Schema Integration

- **Schema integration** is the activity of integrating the schemas of existing or proposed databases into a global, unified schema.

- **Schema integration** occurs in two contents:

(1) View Integration (in database design)

- Its goal is to produce an integrated schema starting from several application views that have been produced independently.

- Reasons for such integration

(a) The structure of the database for large applications is too complex to be modeled by a single designer in a single view.

(b) User groups typically operate independently in organization and have their own requirements and expectations of data, which may conflicts with other user groups.

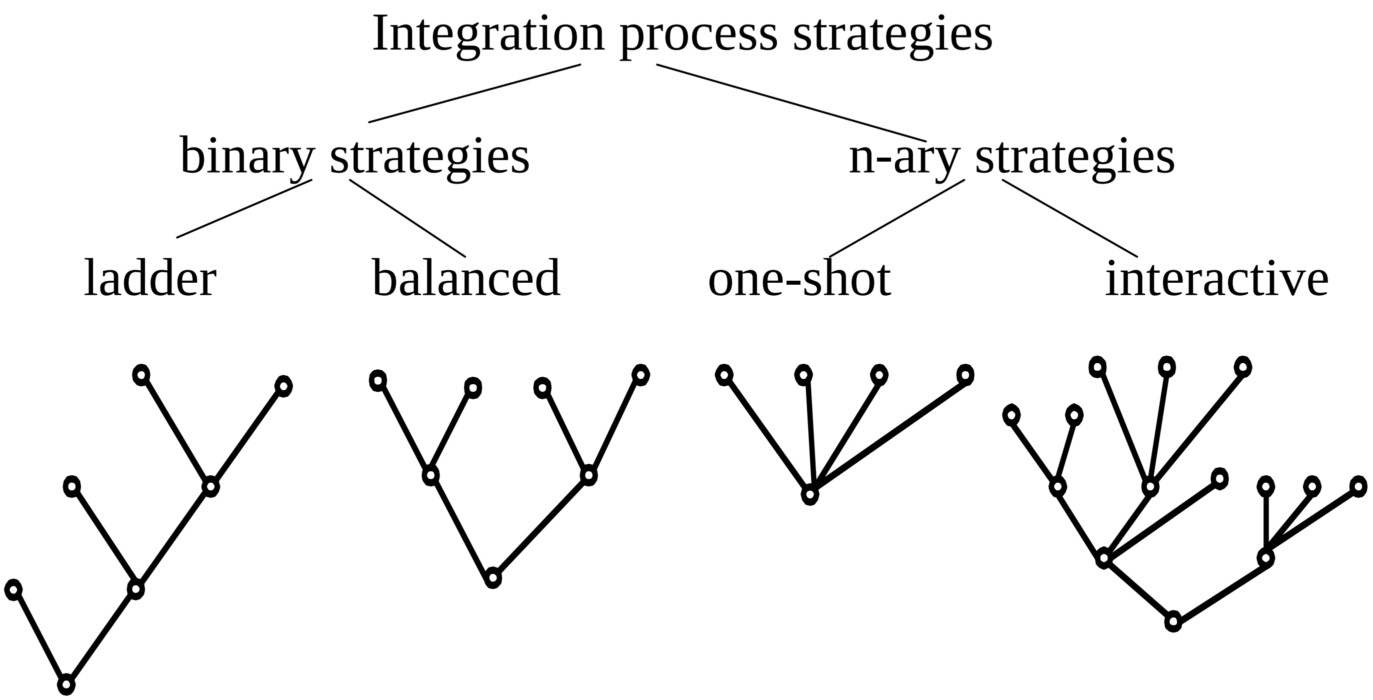
Ref: Batini, …, A comparative Analysis of
Methodologies for Database Schema Integration
ACM computing survey 1986

(2) **Database integration** (in distributed database management)

- A distributed database is a collection of data that logically belong to the same system but are spread over the sites of a computer network.

- Database integration produces a global schema of a collection of database. This global schema is a virtual view of all database taken together in a distributed database management.

Integration processing strategies



Integration process strategies

binary strategies

n-ary strategies

ladder

balanced

one-shot

interactive

Steps and goals of the integration process

(1) Preintegration.

- Choose integration processing strategies
- This governs the choice of schemas to be integrated

(2) Comparison of the schemas.

- Schemas are analyzed and compared to determine the correspondences among concepts and detect possible conflicts.

(3) Conforming the schemas.

- Once conflicts are detected, an effort is made to resolve them so that the merging of various schemas is possible.
- Automatic conflict resolution is generally not feasible; interaction with designers is required.

(4) Merging and Restructuring.

- The schemas are ready to be superimposed, giving rise to some intermediate integrated schema(s).

- The intermediate results are analyzed, and, if necessary, restructuring in order to achieve the criteria:

(a) completeness and correctness

- the integrated schema must contain all concepts appear in any component schema correctly.

(b) minimality

- no redundancy in the integrated schema

(c) understandability

- the integrated schema should be easy to understand for the designer and the end users.

Assumptions for integration:

- all views (schemas) are correct but different views of the global database

- some views may have less information than others

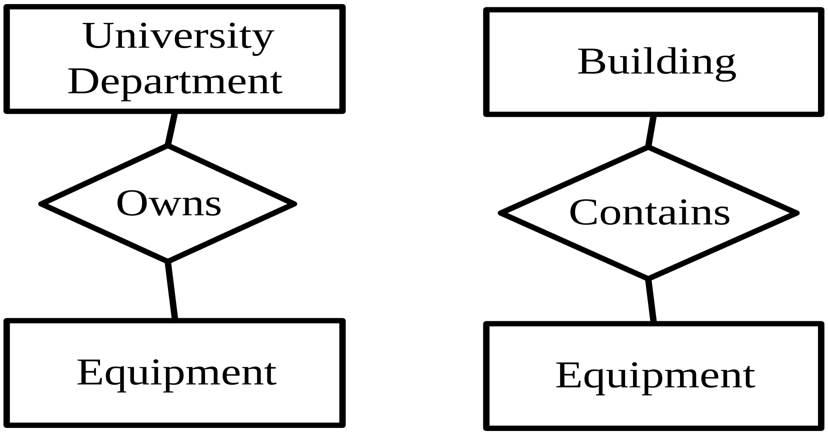
Comparison of schemas & Conforming the schemas

- Check all conflicts in the representation of the same objects in different schemas.

