
S1	P2	300
S1	P3	500
S2	P1	250
S2	P2	500
S3	P2	300

FK

FK Video

https://www.w3schools.com/sql/sql_foreignkey.asp

PK

The Supplier records

Sno	Name	Status	City
S1	Suneet	20	Qadian
S2	Ankit	10	Amritsar
S3	Amit	10	Amritsar

PK

The Part records

Pno	Name	Color	Weight	City
P1	Nut	Red	12	Qadian
P2	Bolt	Green	17	Amritsar
P3	Screw	Blue	17	Jalandhar
P4	Screw	Red	14	Qadian

The Shipment records

PK

Sno	Pno	Qty
S1	P1	250
S1	P2	300
S1	P3	500
S2	P1	250
S2	P2	500
S3	P2	300

FK

FK Video

Insert Operation

The Supplier records

Sno	Name	Status	City
S1	Suneet	20	Qadian
S2	Ankit	10	Amritsar
S3	Amit	10	Amritsar

S

The Part records

Pno	Name	Color	Weight	City
P1	Nut	Red	12	Qadian
P2	Bolt	Green	17	Amritsar
P3	Screw	Blue	17	Jalandhar
P4	Screw	Red	14	Qadian

P

The Shipment records

Sno	Pno	Qty
S1	P1	250
S1	P2	300
S1	P3	500
S2	P1	250
S2	P2	500
S3	P2	300
S2	P3	100

$S2 \rightarrow P3 \rightarrow 100$

Update Operation

PK

The Supplier records

Sno	Name	Status	City
S1	Suneet	20	Qadian
S2	Ankit	10	Amritsar
S3	Amit	10	Amritsar

S1 → Name

PK

The Part records

Pno	Name	Color	Weight	City
P1	Nut	Red	12	Qadian
P2	Bolt	Green	17	Amritsar
P3	Screw	Blue	17	Jalandhar
P4	Screw	Red	14	Qadian

P1 → Name

The Shipment records

Sno	Pno	Qty
S1	P1	~~250~~ 600
S1	P2	300
S1	P3	500
S2	P1	250
S2	P2	500
S3	P2	300

S1 P1 ~~255~~ 600

Delete Operation

The Supplier records

Sno	Name	Status	City
~~S1~~	~~Suneet~~	~~20~~	~~Qadian~~
S2	Ankit	10	Amritsar
S3	Amit	10	Amritsar

S1 X

SX

The Part records

Pno	Name	Color	Weight	City
P1 ✓	Nut	Red	12	Qadian
P2 ✓	Bolt	Green	17	Amritsar
P3 ✓	Screw	Blue	17	Jalandhar
P4	Screw	Red	14	Qadian

P ✓

The Shipment records

Sno	Pno	Qty
S1	P1	250
S1	P2	300
S1	P3	500
S2	P1	250
S2	P2	500
S3	P2	300

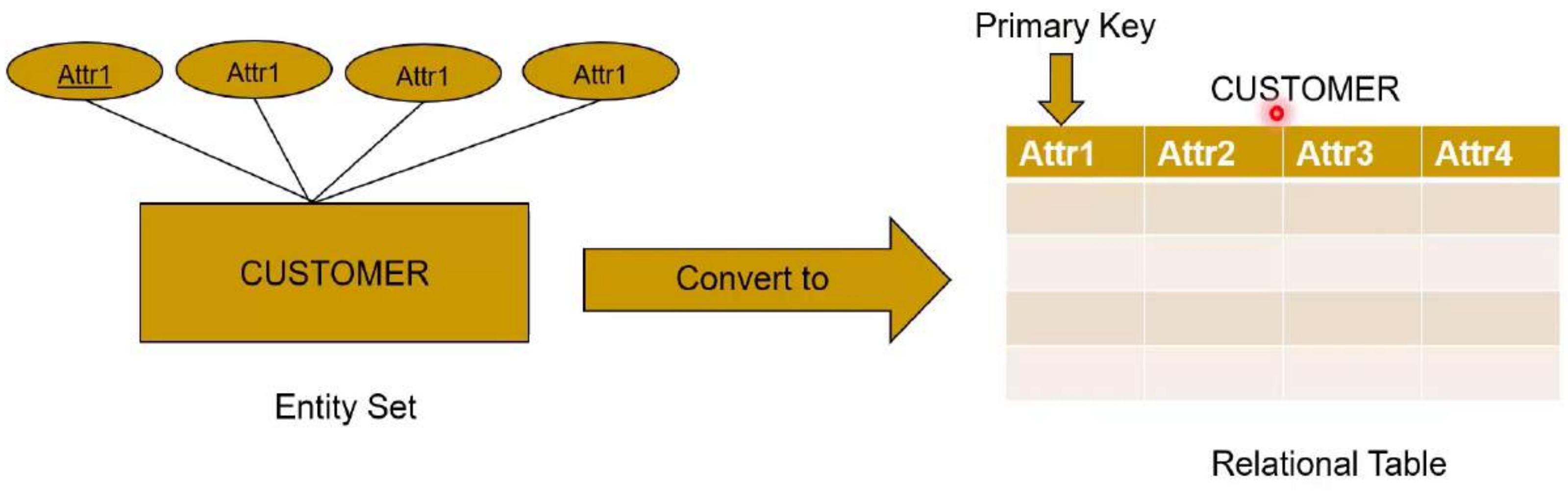
SP X

==

ER MODEL:-

https://www.javatpoint.com/dbms-er-model-concept

Conversion of Entity Set to Relational Table

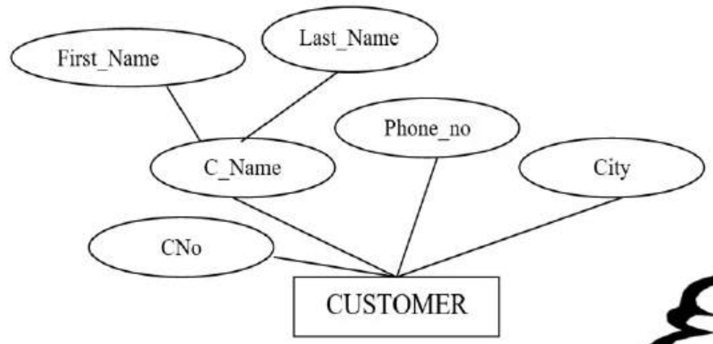


Handling of Attributes

- For every simple attribute create a column of the table.
- The key attribute will become primary key of the table.
- However, composite attribute, multi-value attribute and derived attribute need special treatment.

How to handle Composite Value Attribute?

Handling of Composite Attribute



CNo	First_Name	Last_Name	Phone_No	City
100	Rahat	Bhatia	2444566	Patiala
101	Ruhani	Sharma	4547235	Jalandhar
102	Raj	Singh	3445432	Amritsar

```
SELECT CNO,
FIRST_NAME|| ' '||LAST_NAME AS FULL_NAME
FROM CUSTOMER;
```

How to handle Multiple Value Attribute?

Possible Solution: Create a column for each possible value of attribute for a record

Cno	C_Name	City	Mobile_No1	Mobile_No2	Mobile_No3
100	Rahat Bhatia	Patiala	1876115046	8739593711	8739593715
101	Ruhani Sharma	Jalandhar	3475784928		
102	Rishan	New Delhi	7457483929	8734648483	

```
SELECT C_NAME FROM CUSTOMER WHERE
MOBILE_NO1=8739593711
OR
MOBILE_NO2=8739593711
OR
MOBILE_NO3=8739593711;
```

Second Solution

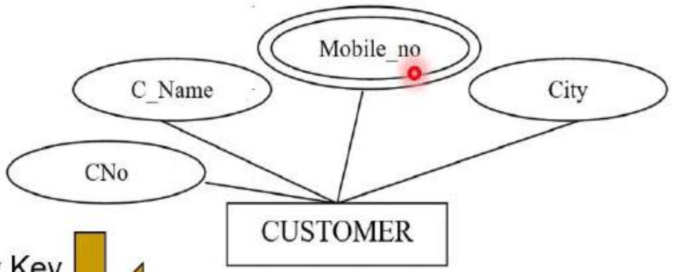
Store the multiple values of the column by comma separation.

