Network Model

By DBTG   (Database Task Group)   in   1971

DEPT

D2

DEPARTMENT

DEPT

D3

DEPT

D1

DEPARTMENT

DEPARTMENT

EMP

E10

EMP

E3

EMP

E4

EMP

E22

EMP

E12

EMP

E13

EMP

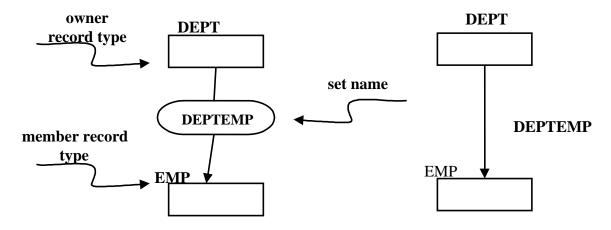E31

EMP

E16

EMP

E33

Fig 24.1   The Department-employee database

owner record type

DEPT

set name

DEPT

DEPTEMP

DEPTEMP

member record type

EMP

EMP

Fig 24.2   Structure of the set DEPTEMP

Three level network example

- a record type may be both a owner and member of two set types
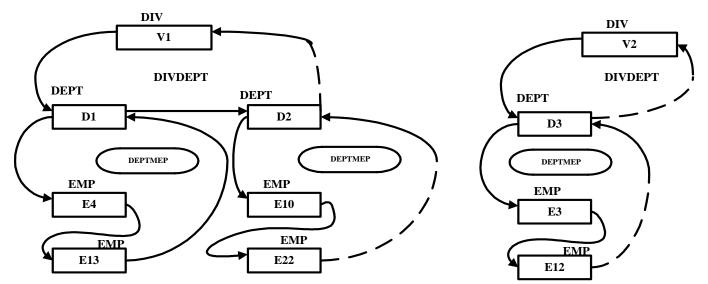

Fig   24.3   The division-department-employee database
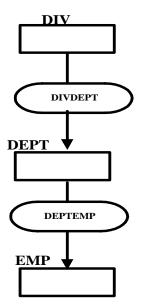
The set construct: Hierarchical Examples

Fig   24.4   Structure of the sets DIVDEPT and DEPTEMP
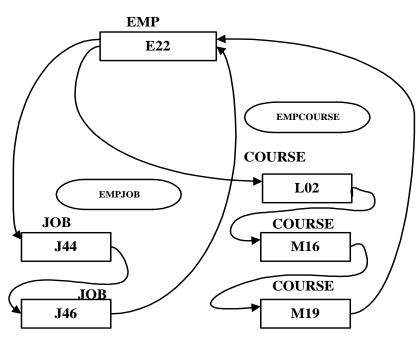
One owner with two members

EMP

| E22 |

EMPCOURSE

COURSE

| L02 |

EMPJOB

JOB

| J44 |

COURSE

| M16 |

JOB

| J46 |

COURSE

| M19 |

Fig   24.5   The employee-history database

EMP

DBTG DATA Structure
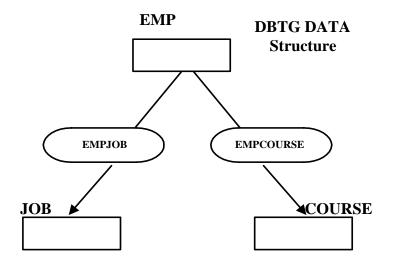
EMPJOB

EMPCOURSE

JOB

COURSE

Fig   24.6   Structure of the sets EMPJOB and EMPCOURSE
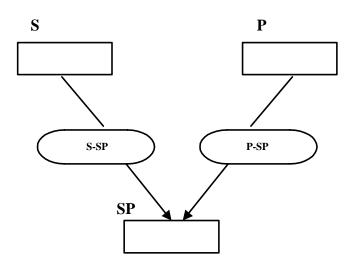
Many-to-many relationship