(1) **Naming Conflicts.** There are 2 types

(a) **Homonyms**: the same name is used for two different concepts (objects).

e.g.



The same “Equipment” in the two schemas refer to different things. The first Equipment refers to computer, copiers, etc. and the second Equipment refers to airconds, furniture, etc.

e.g. Name in EMPLOYEE entity is different from the Name in a part entity.

(b) **Synonyms.**

The same concept is described by different names.

e.g. SNO and S# (attribute)
EMP and EMPLOYEE (entity)
TAKE and ENROL (relationship)

(2) **Data type conflicts**

The same attribute with different data types

e.g. integer & real ⇒ integer
e.g. char (20) & char (30) ⇒ char(20)
e.g. different range values
positive integer & [1,100] ⇒ [1,100]

(3) Generalization & Specialization

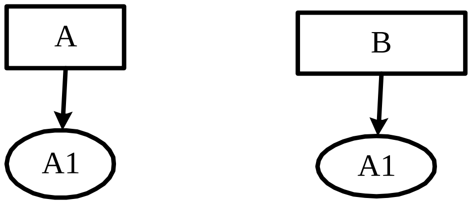
e.g. Part and REDPART (in two schemas represented as
REDPART ISA PART

e.g. EMPLOYEE and MANAGER represented as
MANAGER ISA EMPLOYEE

e.g. WOMAN and PERSON

note: If an attribute is the attribute of two entity types related by ISA relationship in 2 views, then the data type of the attribute of the substype (subclass) entity type can be more restricted than the data type of the attribute of its supertype entity type.

e.g.

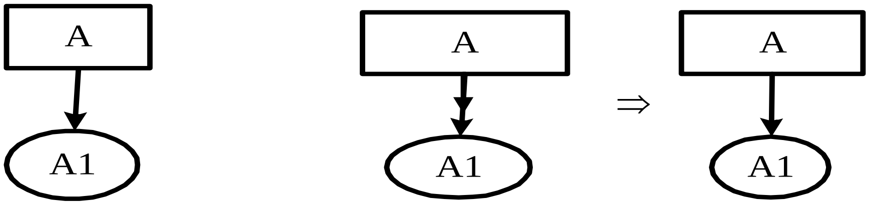


A ISA B
A1 in A is integer & A1 in B is real (no conflict).

(4) Cardinality Conflicts of attributes

- different cardinalities for the same attributes of an entity type (or relationship type) in 2 views.

e.g.



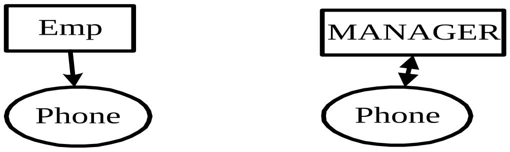
Similarly 1:1 & m :1 attribute ⇒ 1:1
m: 1 & m : m attribute ⇒ m :1

Note.

(1) The second schema contains less information than the first one.

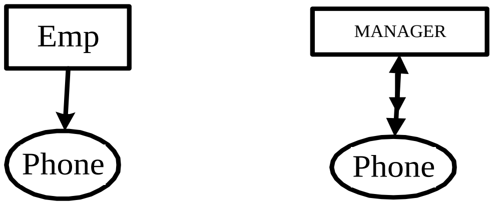
(2) Involving ISA relationship.

(a)



MANAGER ISA EMP (No conflict)

(b)

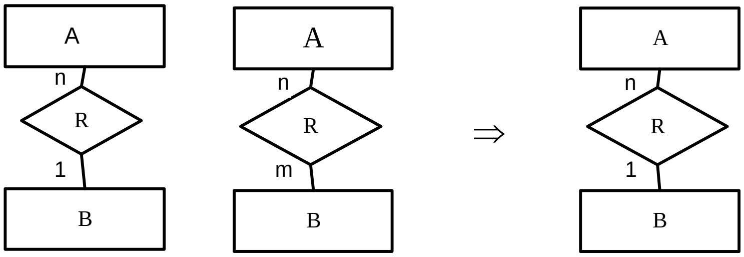


and MANAGER ↛ Phone

Conflict, wrong design for view 1

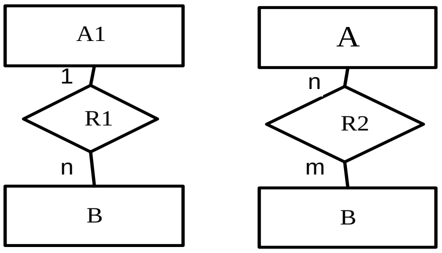
(5) Cardinality conflicts of entity types in relationship types

e.g.