Cno	C_Name	City	Mobile_Nos
100	Rahat Bhatia	Patiala	1876115046, 8739593711, 8739593715
101	Ruhani Sharma	Jalandhar	3475784928
102	Rishan	New Delhi	7457483929, 8734648483

Integrity of data is lost.
Still Retrieve and Updation and Deletion operations are tricky
Will be very tough and tricky.
Simplicity of RDBMS is lost…

Handling of Multi-value Attribute

- Solution: Two Tables



Primary Key

Foreign Key

Cno	C_Name	City
100	Rahat Bhatia	Patiala
101	Ruhani Sharma	Jalandhar
102	Rishan	New Delhi

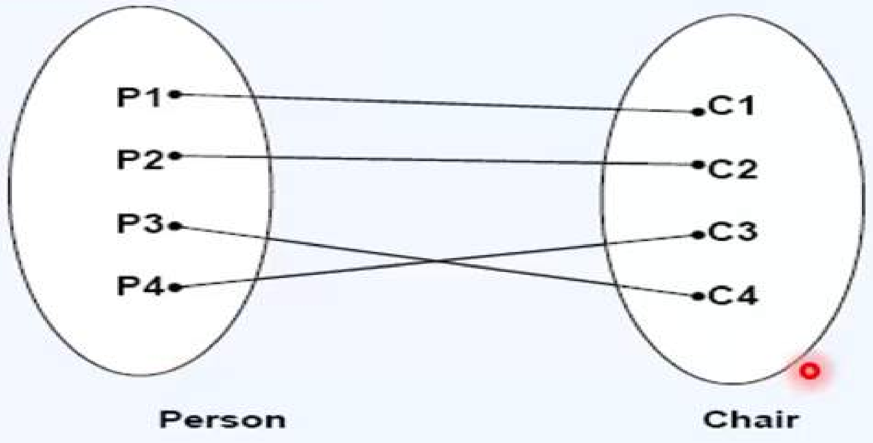
Table: Customer

Primary Key

Cno	Mobile_No
100	1876115046
100	8739593711
100	8739593715
101	3475784928
102	7457483929
102	8734648483

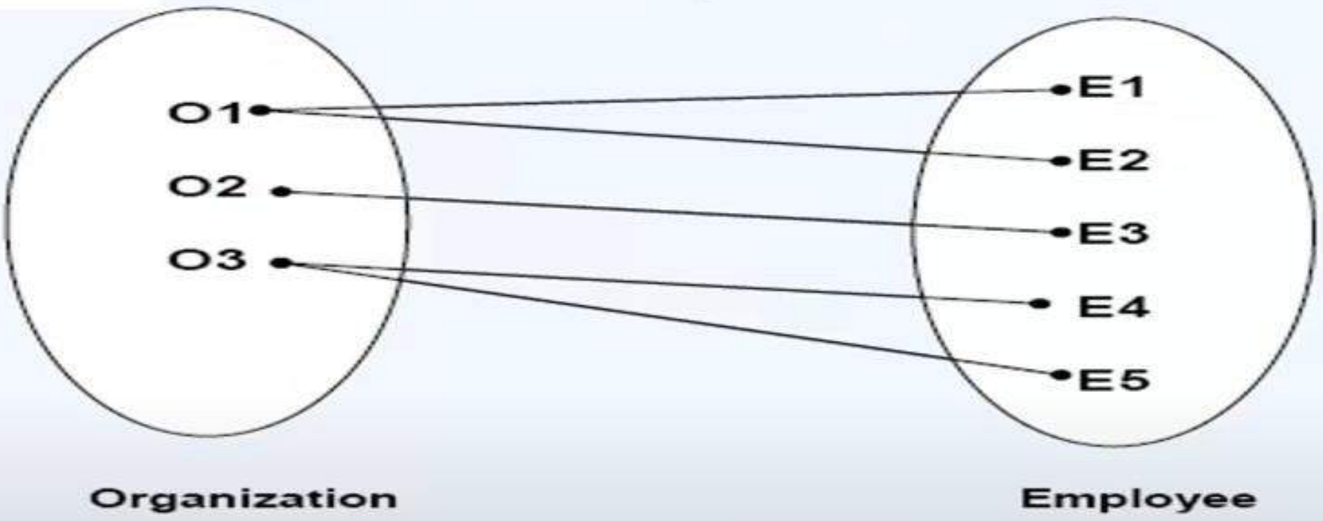
Table: Cust_Mobile

One - To - One



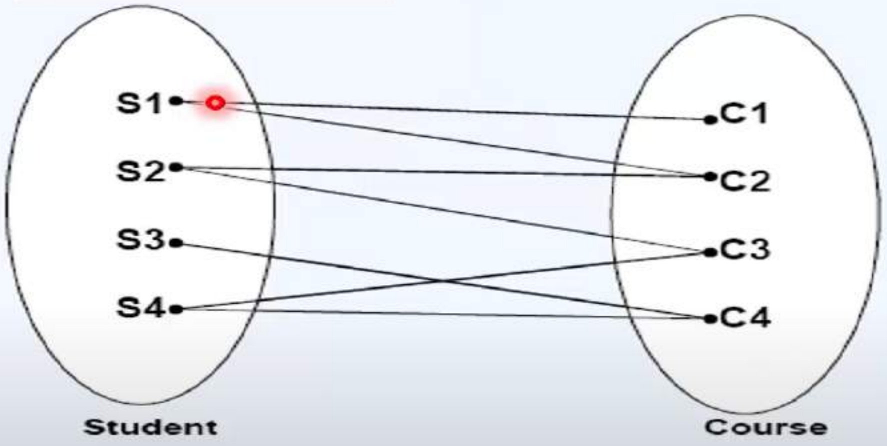
One instance of entity type Person
is related
to one instance of the entity type Chair.

One -to- Many



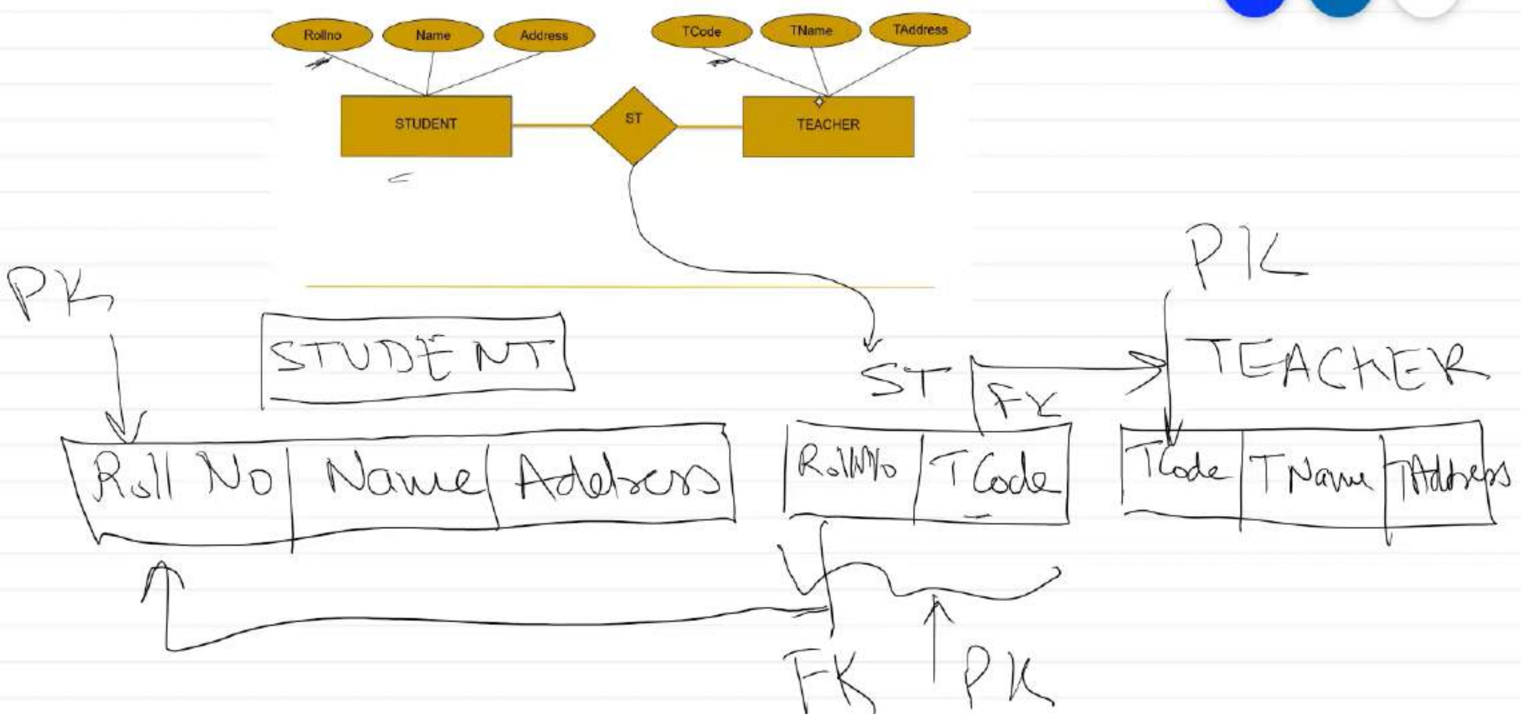
One instance of entity type Organization
is related
to multiple instances of entity type Employee

