Normal Form for Sets of Nested Relations

1. Background

2. A Norm Form for sets of nested relations (NF - nested relations)

3. A Decomposition method for NF-nested relations design

4. NF - nested relation schema Design using Entity - Relationship Approach.

Terms:

- *non-first normal form (NF^2) relation*
- *not-necessarily normalized (N3) relation*
- *nested relation*

1. Background

- A **non-first normal form (NF2) relation** is a relation which is not in *first normal form (1NF)*,

  ie. it may contain repeating groups (possibly nested)

  e.g. Emp (Emp#, Name, Address, (Child),
  Sal-history (Date, Salary))

- Codd recognized that unnormalized relations may contain redundancies and cause updating anomalies

- Normalize the NF2 relations to 1NF,then subsequently to 2NF, 3NF, BCNF, etc.

- A **not-necessarily normalized relation (N3)** or **nested relation** is a relation which is either a NF2 relation or a normalized relation.

- Many applications are more suitably represented by NF2 relations than normalized relations.

  e.g. Employee relation, invoices or purchase orders, picture data processing, etc.

2. <u>A Normal Form for Sets of Nested Relations</u>

Example.

Emp (Emp#, Name, Address, (Child),
Sal-history (Date, Salary))

Emp					
Emp#	Name	Address	(Child)	Sal-history	
				Date	Salary
001	Ang	Kent Ridge, 0511	mei ling	Mar 1980	2,400
		Singapore	Hock	Jan 1981	2,700
			sing	Jan 1982	3,100
003	Chan	Main street, 1057		July 1999	1,500
		Singapore		Jan 1980	1,600
				Jan 1981	1,800
				Jan 1982	1,950

Form 1: N3 relation Emp

- This N3 relation Emp has two repeating groups, namely, (Child) and Sal-history (Date, Salary)
- We allow *nested* repeating groups.
- The 2 repeating groups in Emp are **first level repeating groups.**
- A repeating group of a **first level** repeating group of R is called a **second level-repeating group** of R.

Definition:

Let A be a set of single valued attributes, and B be a set of single valued attributes and/or repeating groups of a nested relation R.

B is **extended functionally dependent** on A, denoted by

$$A \Rightarrow B$$

if whenever two tuples of R agree on all columns of A, then they also agree on columns of B.

e.g. Emp# $\Rightarrow$ Name, Address, (Child), Sal-history (Date, Salary)

<u>Theorem 1.</u>

Properties for extended functional dependencies

(1) $A \Rightarrow (B)$ iff $A \rightarrow\rightarrow B$

(2) If B is a set of single valued attributes, then $A \Rightarrow B$ is the same as $A \rightarrow B$

(3) $A \rightarrow B$ and $B \Rightarrow C$ implies $A \Rightarrow C$

(4) $A \Rightarrow B$ and C is a single valued attribute implies $AC \Rightarrow B$

- Let K be a set of single valued attributes of a nested relation R and A be the set of all single valued attributes and (first level) repeating groups of R.

  K is called a **candidate key** (or simply **key**) of R iff

  (1) $K \Rightarrow A$
  (2) there is no proper subset K_1 of K such that $K_1 \Rightarrow A$

- **Prime attribute**
  Non-prime attribute
  Primary key

- Let K be the primary key of a nested relation R and let K_1 be a set of single valued attributes in a first level repeating group G of R and A be the set of all **first level attributes** of G

  K_1 is called a **candidate key** (or simply **key**) of the repeating group G iff

  (1) $K, K_1 \Rightarrow A$
  (2) there is no proper subset K_2 of K_1 such that:
  $K, K_2 \Rightarrow A$

- Given a nested relation R with K as its primary key, **the set of basic dependencies** of R, denoted as BD(R) is defined as:

(1) All non-trivial full FDs which involve single valued attributes of R and of the form A $\rightarrow$ B such that A is a key of R or B is part of a key of R, is in BD(R).

(2) For each (first level) repeating group G of R, the extended functional dependency k $\Rightarrow$ G is in BD(R).

(3) For each (first level) repeating group G of R, the set of basic dependencies of the relation formed by K and attributes of G, is a subset of BD(R).

e.g. Emp (Emp#, Name, Address, (Child),
Sal-history (Date, Salary))

BD(Emp) = { Emp# $\rightarrow$ Name, Address
Emp# $\Rightarrow$ (Child), Sal-history (Date, Salary)
Emp#, Date $\rightarrow$ Salary}

Emp# is the only key of the nested relation EMP. Child and Date are the keys of the repeating groups (Child) and Sal-history (Date, Salary)

Definition 1.

A nested relation R (with primary key K) is called a **normal form nested relation (NF - N relation)** iff

(1) The relation R has at least one key.

(2) All the single valued attributes of R form a 3NF relation with all keys of R as its keys.

(3) If there is no repeating group in R and R is all key, then R is also in 4NF.

(4) For each repeating group G of R, all the attributes of G and K form an NF-nested relation.

(5) Any non-trivial extended FD (ie. not a trivial FD) which involves only single valued attributes, any level of repeating groups, and/or single valued attributes of any level of repeating group of R, can be derived from BD(R).

Note:

Condition (5) ensures that there is no non-trivial FDs among repeating groups which can not be derived from BD(R).

Example:

DEPARTMENT (D#, Clerk (Ce#, C-age),
Engineer (Ee#, E-age, Experience (ED#, Date)))

IF we assume that each clerk and engineer only works for one department,

THEN the nested relation DEPARTMENT is in NF-N.