Note: This is similar to $\{A \rightarrow B\} \cup \{A \twoheadrightarrow B\}=\{A \rightarrow B\}$

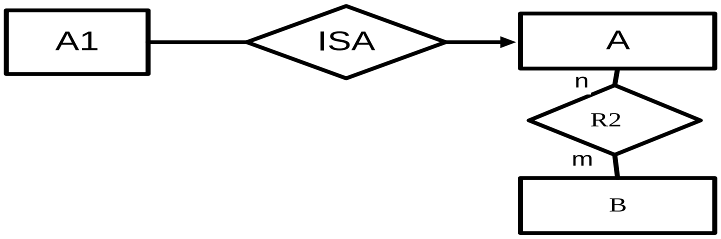
e.g. involving ISA relationship



and A1 ISA A

R1 ⊆ R2

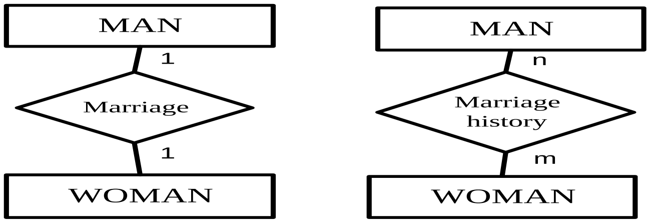
⇒



Also B → A1 in R2

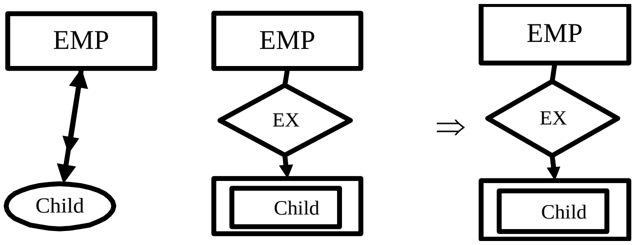
R1 is obtained by restricting A in R2 by A1.

Note. No conflict, the two relationship types are not the same.

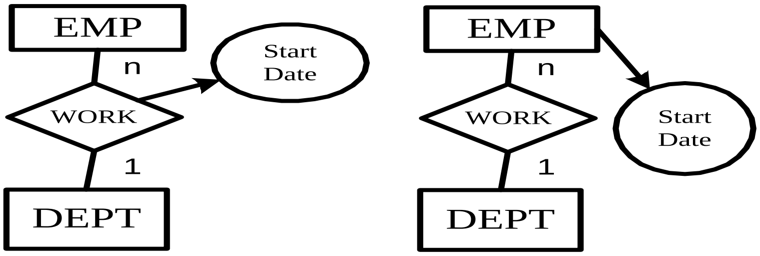


(6) **Structure conflicts** the same concept is represented by different (level) constructs in different schemas.

e.g. a class of objects (e.g. Child) is represented as an entity type in one schema and as an attribute in another schema.



e.g. an attribute (e.g. start_date) is represented as an attribute of a relationship set in one schema and as an attribute of an entity type in another schema.

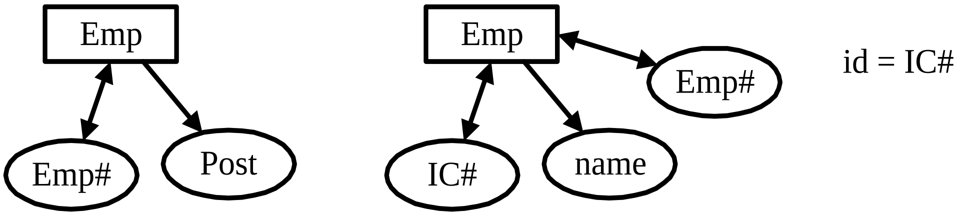


which one is correct? Depends! Ask the designer.
More detail from pages 20 to 36.

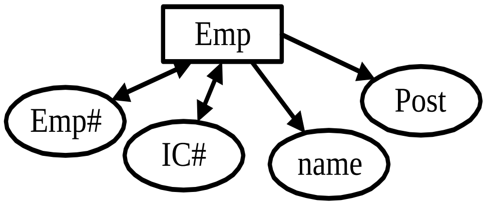
(7) Identifier (Primary Key) Conflicts

- Different keys are assigned as identifiers of the same entity type in different schemas.

e.g.



⇒



which key (EMP# or IC#) is the identifier?
Ask the designer.

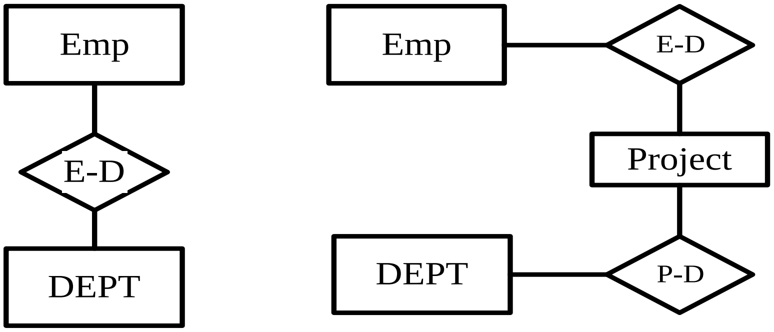
Merging and Restructuring of Schemas

- superimpose schemas (completeness & correctness)
- remove redundancies (minimality)
- restructure the intermediate integrated schema to achieve understandability.