Many-to-Many

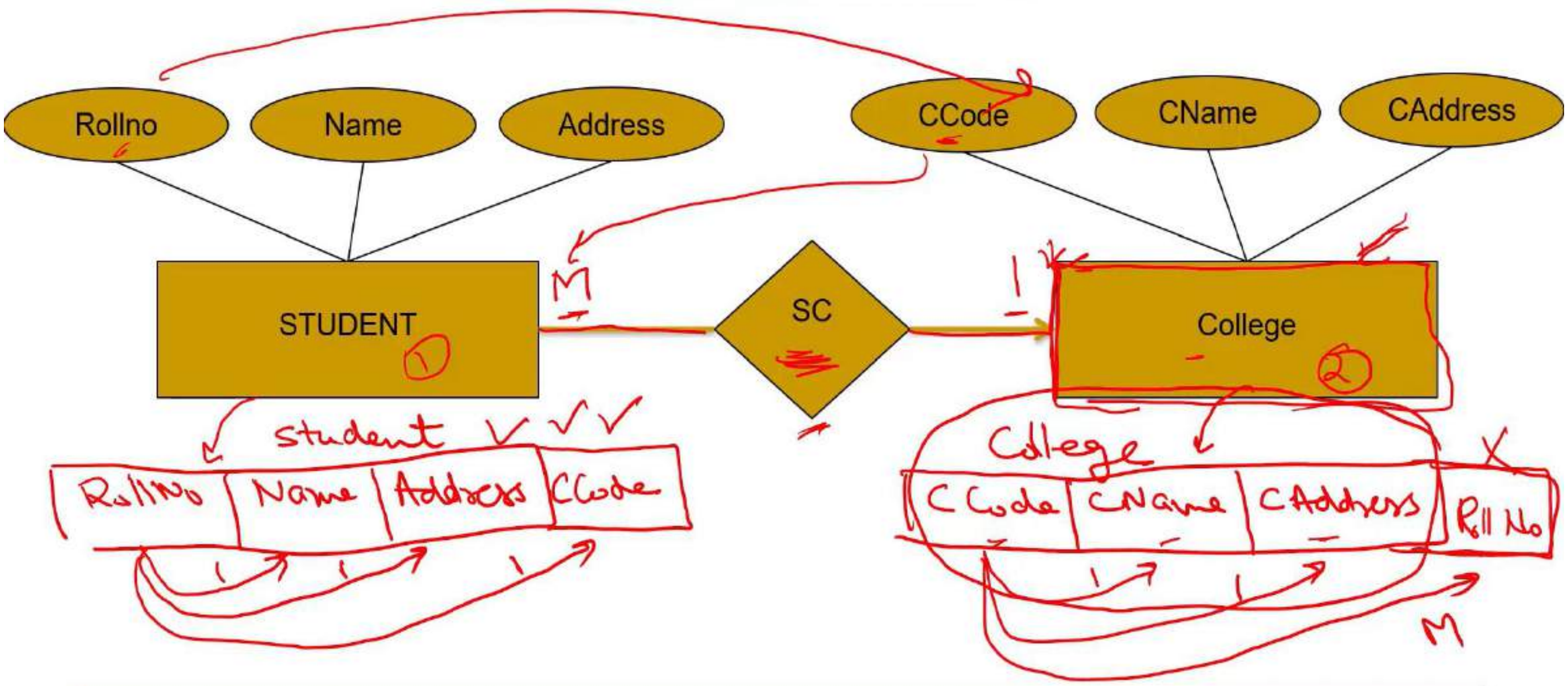


Multiple instances of one Entity are related to multiple instances of another Entity.

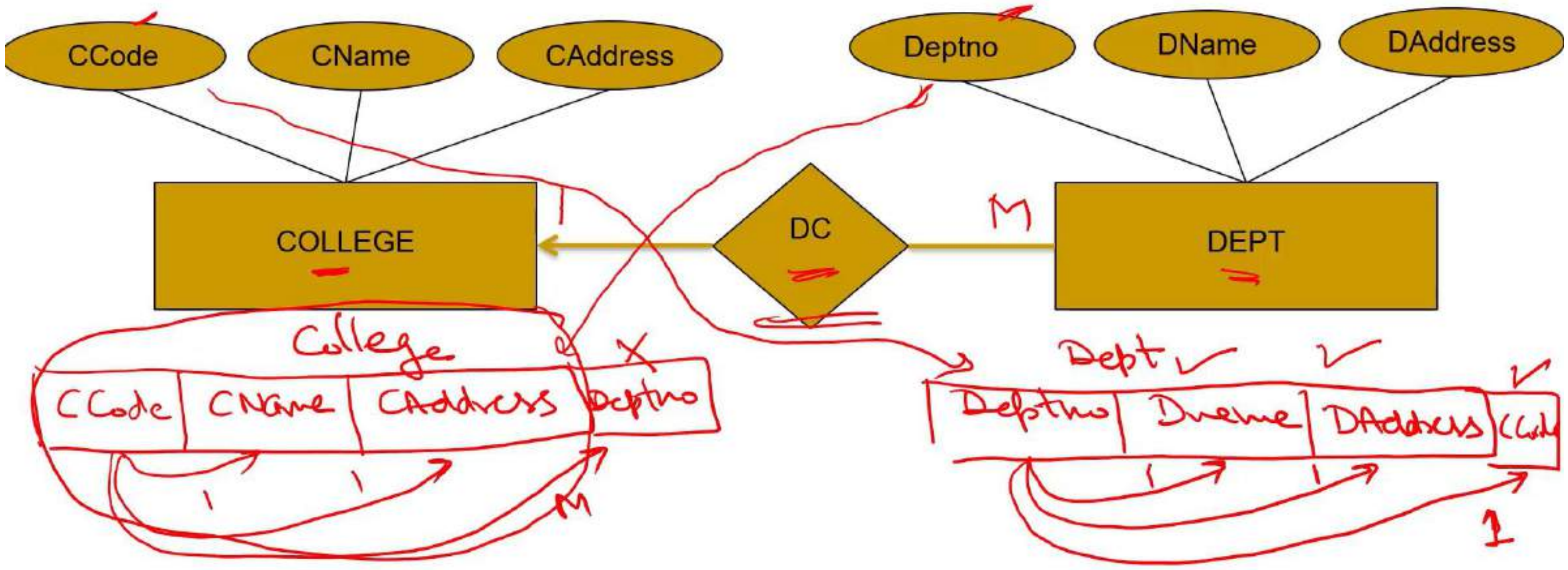
Handling of Many to Many Type of Relationship



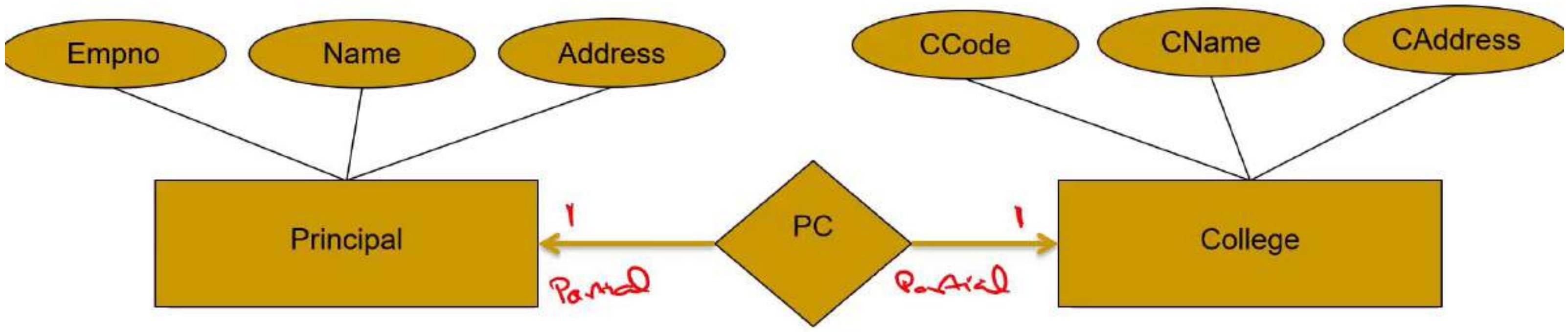
Handling of Many to One Type of Relationship



Handling of One to Many Type of Relationship

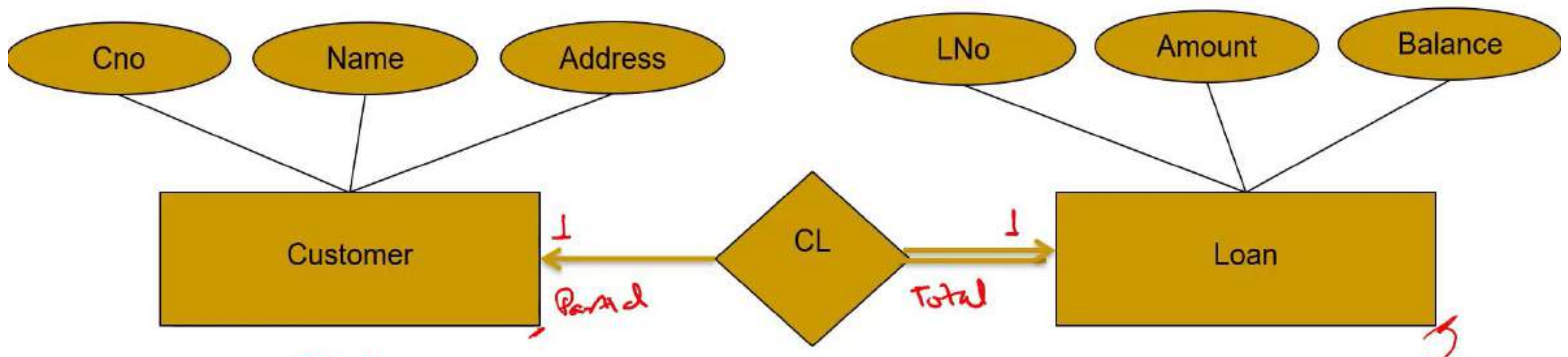


Handling of One to One Type of Relationship



Rule:
For One to One type of relationship, there is no need to create a separate table for the relationship.
Copy the primary key of any of one entity set towards another entity set.