ELSE DEPARTMENT is not an NF-nested relation (since relation formed by the repeating groups clerk or Engineer is not in 3NF)

Theorem 2:

(1) All 4NF and 5NF relations are NF-nested relations.

(2) A 3NF or BCNF relation which is not an all key relation, is an NF-nested relation.

(3) Any **1NF** relation which is not in 3NF, is also not in NF-N.

<u>Definition 2.</u>

A **set** of nested relations S is said to be in **normal form (NF-N)** iff

(1) Each nested relation of S is an NF-nested relation.

(2) Given any relation R of S and any non-trivial extended FD $A \Rightarrow B$ in BD(R), $A \Rightarrow B$ **can not** be derived from the union of the set of basic dependencies of relations in S', where S' is the same as S except the attribute B' is removed from the relation R.

Note: The second condition ensures that there is no redundancy (due to redundant dependencies) among the NF-nested relations of a database.

<u>Example:</u>

A database consists of three nested relations:

R_1 (A, B, (C))
R_2 (B, (D))
R_3 (A, (D))

With $A \rightarrow B$, $A \Rightarrow (C)$, $B \Rightarrow (D)$, $A \Rightarrow (D)$

- Clearly each nested relation is in NF-N.

- The database is not in NF-N since $A \Rightarrow (D)$ can be derived from $A \rightarrow B$ and $B \Rightarrow (D)$

NF-nested relations, Hierarchical Databases, Forms

Example: Assume that:

1. O# is only unique for each course
2. C#, O#, S# $\rightarrow$ grade and C#, S# $\not\rightarrow$ grade

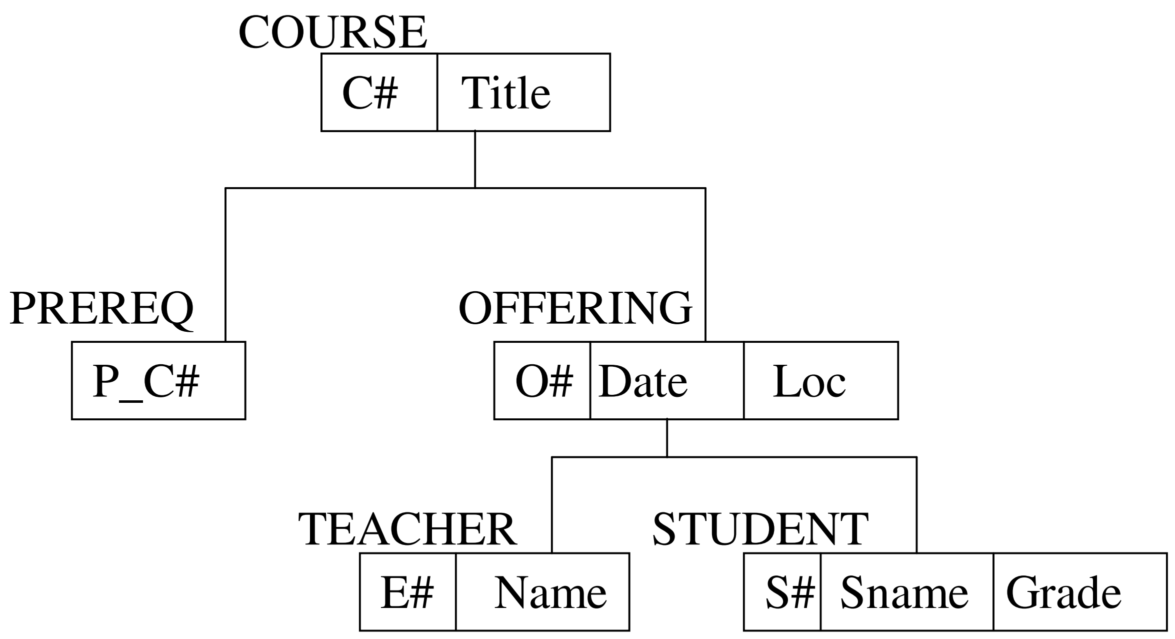

Figure 1. A Hierarchical definition tree

The above hierarchical definition tree can be viewed as the following nested relation

COURSE (C#, Title, PREREQ (P_C#),
OFFERING (O#, Date, Loc, TEACHER (E#, Name),
STUDENT (S#, Sname, Grade)))

with extended FDs:

C# ⇒ Title, PREREQ (P_C#),
OFFERING (O#, Date, Loc, TEACHER (E#, Name),
STUDENT (S#, Sname, Grade))

C#, O# ⇒ Date, Loc, TEACHER (E#, Name),
STUDENT (S#, Sname, Grade)

C#, O#, S# → Grade, Sname

E# → Name

S# → Sname

The nested relation is not in NF-N. If we drop Name and Sname from figure 1, then the corresponding N relation is in NF-N.

COURSE										
		Prereq	Offering							
						Teacher		Student		
C#	Title	P_C#	O#	Date	Loc	E#	Name	S#	Sname	Grade

Form 2. Relation COURSE

Comparing NF-N with other Normal Forms

1. Unlike our definition, other definitions given in [Makinouchi], [Roth & Korth], and [Ozsoyoglu & Yuan] only apply to **single** relations , and hence global redundancy among relations of a database may occur.

2. The NNF relation defined in [Ozsoyoglu & Yuan] only considered MVDs, it will not be able to distinguish the essential difference between

   $A \rightarrow BC$, and $A \rightarrow\rightarrow B \mid C$

3. Unlike the definition of an extended FD given in [Makinouchi], we do not allow repeating groups to appear as part of the left side of an extended FD (This can be avoided).