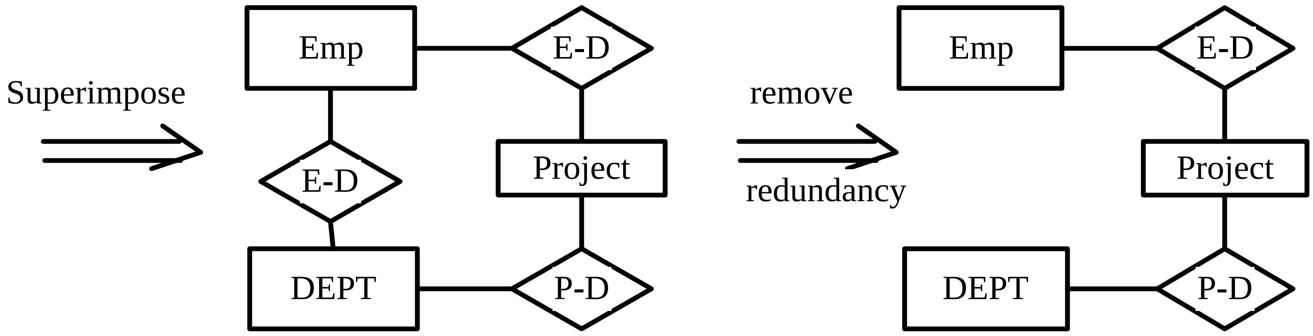
(1) Join relationships

a relationship type in one schema is the join of other two (or more) relationship types in another schema.

e.g.



$$E\text{-}D = E\text{-}P \bowtie P\text{-}D[Emp, Dept]$$

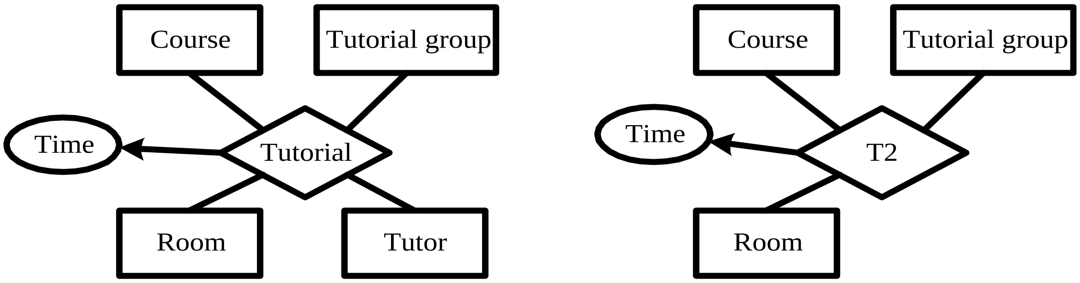


E-D becomes a view.

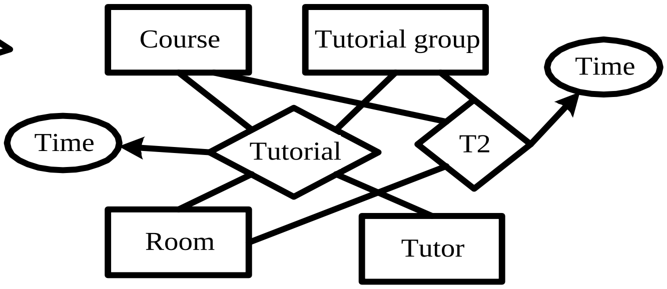
(2) Project Relationships

- a relationship type in one schema is the projection of another relationship type in another schema

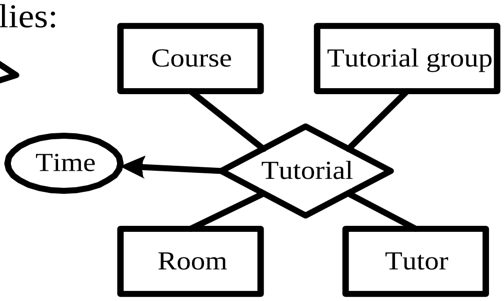
e.g.



Superimpose



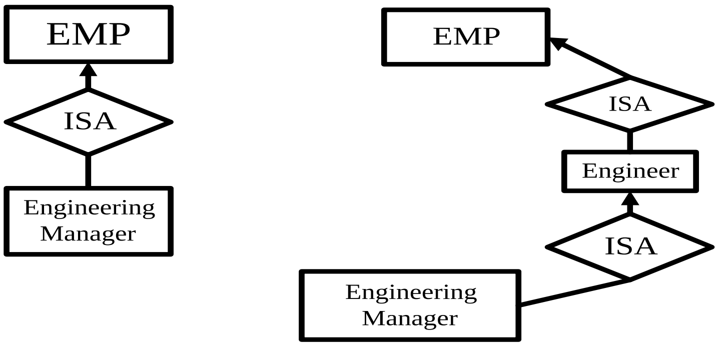
Designer replies:



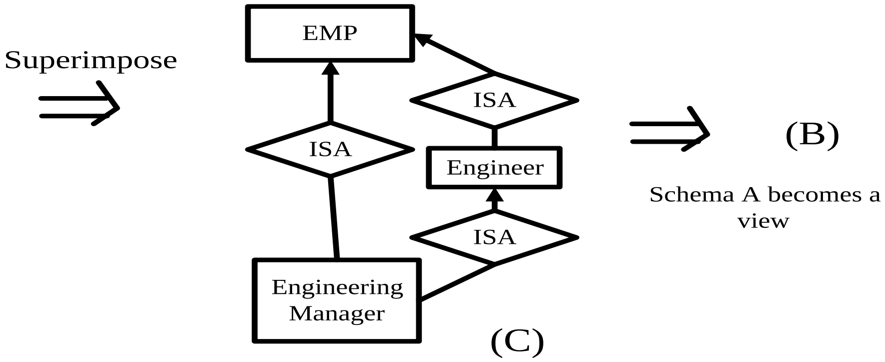
T2 becomes a view. T2=Tutorial[C,G,R,T]

(3) Subtype (ISA) redundancies

(a) A ISA B & B ISA C $\Rightarrow$ A ISA C
e.g.