Handling of One to One Type of Relationship



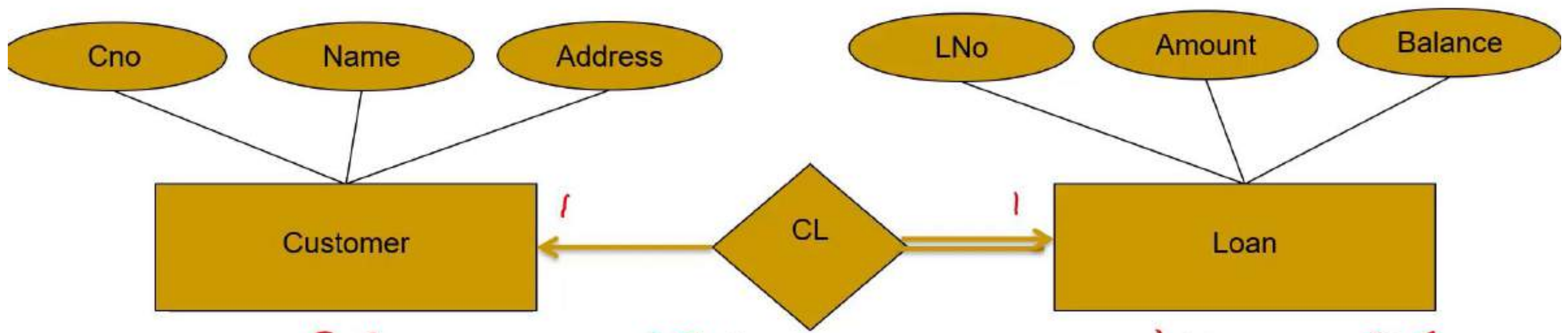
Customer

CNo	Name	Address

Loan

LNo	Amount	Balance	CNo

Handling of One to One Type of Relationship



Customer 90%

CNo	Name	Address	LNo
C1	A	ASR	Null
C2	B	JAL	L1
C3	C	PTA	Null
C4	D	PTA	Null

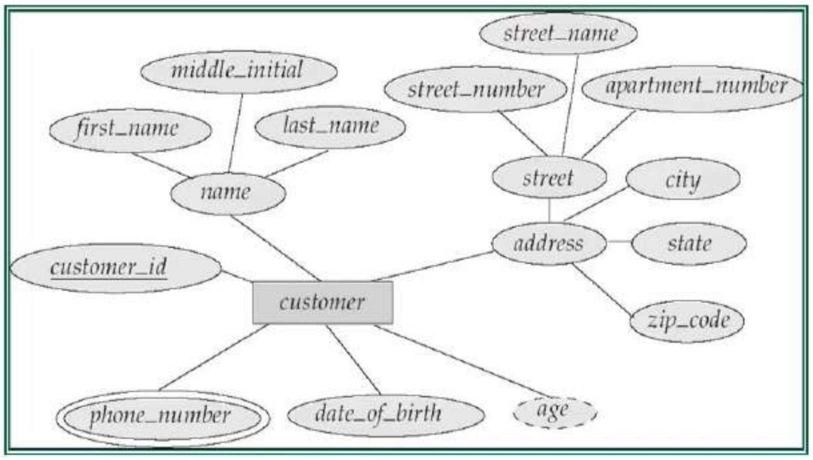
10%

Loan

LNo	Amount	Balance	CNo
L1	—	—	C1
L2	—	—	C2
L3	—	—	C3

Rule:
If you have any of entity set with total participation, then always copy primary key of one entity set towards an entity set having total participation.

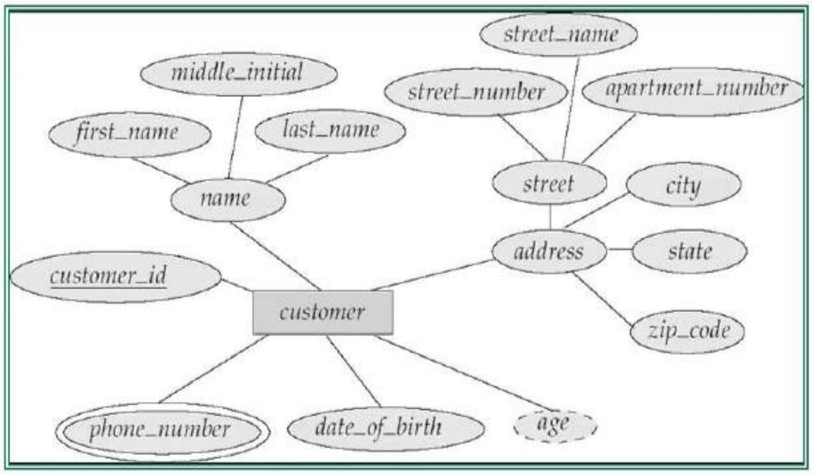
Example



- No need to create a column for age.
- Its value will be derived from date_of_birth column
- The query for this is:

SELECT
(SYSDATE- DATE_OF_BIRTH)/365 AGE
FROM CUSTOMER;

Conversion of Entity Set to Tables



- Table: CUSTOMER
- Columns:

customer_id, first_name,
Middle_initial, last_name, stree_number,
street_name, apartment_number, city,
state, zip_code, date_of_birth
There will be 11 columns

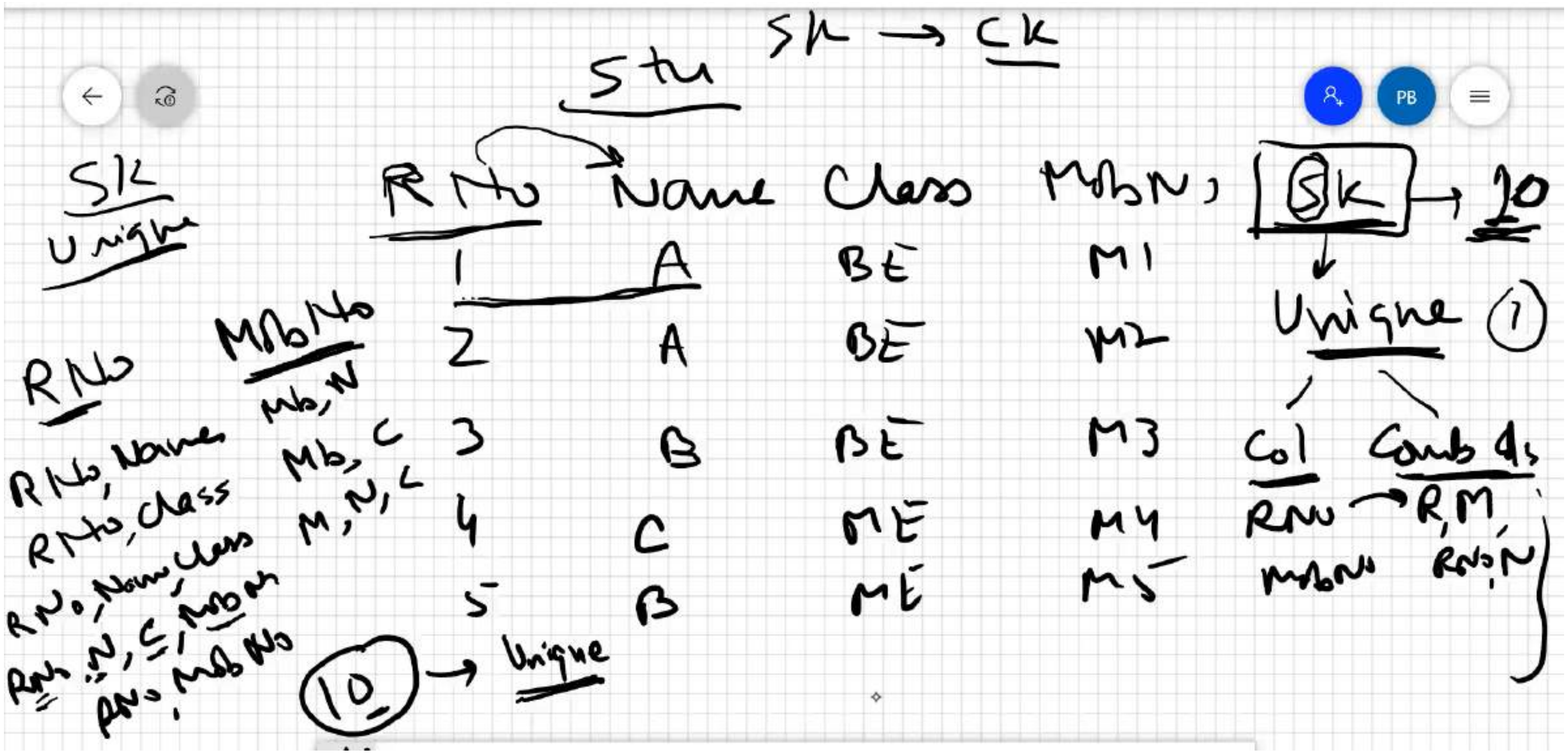
- Table: Customer_phone
- Columns:

Customer_id, phone_number

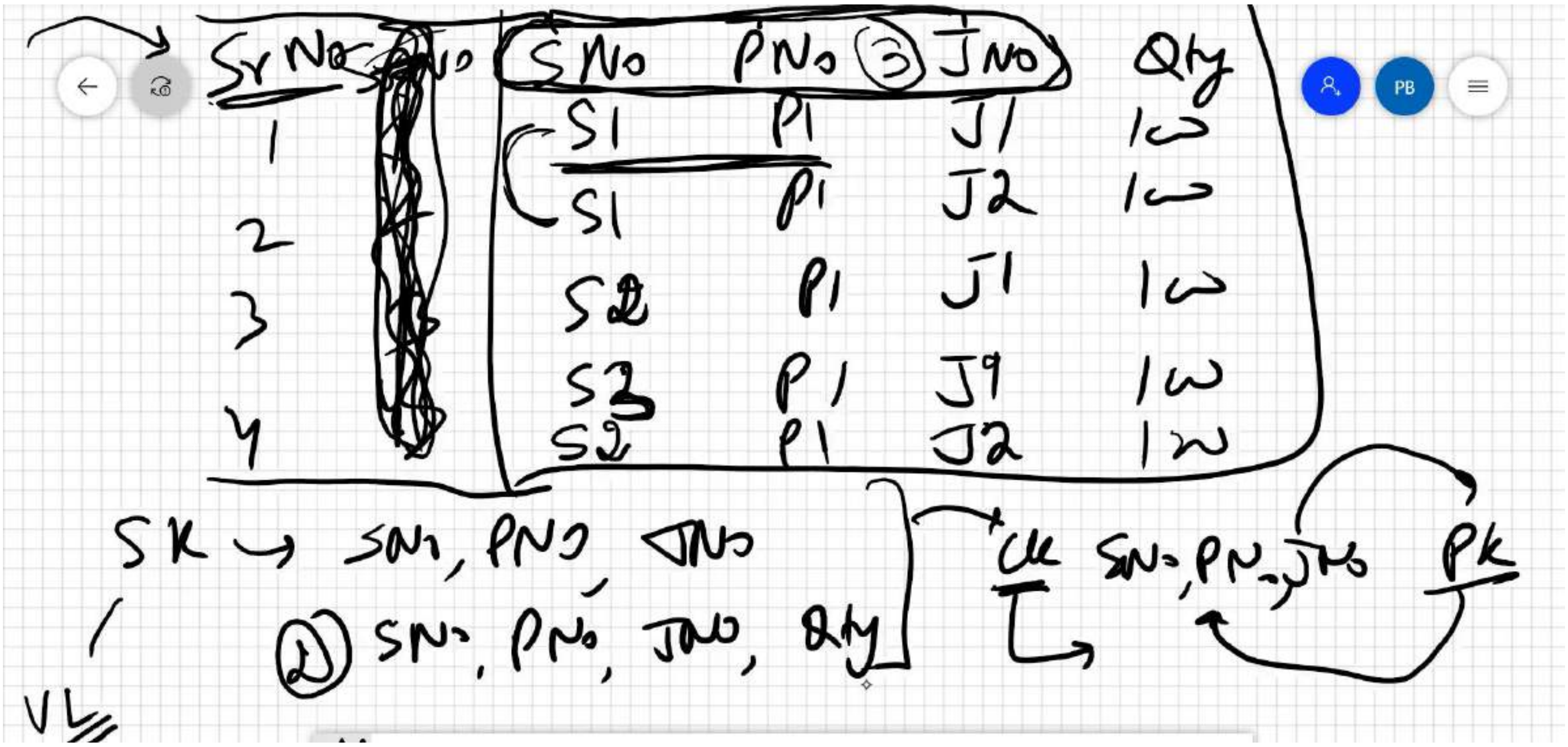
==
==
=================================

https://www.javatpoint.com/dbms-keyshttps://www.gatevidyalay.com/keys-in-dbms/https://beginnersbook.com/2015/04/super-key-in-dbms/

SUPER KEY COMBINATIONS:-



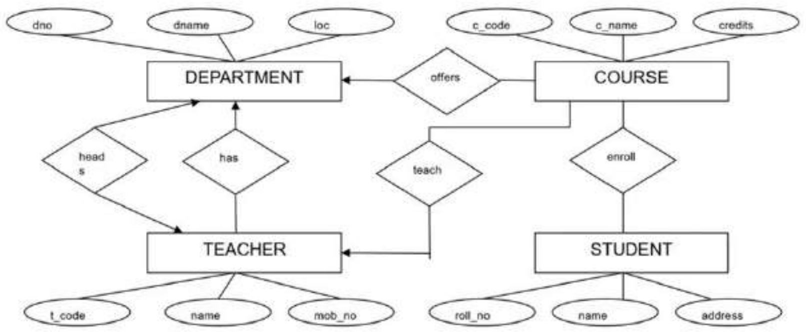
RNo	Name	Class	MobNo
1	A	BE	M1
2	A	BE	M2
3	B	BE	M3
4	C	ME	M4
5	B	ME	M5



SrNo	SNo	PNo	JNo	Qty
1	S1	P1	J1	100
2	S1	P1	J2	100
3	S2	P1	J1	100
4	S3	P1	J9	100
	S2	P1	J2	100

Case Study of University Management System

- Consider, a university contains many departments. Each department can offer any number of courses. Many teachers can work in a department. A teacher can work only in one department. For each department there is a Head. A teacher can be head of only one department. Each teacher can take any number of courses. A course can be taken by only one instructor. A student can enroll for any number of courses. Each course can have any number of students.

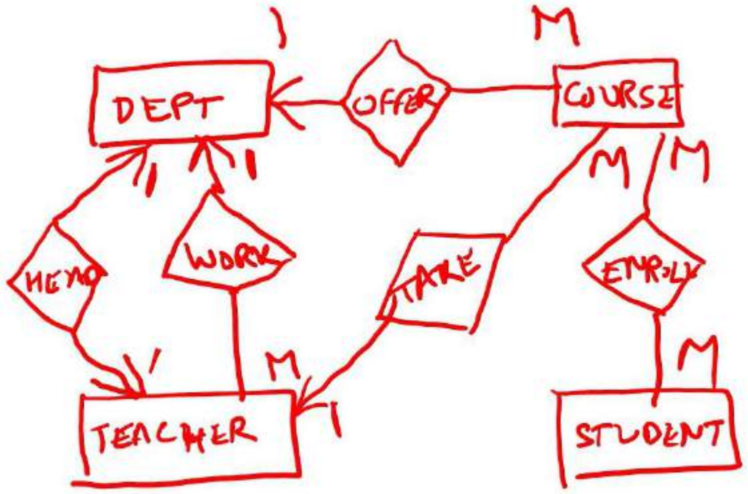


Case Study of University Management System

- Consider, a university 1 contains many departments. 2
- Each department can offer any number of courses. 3
- Many teachers can work in a department. A teacher can work only in one department. 4
- For each department there is a role of Head which is performed by the Teacher. A teacher can act as head of only one department.
- Each teacher can take any number of courses. A course can be taken by only one instructor.
- A 5 student can enroll for any number of courses. Each course can have any number of students.