4. Unlike the definition given in [Ozsoyoglu & Yuan], the relation formed by all the single valued attributes of an NF-N relation may not be in BCNF (but definitely in 3NF).

5. An NF relation defined in [Makinouchi] does not allow the existence of FDs which define among the key of a relation and the attributes of a repeating group.

   e.g. Student (Student#, Name,(Course#, Mark))

   with

   Student#, Course# $\rightarrow$ Mark
   Student# $\Rightarrow$ Name, (Course#, Mark)

   is an NF-N relation but not an NF relation.

A Decomposition Method for NF-N Relations Design

(D1) *(Remove Transitive dependencies)*

If there is a non-prime attribute of a nested relation R which is **transitively dependent** on some key of R, decompose the relation to 2 relations

(D2) *(Remove non-trivial MVD)*

Let R be a 1NF relation . IF all attributes of R can partitioned into 3 sets A, B, C such that:

$$A \rightarrow\rightarrow B \mid C$$

and A is not a key of R, then make B and C as 2 repeating groups of R.

(D3) *(Remove non-trivial MVD)*

Let G to be a repeating group of a nested relation R (with key K). If G can be partitioned into 2 non-empty subsets, say A and B, such that

$$K \rightarrow\rightarrow \quad A \mid B$$

is a non-trivial FD, decompose the repeating group G into 2 repeating groups, A and B of R.

(D4) *(Remove transitive dependencies)*

Let B be a single valued attribute of some level of repeating group of a nested relation R and A be a set of single valued attributes of R which is neither a key nor a super key of R.

If A$\rightarrow$ B is a full dependency, remove B from R and create a new relation which consists of attributes of A and B.

(D5) *(Remove extended transitive dependencies)*

Let G be an any level repeating group of a nested relation R and A be a set of single valued attributes of R which is neither a key nor a super key of R.

If A $\Rightarrow$ G, remove G from R and create a new nested relation with A as its single valued attributes and G as its repeating group.

<u>**Example:**</u>

Let R (<u>A</u>, B, C, (<u>D,</u> E, F), (<u>G</u>)) be a nested relation with:

$A \Rightarrow B, C, (D, E, F), (G)$
$C \Rightarrow (G)$
$A, D \Rightarrow E, F$
$B \rightarrow F$

(1) Since $C \Rightarrow (G)$ by rule D5, decompose R into:

R_1 (<u>A</u>, B, C, (<u>D</u>, E, F))
R_2 (<u>C</u>, (G))

(2) Since $B \rightarrow F$, by rule D4, decompose R_1 into:

R_{11} (<u>A</u>, B, C, (D, E))
R_{12} (<u>B</u>, F)

All relations R_{11}, R_{12}, R_2 are in NF-N.

Example:

Let R (A, B, C, (D, E, F)) be a nested relation:

$A \Rightarrow (D, E, F)$
$A \rightarrow\rightarrow B \mid C$
$A \rightarrow\rightarrow \{D, E\} \mid F$

(1) Since $A \Rightarrow (D, E, F)$

we have $A \rightarrow\rightarrow \{D, E, F\} \mid \{B, C\}$

Hence by rule D2, we have

R' (A, (B, C), (D, E, F))

(2) Since $A \rightarrow\rightarrow B \mid C$, by rule D3, we have

R''(A, (B), (C), (D, E, F))

(3) Similarly, since $A \rightarrow\rightarrow \{D, E\} \mid F$, by rule D3, we have

R'''(A, (B), (C), (D, E), (F))

R''' is now in NF-N.

4. <u>NF-nested Relations Design Using Entity-Relationship Approach</u>

- Entity-relationship (ER) approach (for database schema design) was proposed by [Chen 76].
- Concepts:
  - entity types
  - relationship types
  - recursive relationship types
  - multivalued attributes
  - special relationships. e.g. ISA, EX, ID, UNION, DECOMPOSE, INTERSECT, etc
  - entity-relationship diagram.