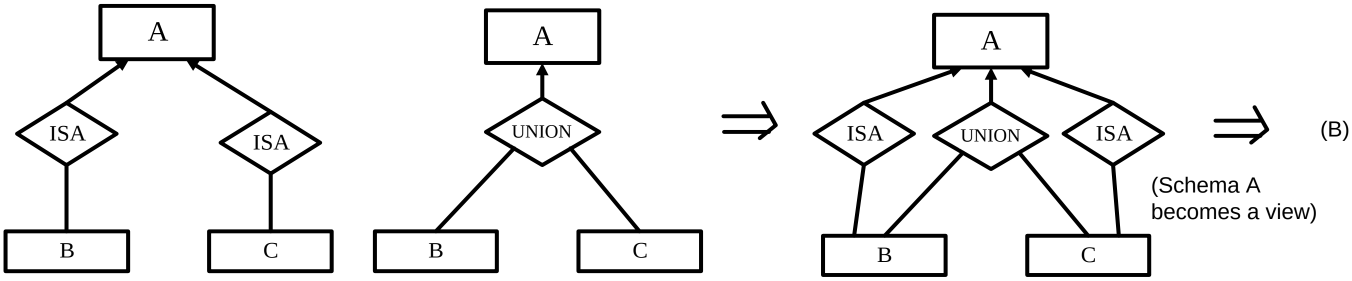


(A) (B)



(C)

(b) A = UNION(B, C) $\Rightarrow$ B ISA A and C ISA A

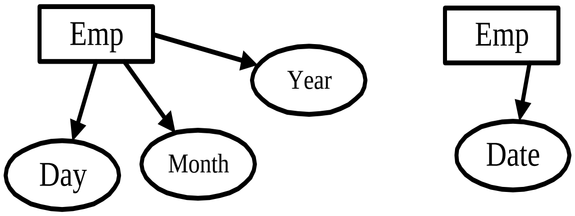


(A) (B) (C)

(c) Similarly for other set operation relationships.

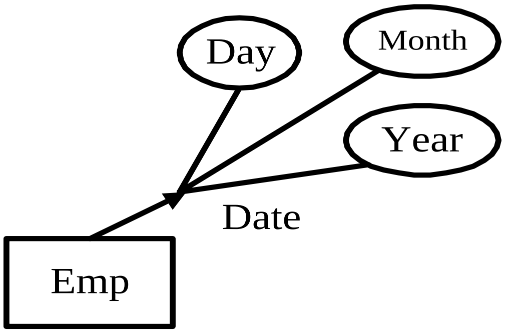
(4) Simple and Composite attributes

(a)



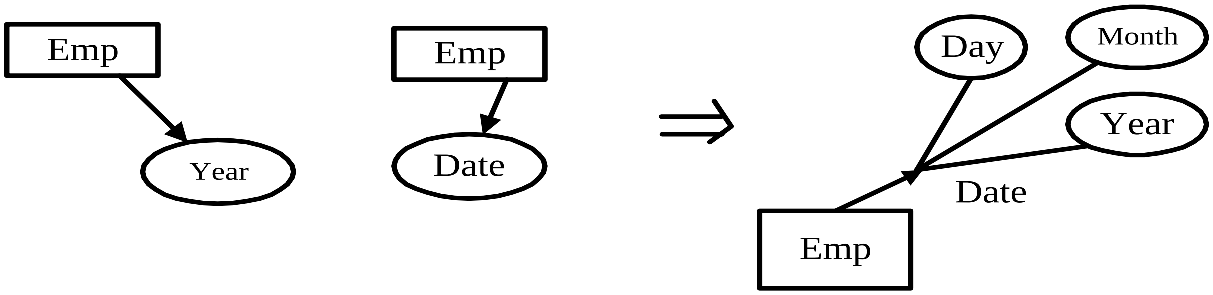
Date is represented as 3 attributes day, month, year in another schema.

⇒

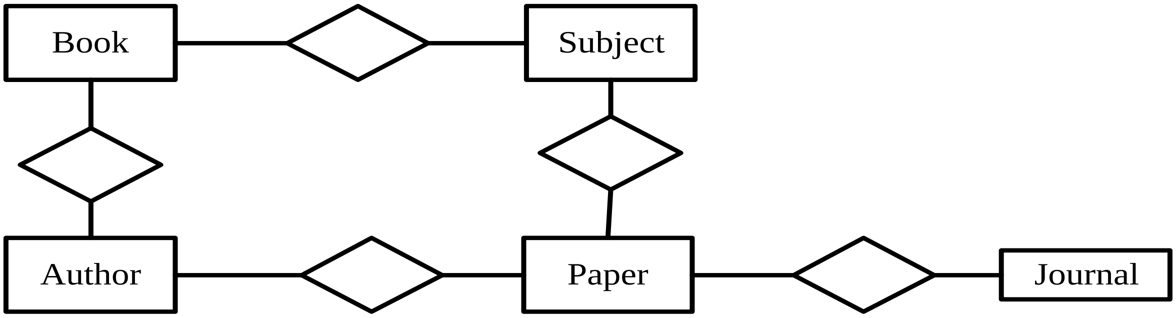


(b) A simple attribute in one schema is a component of an attribute in another schema