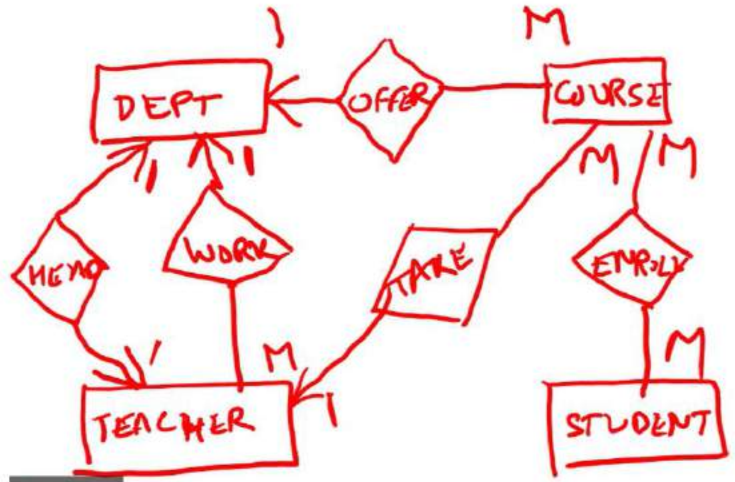
Second Step:
To find relationships among these entities

- Consider, a university contains many departments.
- Each department can offer any number of courses.
- Many teachers can work in a department. A teacher can work only in one department.
- For each department there is a role of Head which is performed by the Teacher. A teacher can act as head of only one department.
- Each teacher can take any number of courses. A course can be taken by only one instructor.
- A student can enroll for any number of courses. Each course can have any number of students.



Step 3:
To identify the key attributes

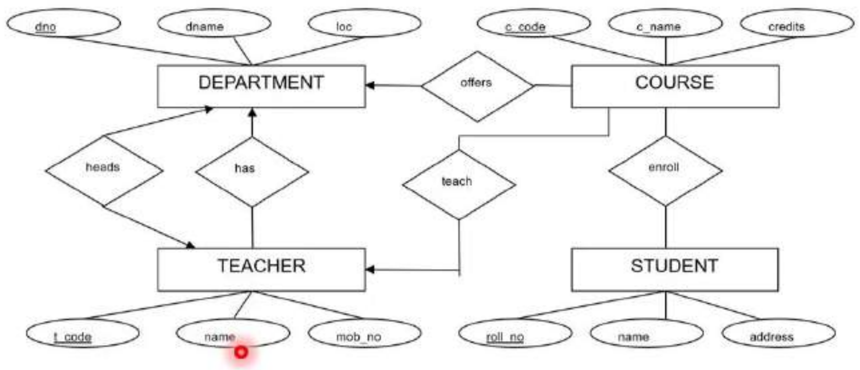
- Following are the primary key attributes for each entity set:
- Dno (Department number) is the key attribute for the Entity DEPARTMENT.
- C_code (Course number) is the key attribute for COURSE Entity.
- Roll_no (Roll number) is the key attribute for STUDENT Entity.
- T_code (Teacher code) is the key attribute for TEACHER Entity.



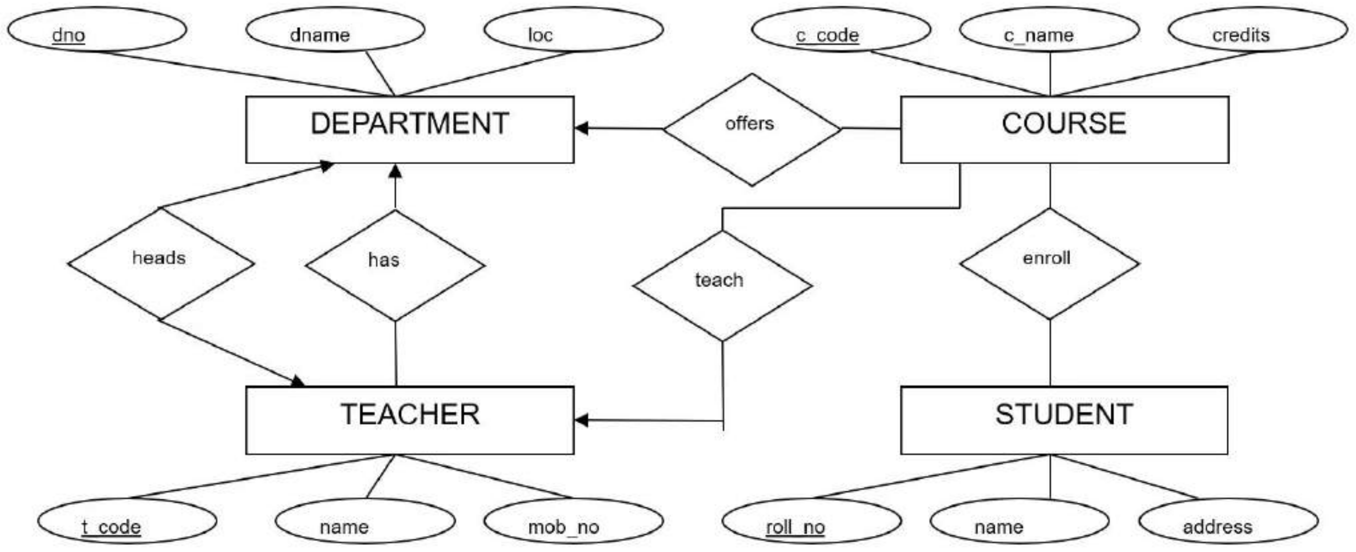
00:05:44

Step 4:
To identify other relevant attributes

- Following are the other relevant attributes for each entity set:
- DEPARTMENT entity will have other relevant attributes as dname, loc.
- For COURSE entity, c_name, credits.
- For TEACHER entity, name, mob_no
- For STUDENT entity, name, address



Step 5:
To draw the complete ER diagram

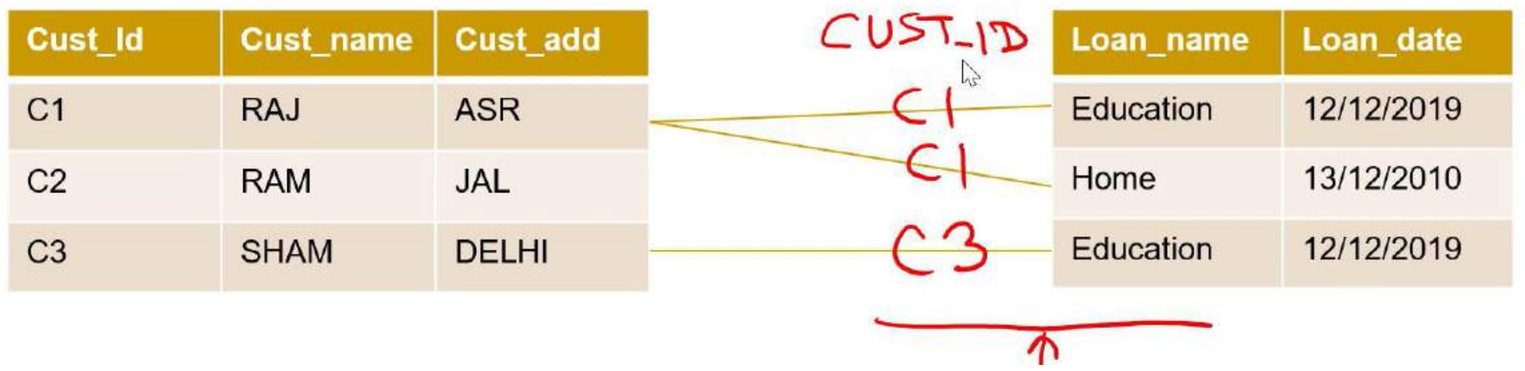


===

===

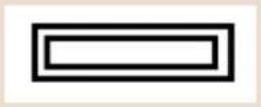
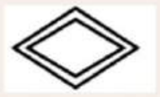
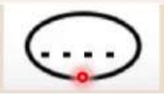
================================

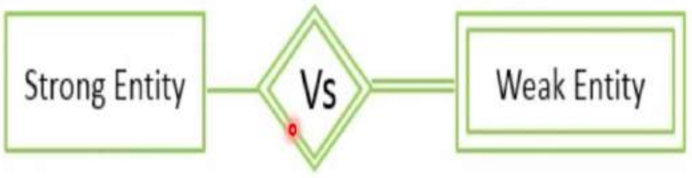
Strong and Weak Entity Set


CUST_ID
C1
C1
C3

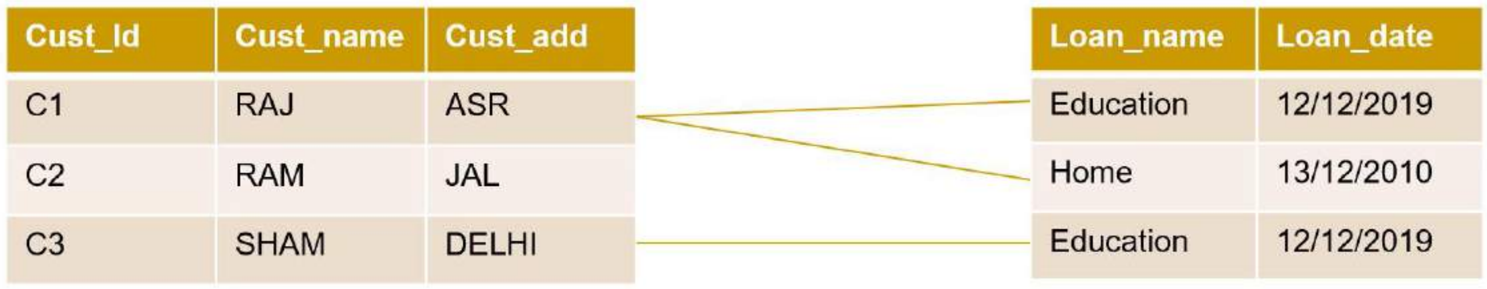
Cust_Id	Cust_name	Cust_add
C1	RAJ	ASR
C2	RAM	JAL
C3	SHAM	DELHI