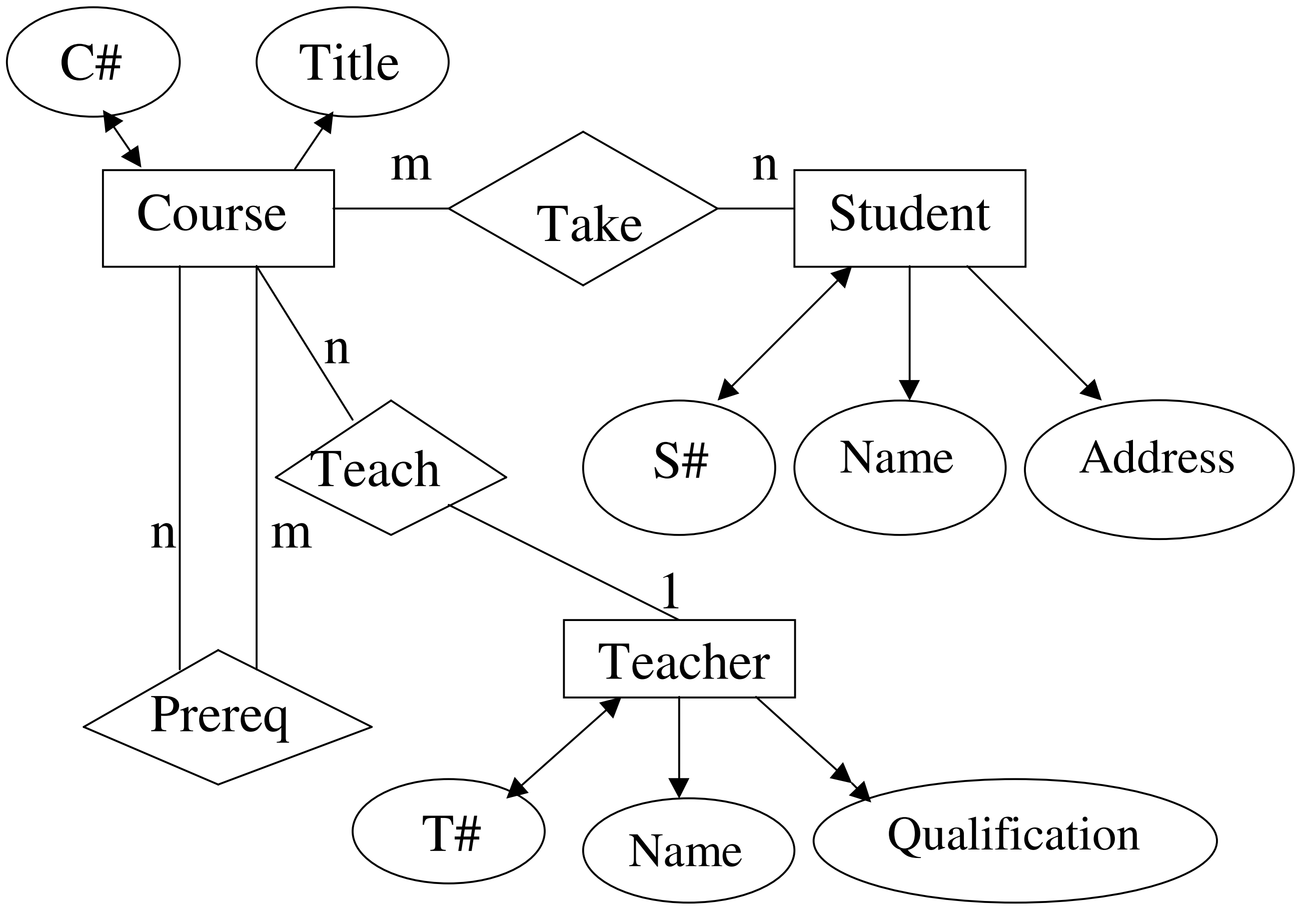

Figure 2. An ER diagram

- A **normal form ER diagram** was defined in [Ling 85] with the following objectives:

(1) To capture and preserve all the semantics of the real world of a database which can be expressed in term of FDs, MVDs, JDs, by representing them explicitly in the ER diagram.

(2) To ensure that all the relationship sets represented in the ER diagram are non-redundant.

(3) To ensure that all the relations translated from the ER diagram are in good normal form, either in 3NF or 5NF.