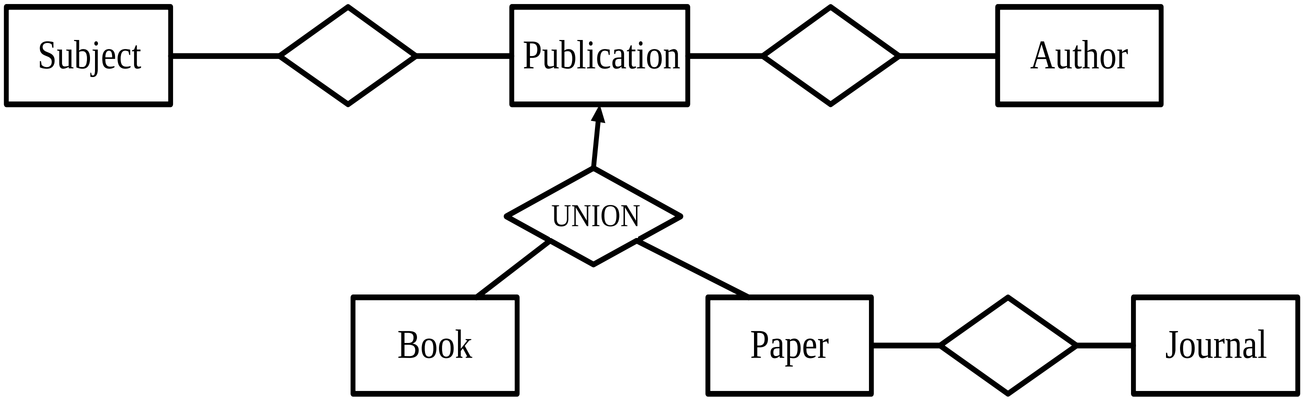
e.g.



(5) Improving understandability



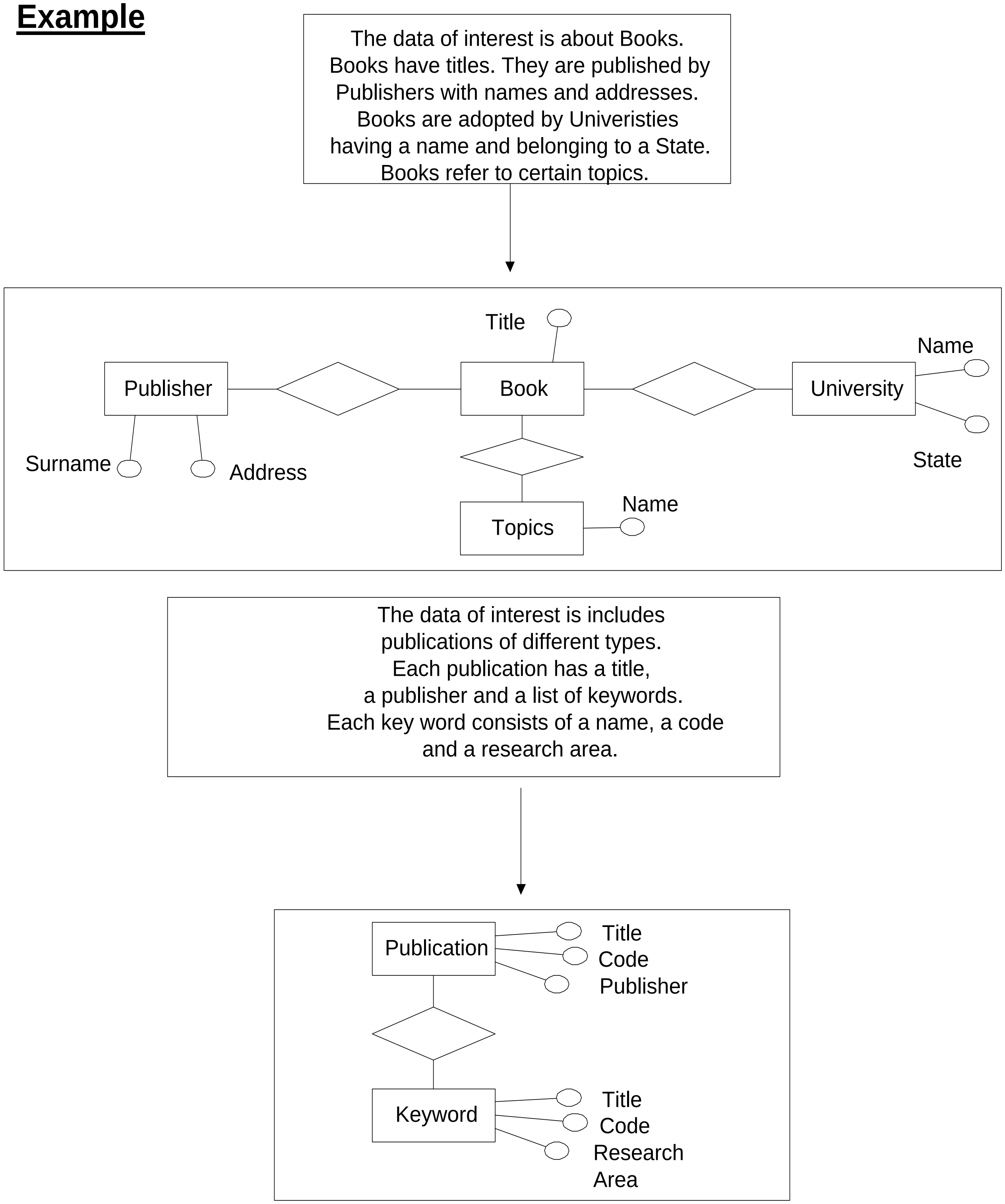
(A)



(B)

- Schema (B) is name readable than schema (A)
- difficult to do the transformation automatically

Example



The data of interest is about Books.
Books have titles. They are published by
Publishers with names and addresses.
Books are adopted by Univeristies
having a name and belonging to a State.
Books refer to certain topics.

The data of interest is includes
publications of different types.
Each publication has a title,
a publisher and a list of keywords.
Each key word consists of a name, a code
and a research area.