Fig   24.14 Structure of the sets S-SP and P-SP.
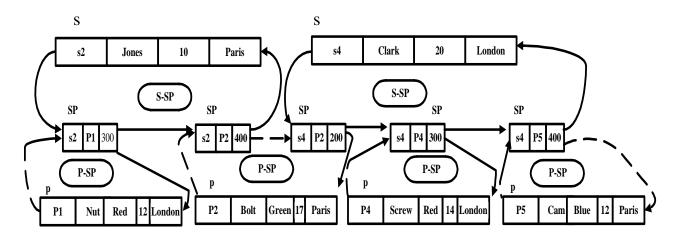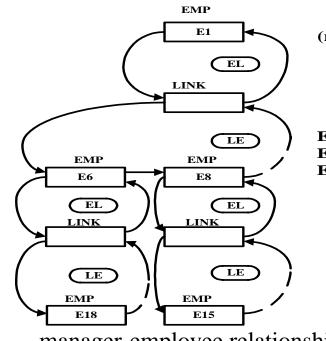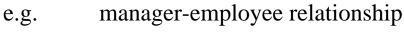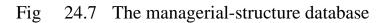

Fig   24.25 (Part of) the suppliers-and-parts database

Recursive relationship

EMP

E1   **(not good)**

EL

LINK

LE   **E1 is the manager of E6 and E8**
E6 is the manager of E18
E8......E1

EMP   **EMP**
E6   **E8**

EL   **EL**
LINK   **LINK**

LE   **LE**

EMP   **EMP**
E18   **E15**

e.g.   manager-employee relationship

Fig   24.7   The managerial-structure database

EMP   **better solution**

EL   **LE**   **EMP**

E#	...

LINK   **LINK**

E#	M#

Fig   24.8   Structure of the sets EL and LE

Note: EMP

not allowed

better solution

Emp#	Mgr#
E6	E1
E8	E1
E18	E6
E15	E8

EMP

Emp#	...

superior   subordinate

LINK

Emp#	Mgr#

EMP

E1

Subordinate

LINK

E6	E1

LINK

E8	E1

EMP

E6

EMP

E8

LINK

E18	E6

LINK

E15	E8

EMP

E18

EMP

E15

Part and Subpart relationship

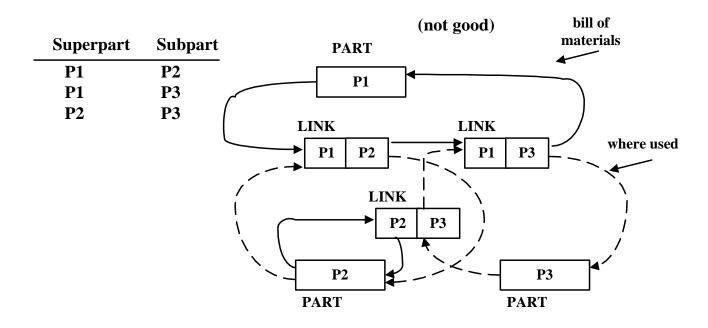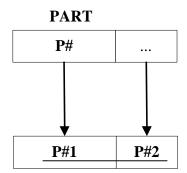| Superpart	Subpart
P1	P2
P1	P3
P2	P3

(not good)

Fig   24.11   The parts database.

- P1 contains P2 and P3 as components
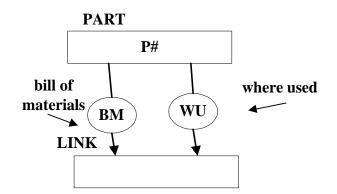- P2 also contains P3 as components

better solution

Fig   24.12 Structure of the sets BM and WU

MEMBER CLASS

Each member subentry in the schema must include a specification of the <u>membership class</u> for the record type concerned in the set type concerned. Membership class is specified by means of the **INSERTION/RETENTION** entry, and may therefore be thought of as a combination of an insertion class and a retention class. The insertion class is **AUTOMATIC** or **MANUAL**. The retention class is **FIXED**, **MANDATORY**, or **OPTIONAL**.