Example: A library system

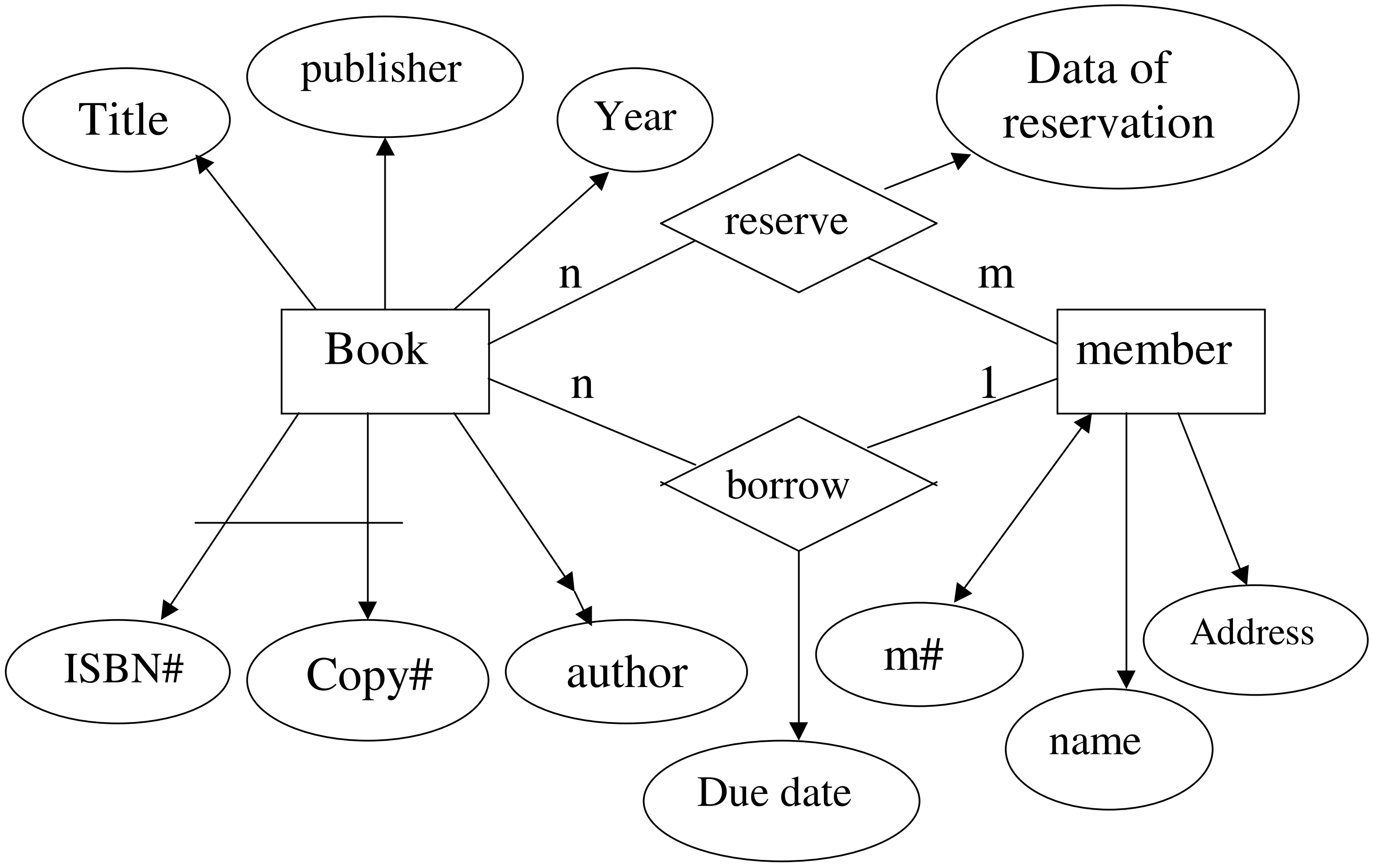

Figure 3. A non-normal form ER diagram for a library system

- The above ER diagram is **not** in normal form, since:

$$\text{ISBN\#} \rightarrow \text{title, publisher, year}$$
$$\text{ISBN\#} \rightarrow\rightarrow \text{author}$$

Also members reserve any copy of a book.

- A normal form ER diagram is as follow:

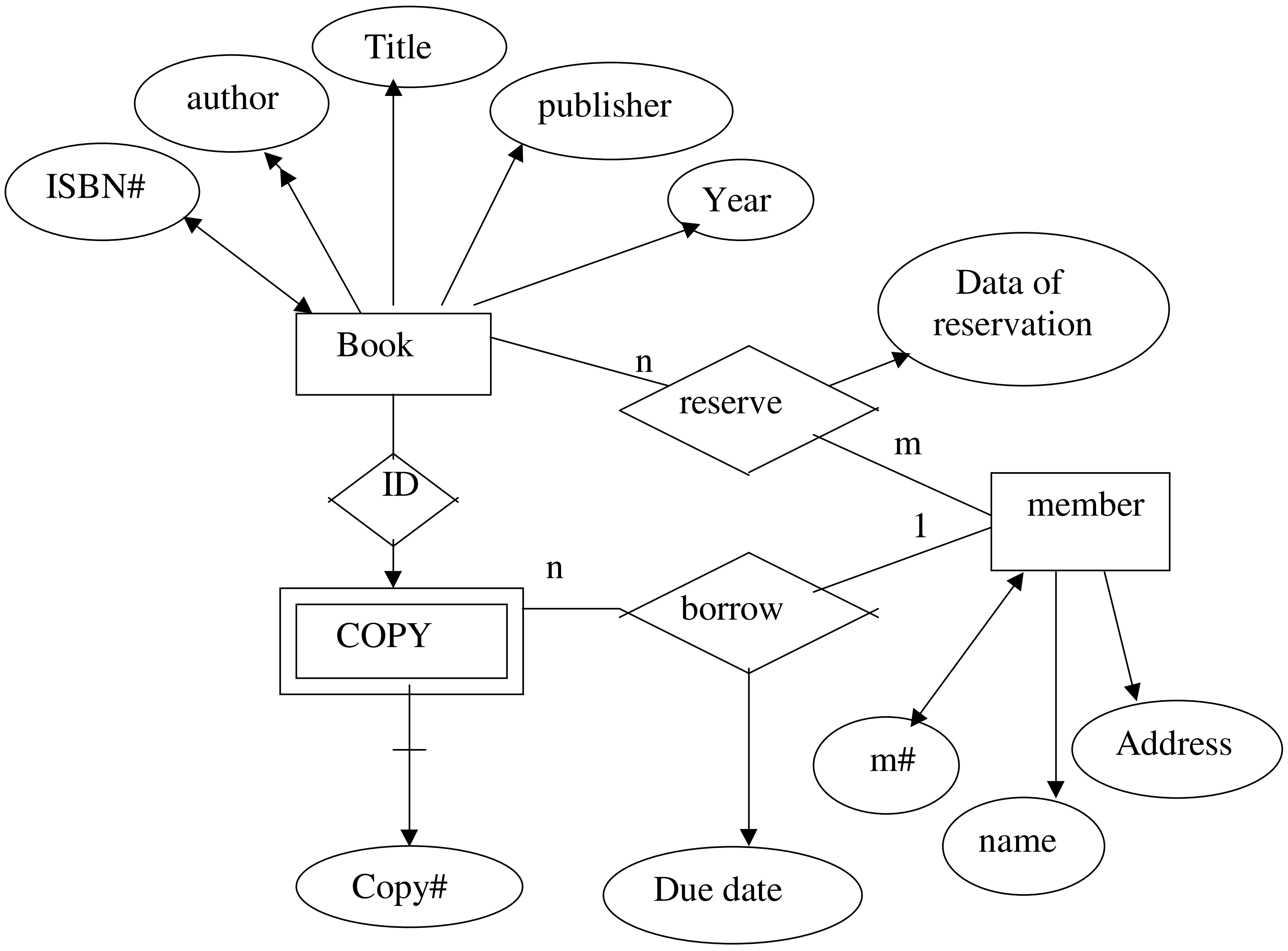

Figure 4. A normal form ER diagram for a library system

Alogorithm 1

The main steps involved in NF-nested relations design using ER approach are as follows:

Step 1. *Construct an ER diagram for the given database*

(a) Identify the entity types and their attributes.
(b) Identify the relationship types and their attributes.
(c) Identify the keys and identifier of each entity type and relationship type.