Figure 3. Examples of requirements and corresponding schemas

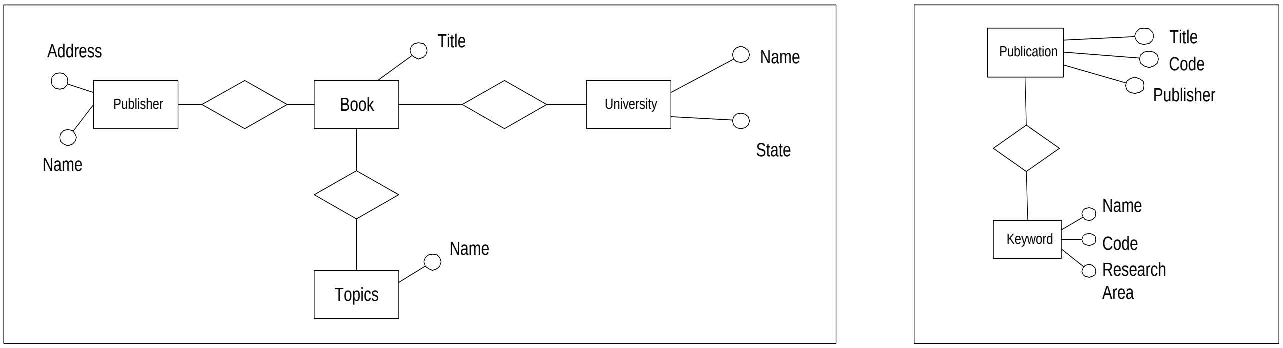


Figure 4a. Original Schemas

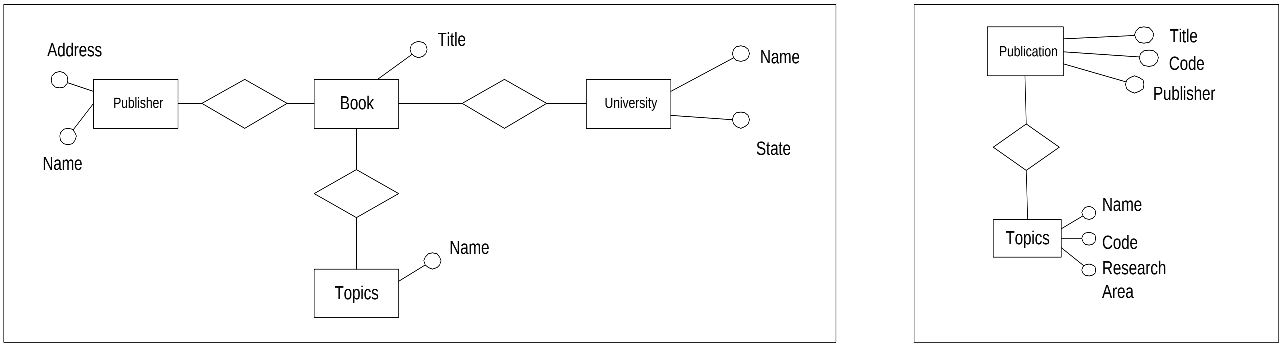


Figure 4b. Choose "Topics" for "Keyword (Schema 2)

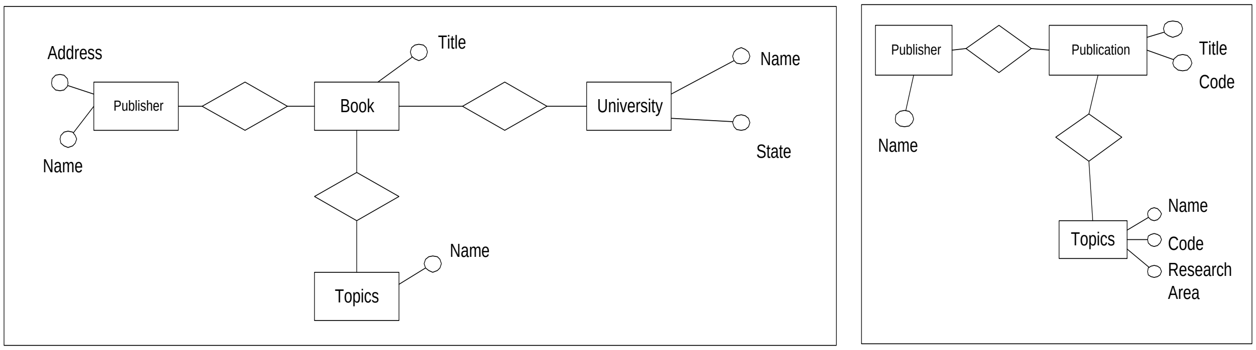


Figure 4c. Make Publisher into an entity (Schema 2)

Figure 4d: Superimposition of schemas

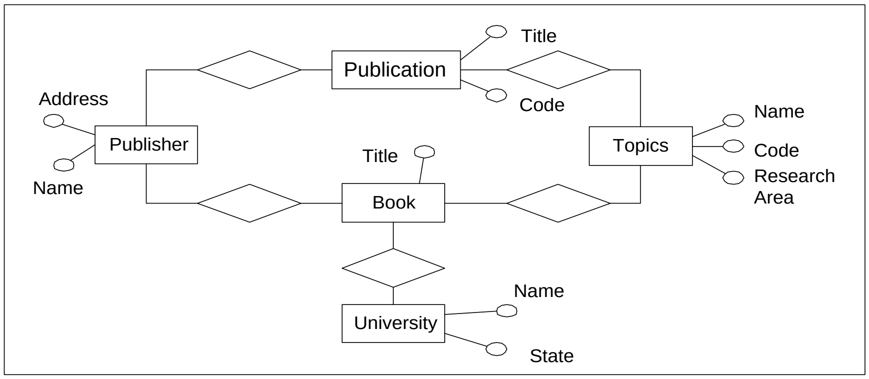


Figure 4e: Creation of a subset relationship.