- <u>Retention class</u>   (**FIXED** or **MANDATORY** or **OPTIONAL**)

    If the membership of M in OM is **<u>FIXED</u>**, then once an occurrence of M (m, say) has been entered into an occurrence of OM, it can never have any existence in the database *not* as a member of that occurrence of OM. Specifically, it can never be taken out of the OM occurrence by means of a <u>DISCONNECT</u> operation, nor may it be transferred from one OM occurrence to another by means of a <u>RECONNECT</u> operation. The only way to destroy the association between *m* and OM is by deleting m entirely from the database by means of an ERASE operation, Note the implication here that if an occurrence of O is deleted (erased), all corresponding occurrences of M must also be deleted.

    If the membership of M in OM is **<u>MANDATORY</u>**, then once an occurrence of M (*m*, say) has been entered into an occurrence of OM, it can never have any existence in the database not as a member of *some* occurrence of OM. Specifically, it can never be taken out of the OM occurrence be means of DISCONNECT operation, but it may be transferred from one OM occurrence to another by means of <u>RECONNECT</u> operation.

    Finally, if the membership of M in OM is **<u>OPTIONAL</u>**, an occurrence of M can be removed from an occurrence of OM (e.g., by means of a DISCONNECT operation) without being entirely erased from the database.
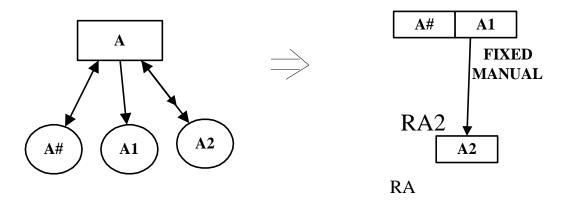
- Insertion class (**AUTOMATIC** or **MANUAL**)

    If the membership of M in OM is **<u>AUTOMATIC</u>**, then when an occurrence of M (*m*, say) is first created and placed in the database (by means of a <u>STORE</u> operation), the DBMS will automatically connect it into the appropriate occurrence of OM. (It is in general up to the program that stores *m* does not cause this automatic connection; see section 24.7.) On the other hand, if the membership of M in OM is **<u>MANUAL</u>**, storing an occurrence does not cause this automatic connection; to connect *m* into an OM occurrence, the program must be issue an explicit <u>CONNECT</u> operation.

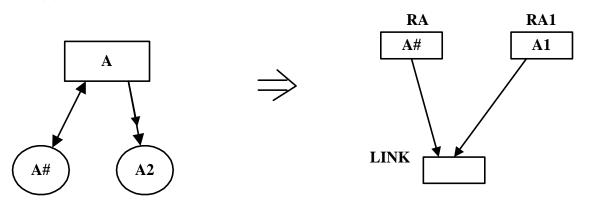# Converting an ER diagram to a DBTG diagram

__Rule 1__   For each entity type A, all its single value attribute form
a record type, say RA, in the network database.
Each 1: m multivalued attribute of A form a record
which connects to the record type RA with membership
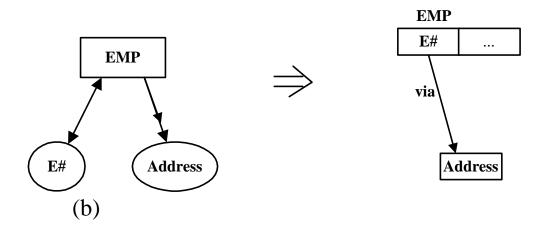class as FIXED and MANUAL.

Membership class: (1) Retention class: fixed, mandatory, optional
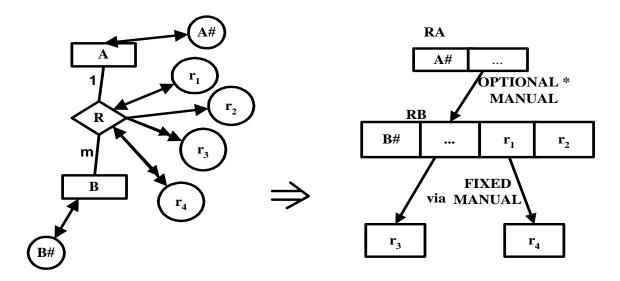(2) Insertion class: automatic, manual