Step 2. *Convert the ER diagram constructed to a normal form ER diagram. [Ling 85]*

(a) Ensure that all attribute names are distinct and of different semantics.
(b) Convert each entity type to normal form entity type.
(c) Convert each relationship type to normal form relationship type.
(d) Remove redundant relationship types.

Step 3. *Translate the normal form ER diagram obtained to a set of nested relations.*

(a) For each **recursive relationship type** in the ER diagram, we assign each arc which connects an entity type and the recursive relationship type a **unique role name** if there is no role name assigned.

(b) {Assign identifier for entity types involved **ISA, UNION, INTERSECT, DECOMPOSE**, etc}

(i) Let entity types A and B be involved by A ISA B, and K be the identifier of B.

If A has no identifier, then assign a unique name say K_1, as the identifier of A. Else let the identifier of A be K_1.

Record the constraint:

K_1 (of A) ISA K (of B)

(ii) If entity types A_1, A_2, …, A_n, and B are involved by

B = UNION (A_1, A_2, .., A_n)

we handle this as $\mathbf{A_i}$ **ISA B** for i = 1, 2,…, n.

(iii) INTERSECT, DECOMPOSE, etc. special relationships can be handled similarly.

(c) {Generate a nested relation RE, for each regular entity type E}

(i) All the single valued attributes of E are single valued attributes of RE.

(ii) The identifier and keys of E are the primary key and keys of RE.

(iii) Each m:n multi-valued attribute of E is a repeating group of RE. The attributes of this repeating group is the key of the repeating group.

(iv) Each 1:m attribute B of E is a repeating group of RE with constraint $B \rightarrow K$, where K is the primary key of RE.

(v) All attributes of a weak entity type B whose existence is dependent on the existence of E, form an embedded relation of RE by applying step 3 recursively.

Note :

Regular entity types include entity types which involve with ISA relationship.

(d) {Generate a nested relation RR, for each *non-recursive* regular relationship type R}

(i) All the identifiers of the entity types participating in R and all the single valued attributes of R are the single valued attributes of RR.

(ii) The identifier, keys, and 1:1 attributes of R are the primary key and keys of RR.

(iii) Each m:n attribute of R forms a repeating group of RR.

(iv) Each 1:m attribute B of R is a repeating group of R with constraint $B \rightarrow K$ where K is the primary key of RR.

(v) If $C \rightarrow D$ is a non-trivial full FD in the set of dependencies of RR, D is a single attribute, and C is not a key of R, then

record constraint $C \rightarrow D$.

(e) {Generate a nested relation RR, for each *recursive relationship type* R}

The way to handle recursive relationship types is similar to the non-recursive relationship types except the "identifier" of the entity type participating in R is replaced by the "role name" of the entity type participating in R.