Loan_name	Loan_date
Education	12/12/2019
Home	13/12/2010
Education	12/12/2019

Symbol	Description
	Weak Entity Set
	Identifying Relationship
	Discriminator or Partial Key

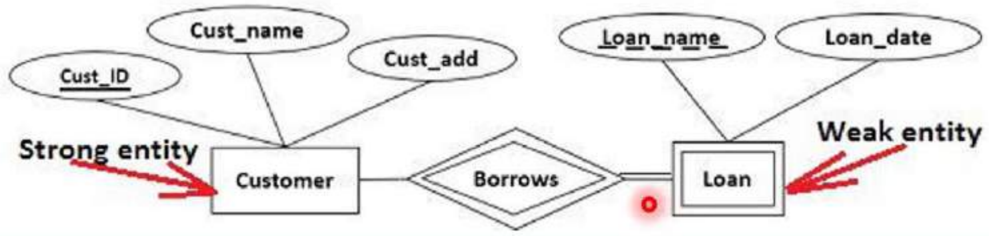


Customer and Loan



Cust_Id	Cust_name	Cust_add
C1	RAJ	ASR
C2	RAM	JAL
C3	SHAM	DELHI

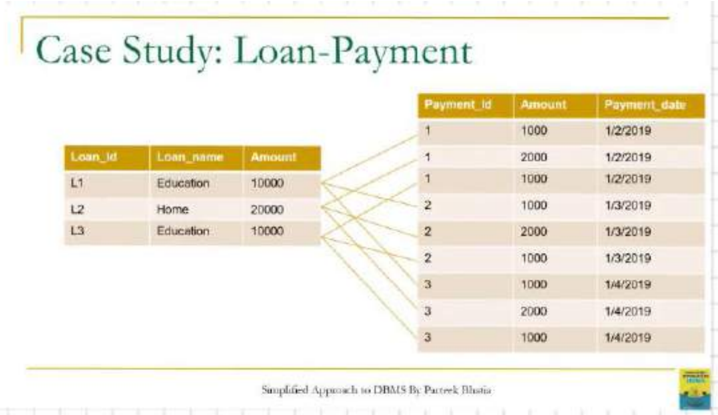
Loan_name	Loan_date
Education	12/12/2019
Home	13/12/2010
Education	12/12/2019



Case Study: Loan-Payment

Loan_Id	Loan_name	Amount
L1	Education	10000
L2	Home	20000
L3	Education	10000

LOANID	Payment_Id	Amount	Loan_date
L1	1	1000	1/2/2019
L2	1	2000	1/2/2019
L3	1	1000	1/2/2019
L1	2	1000	1/3/2019
L2	2	2000	1/3/2019
L3	2	1000	1/3/2019
L1	3	1000	1/4/2019
L2	3	2000	1/4/2019
L3	3	1000	1/4/2019

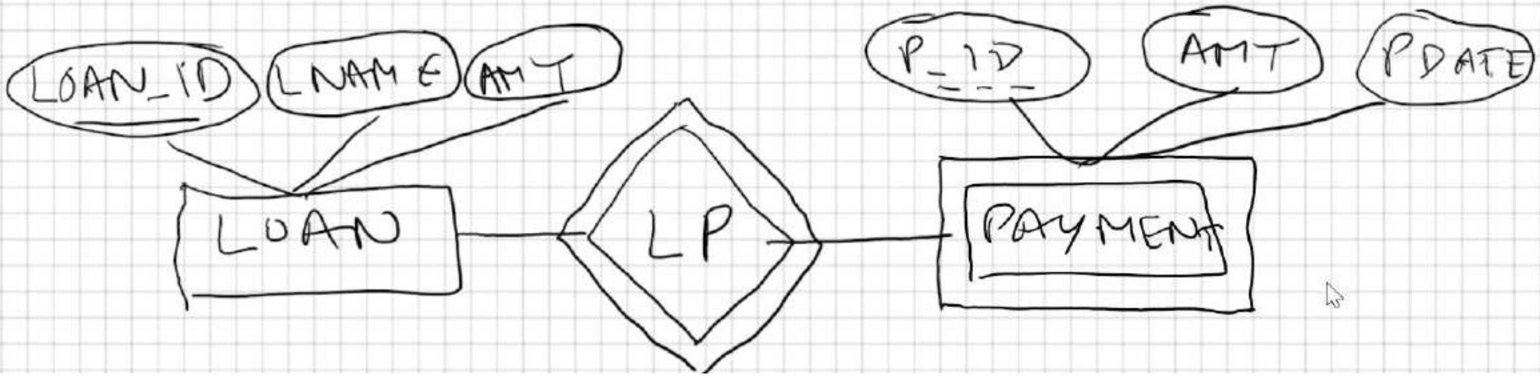


Case Study: Loan-Payment

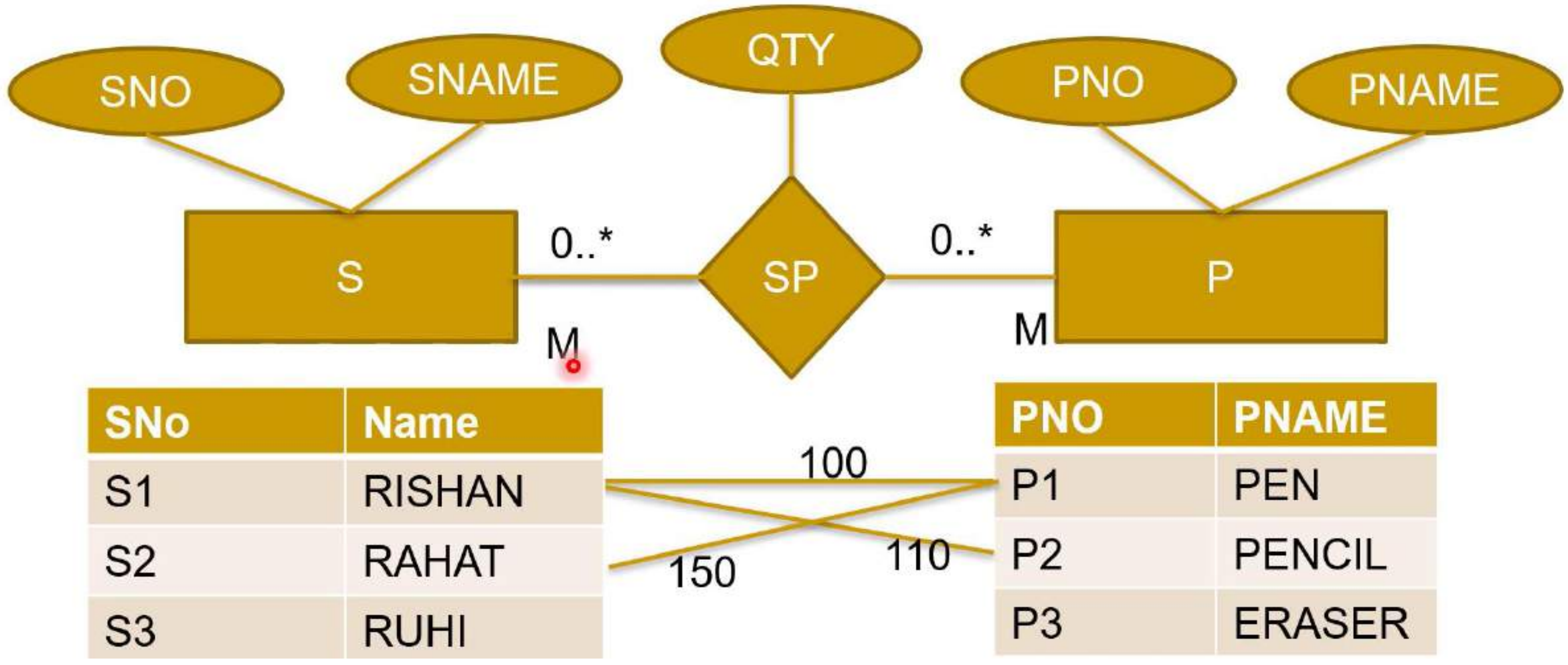
Loan_Id	Loan_name	Amount
L1	Education	10000
L2	Home	20000
L3	Education	10000