For each m : n multivalued attribute of A, we can have
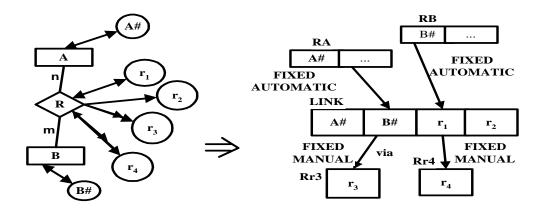several ways to translate it.

(a)

(b)

Rule 2 For 1 : m binary relationship type

Note: all single valued attributes R become attribute of
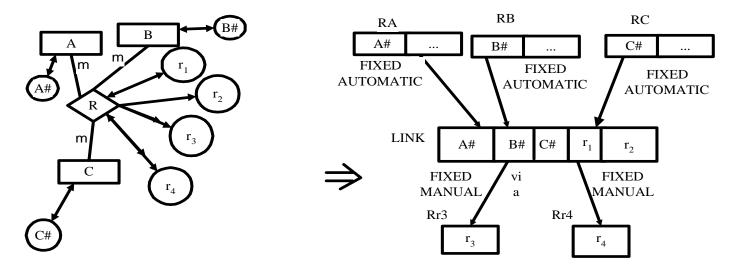the record type RB.

Note: B# and r_1 are unique in RB.

* The relation class is **MANDATORY** if the participation of
entity type B in the relationship set R is **mandatory**.
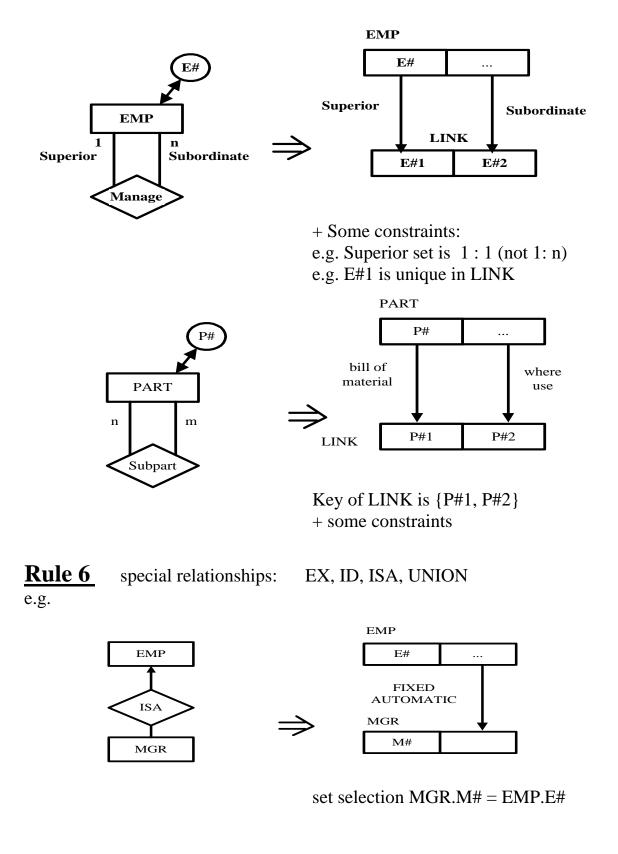
Rule 3   For m : n binary relationship type

Note: need a new adjunct record type **LINK**.
  r1 is unique in LINK, {A#, B#} is also unique in LINK.

Rule 4   For any n-ary (n > 2) relationship type

Note: r_1 and {A#, B#, C#} are unique in LINK.

Rule 5   Recursive relationship type.

+ Some constraints:
e.g. Superior set is 1 : 1 (not 1: n)
e.g. E#1 is unique in LINK

Key of LINK is {P#1, P#2}
+ some constraints

Rule 6   special relationships: EX, ID, ISA, UNION
e.g.

set selection MGR.M# = EMP.E#