<u>**Example:**</u>

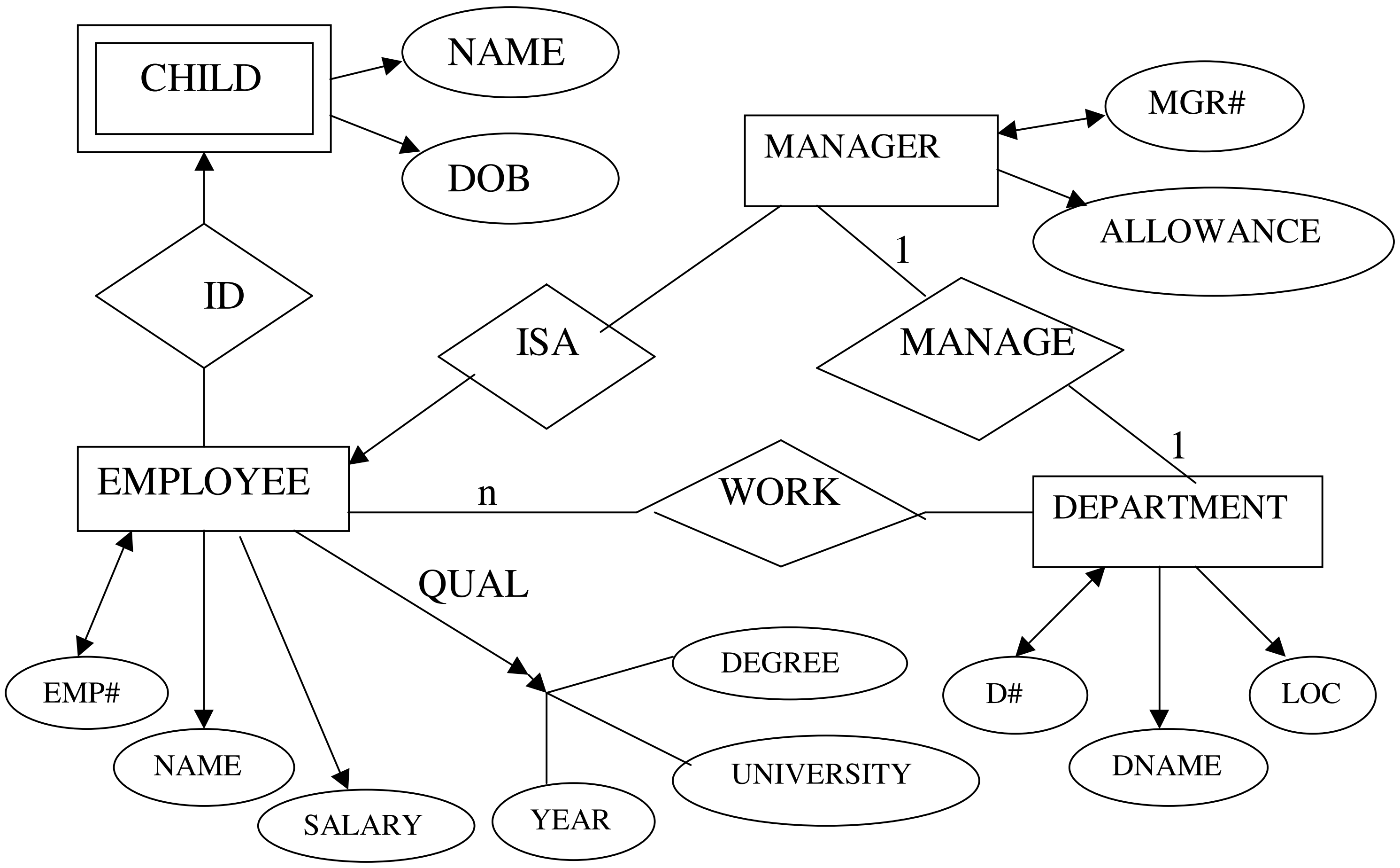

Figure 5. An ER diagram

- The ER diagram is in normal form.
- The identifier MGR# of entity type MANAGER is assigned in step3(b) of Algorithm 1.
- The set of nested relations generated by Algorithm 1:

(1) Relation for entity type **EMPLOYEE** is

EMPLOYEE (<u>Emp#</u>, Name, Salary,
Qual (<u>Degree, University, Year</u>),
Child (<u>Name</u>, DOB))

- It is in NF-N. (also 5NF)

(2) Relation for entity type **DEPARTMENT** is

DEPARTMENT (<u>D#</u>, Dname, LOC)

- It is in NF-N (also 5NF)

(3) Relation for entity type **MANAGER** is

MANAGER (<u>MGR#</u>, Allowance)

- It is in NF-N (and 5NF)

(4) Relation for relationship set **WORK** is

WORK (<u>EMP#</u>, D#)

- It is in NF-N (and 5NF)

(5) Relation for relationship set **MANAGE** is

MANAGE (<u>D#</u>, <u>MGR#</u>)

- It is in NF-N (and 5NF)
- Both D# and MGR# are keys

(6) For the **ISA relationship**

- No relation is generated for ISA relationship
- A constraint:

MGR# (of MAMAGER) ISA
Emp# (of EMPLOYEE)

is generated instead.

The set of nested relations generated is in NF-N.

Another example:

This figure 6 is a normal form ER diagram which has 5 entity types, 3 regular relationship sets, and one ISA relationship. The identifier C# of entity type C is assigned in step 2 of Algorithm 1. The roles name AX, AY, BX, BY are assigned by step 1. Assume that the relationship set R3 has 2 functional dependencies, namely E#, D# $\rightarrow$ A# and A# $\rightarrow$ D#. Clearly the keys of R3 are {E#, D#} and {A#, E#} and we designate {E#, D#} as the identifier of R3.