Payment_Id	Amount	Payment_date
1	1000	1/2/2019
1	2000	1/2/2019
1	1000	1/2/2019
2	1000	1/3/2019
2	2000	1/3/2019
2	1000	1/3/2019
3	1000	1/4/2019
3	2000	1/4/2019
3	1000	1/4/2019

Simplified Approach to DBMS By Pateek Bhatia



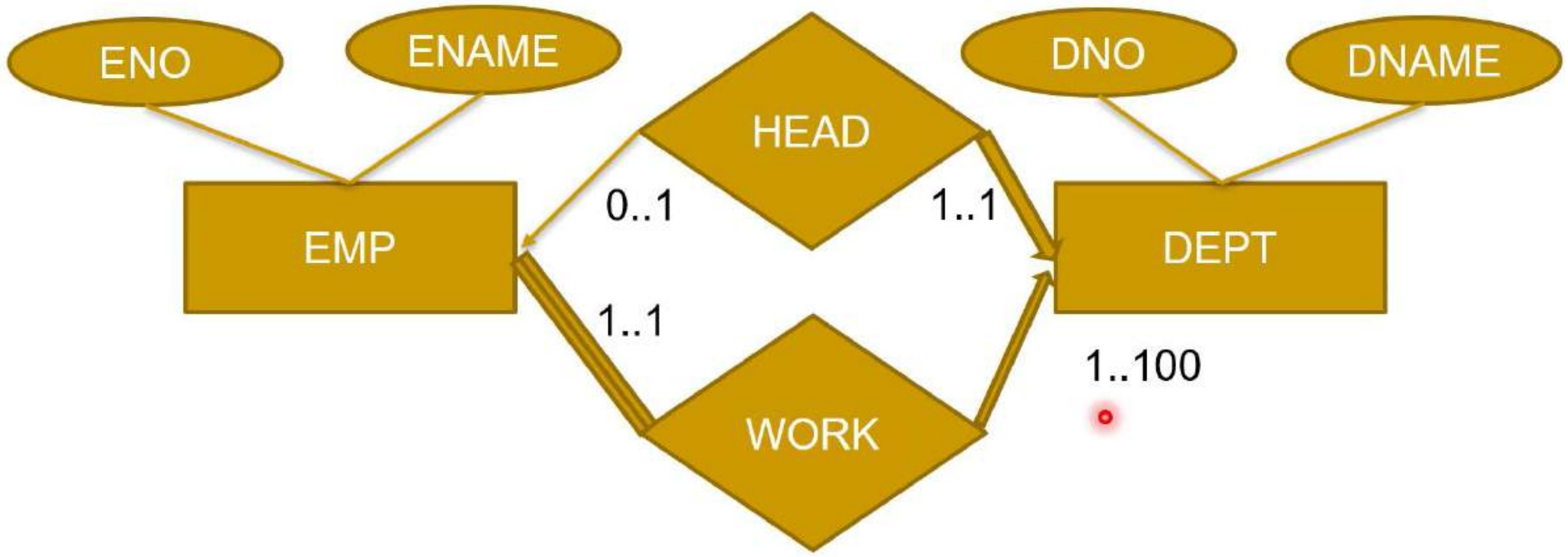
SUPPLIER-PART Relationship



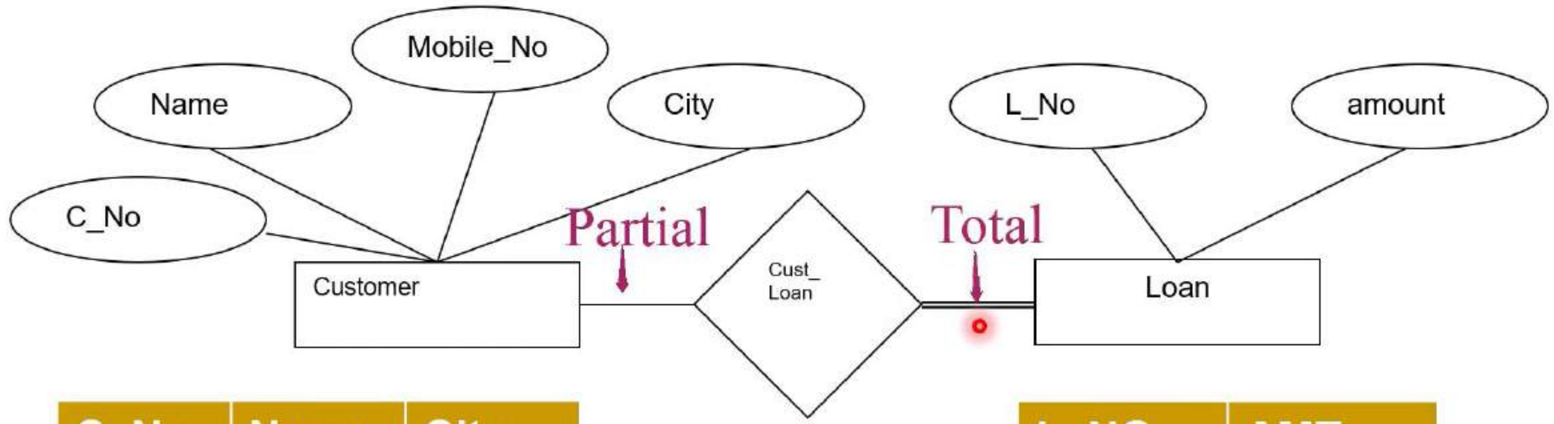
SNo	Name
S1	RISHAN
S2	RAHAT
S3	RUHI

PNO	PNAME
P1	PEN
P2	PENCIL
P3	ERASER

EMP-DEPT Relationships



CUST-LOAN Relationship



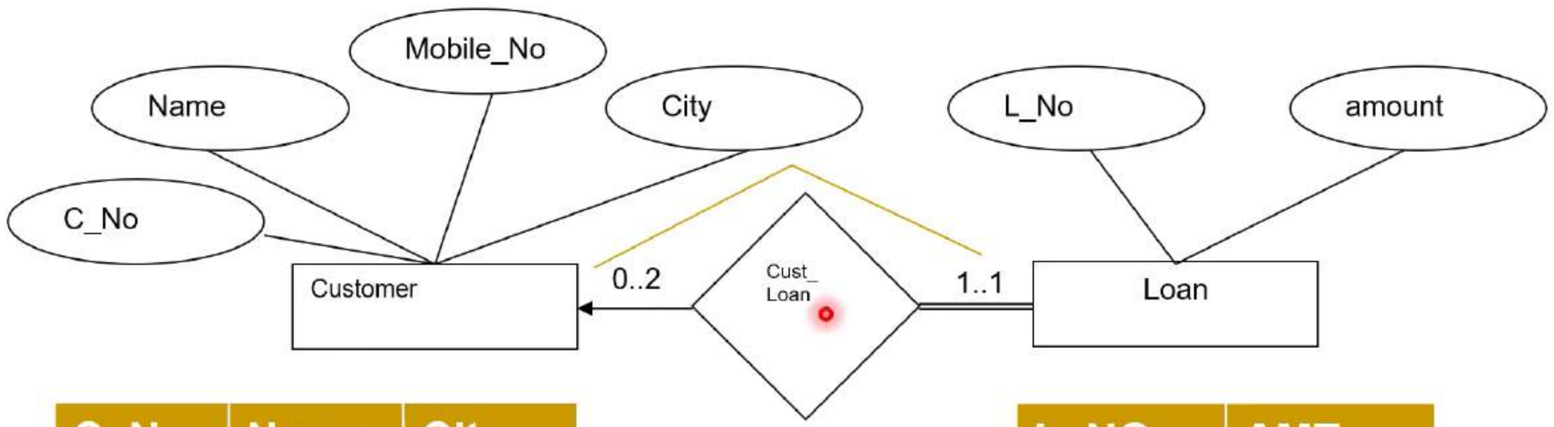
C_No	Name	City
C1	RAJ	ASR
C2	RAM	JAL
C3	SHAM	PTA

L_NO	AMT
L1	10000
L2	20000
L3	10000

Difference between Connectivity and Cardinality of Relationship

Connectivity	Cardinality
The connectivity of a relationship describes the mapping of associated entity instances in the relationship. The values of connectivity are "one" or "many".	The cardinality of a relationship is the actual number of related occurrences for each of the two entities.
Four Types: One to One (1:1) One to Many (1:M) Many to One (M:1) Many to Many (M:M)	An edge between an entity set and a relationship set can have an associated minimum and maximum cardinality, shown in the form *l..h*, where *l* is the minimum and *h* the maximum cardinality.

CUST-LOAN Relationship



C_No	Name	City
C1	RAJ	ASR
C2	RAM	JAL
C3	SHAM	PTA

L_NO	AMT
L1	10000
L2	20000
L3	10000