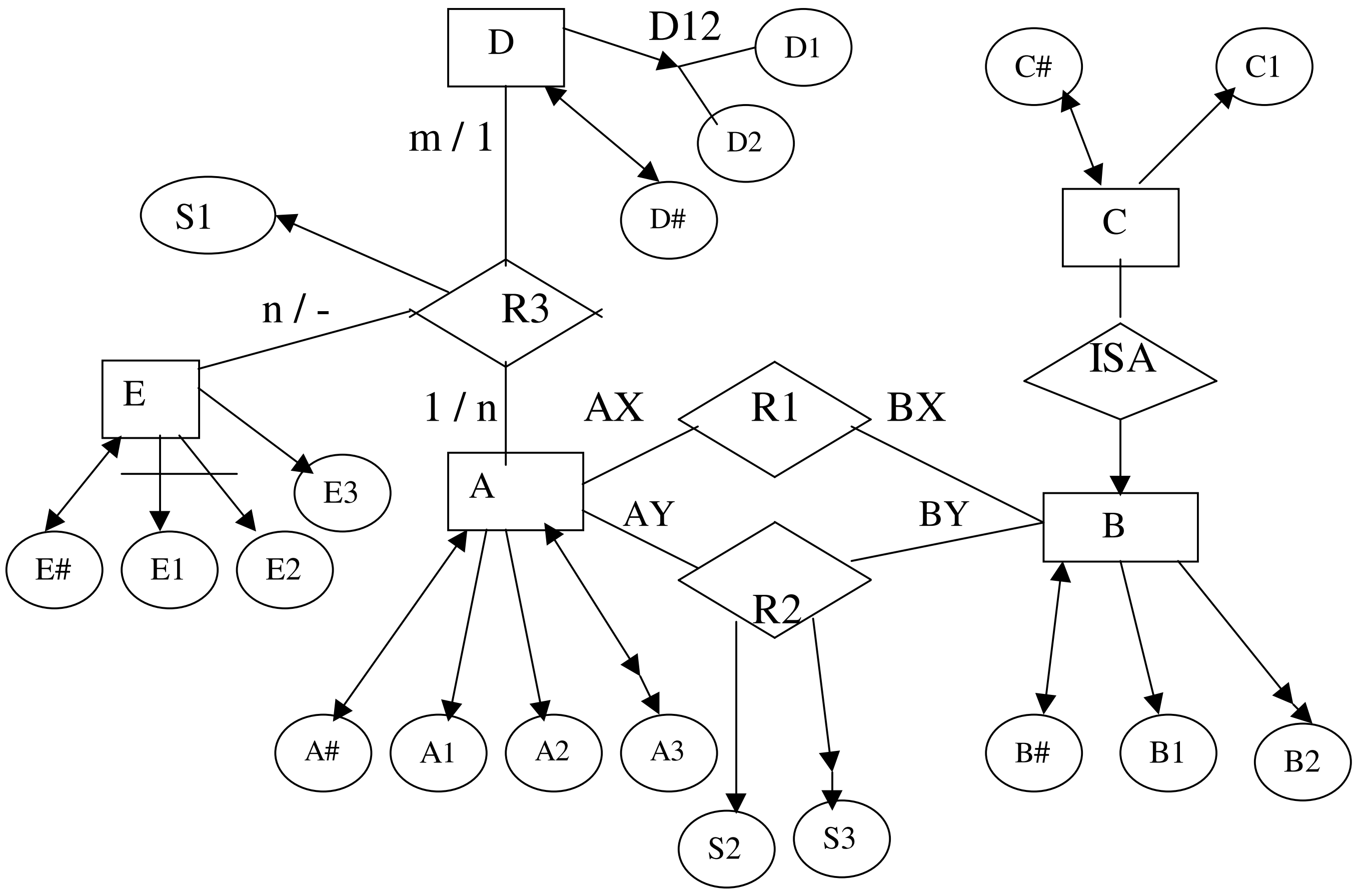

Figure 6. An ER diagram.

(1) *Relation for entity type A:*

EA ($\underline{\text{A\#}}$, A1, A2, ($\underline{\text{A3}}$))

Note that EA is in NF-N.

Constraint: A3 $\rightarrow$ A#

(2) *Relation for entity type B:*

EB ($\underline{\text{B\#}}$, B1, ($\underline{\text{B2}}$))

(3) *Relation for entity type C:*

EC ($\underline{\text{C\#}}$, C1)

Constraint: C# ISA B# (in EB)

(4) *Relation for entity type D:*

ED ($\underline{\text{D\#}}$, D1, D2)

Note that D1 and D2 are components of D12.

(5) *Relation for entity type E:*

EE ($\underline{\underline{\text{E\#}}}$, $\underline{\text{E1, E2}}$, E3)

(6) *Relation for relationship set R1:*

RR1($\underline{\text{AX, BX}}$)

Constraints: AX ISA A# (in EA)

BX ISA B# (in EB)

(7) *Relation for relationship set R2:*

RR2($\underline{\text{AY, BY}}$, S2, ($\underline{\text{S3}}$))

Constraints: AY ISA A# (in EA)

BY ISA B# (in EB)

Note that RR2 is in NF-N.

(8) Relation for relationship set R3:

RR3(A#, $\underline{\underline{\text{D\#, E\#}}}$, S1)

Constraint: A# $\rightarrow$ D#

Theorem 3.

The set of nested relations generated from a normal form ER diagram is in NF-N.

REFERENCES

[1]. P.P. Chen. The Entity-Relationship Model: Toward a Unified View of Data. *ACM Transaction on Database Systems*. 1(1), (1976).

[2]. E. F. Codd. Further Normalization of the Data Base Relational Model. In *Data Base Systems*. Randell Rustin eds, Prentice Hall (1972).

[3]. G. Jaeschke, H. J. Schek. Remarks on the Algebra of Non first Normal Form Relations. *ACM Symposium on Principles of Database Systems.* Los Angeles (1982).

[4]. Special Issue on Nested Relations. *IEEE Data Engineering.* Aug (1988).

[5]. T.W. Ling, F.W. Tompa, T. Kameda. An Improved Third Normal Formal for Relational Database. *ACM Transaction on Database System.* 6(2), (1981).

[6]. T.W. Ling. A Normal Form for Entity-Relationship Diagrams. *Proc. 4th International Conference on Entity-Relationship Approach* (1985).

[7]. T. W. Ling and L. L. Yan. NF-NR: A Practical Normal Form for Nested Relations. Journal of Systems Integration. Vol 4, 1994, pp 309-340.

[8]. A. Makinouchi. A Consideration on Normal Form of Not-Necessarily Normalized Relation in the Relational Data Model. *3rd International Conference on VLDB*. Japan (1977).

[9]. Z. M. Ozsoyoglu, L.Y. Yuan. A new Normal Form for Nested Relations. *ACM Transaction on Database Systems*. 12(1), (1987).

[10]. M. A. Roth, H. F. Korth. The Design of $\neg$1NF Relational Databases into Nested Normal Form. *ACM SIGMOD* (1987).

[11]. M. A. Roth, H. F. Korth, D. S. Batory. SQL/NF, A Query Language for $\neg$1NF Relational Databases. *Information Systems*. 12(1), (1987).

[12]. M. A. Roth, H. F. Korth, A. Siberschatz. Extended Algebra and Calculus for Nested Relational Databases. *ACM Transaction on Database Systems.* 13(4), (1988).

[13]. J. D. Ullman. Principles of Database Systems. Second edition, Computer Science Press (1982).