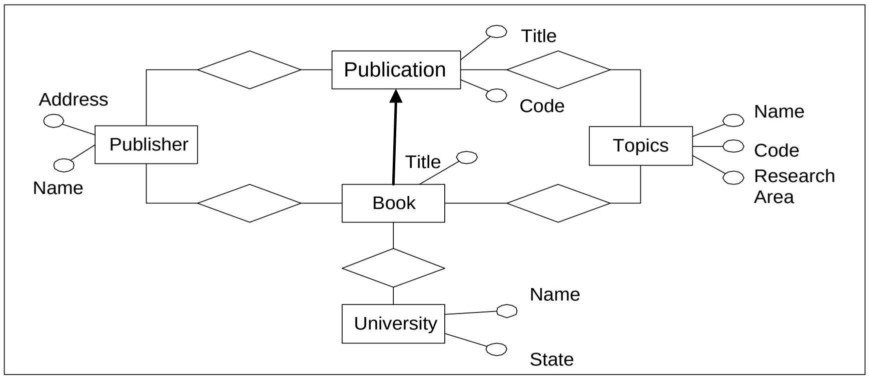


Figure 4f: Drop the properties of Book common to Publication.

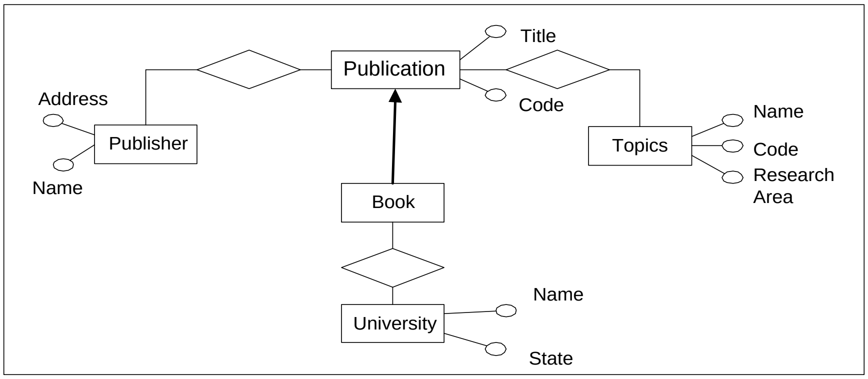


Figure 4. An example of Integration

Structure Conflicts

(Lee and Ling, OOER'95 Proceedings)

Previous works [Bati84, Spac92] suggest four types of structural conflicts:

1. Entity type in one schema modeled as an attribute in another schema.

2. Entity type in one schema modeled as a relationship set in another schema.

3. Relationship set in one schema modeled as an attribute in another schema.

4. Attribute of a relationship set modeled as an attribute of an entity type.

However, we advocate that

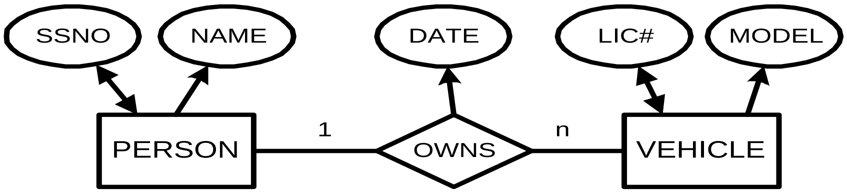
1. Only first type of conflict is meaningful.

2. Rest of conflicts are automatically resolved after we have resolved the first type of conflict.

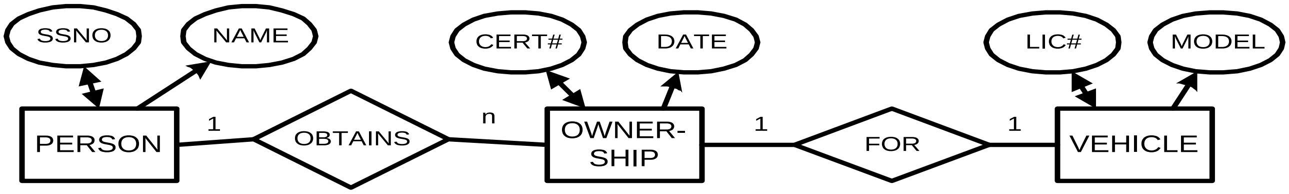
3. Algorithm transforms an attribute into an equivalent entity type with no loss of semantics or functional dependencies.

PREVIOUS WORKS & PROBLEMS

Example 1: Integrate S3 and S4



Schema S3

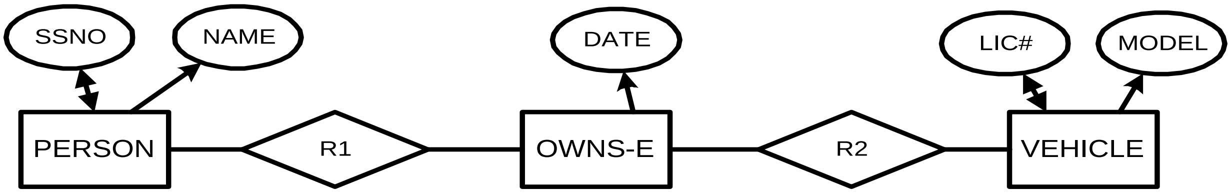


Schema S4

* [Bati84]'s Approach:

1. Reconcile **relationship set** OWNS in S3 with **entity type** OWNERSHIP in S4.

2. Transform relationship set OWNS **into entity type** OWNS-E connected to PERSON and VEHICLE in S3 by relationship sets R1 and R2.



Schema S5: Relationship set OWNS in S3
transformed to entity type OWNS-E by [Bati84]

3. Integrate S4 with S5. Final schema is S4.

*** Problems:**

(1) Semantics of R1 and R2 ?

(2) Cardinalities of participating entity types of R1 and R2 ?

(3) Merge S4 and S5 because they are structurally identical ?

- How do we know S5.R1 ≡ S4.OBTAINS &
  S5.R2 ≡ S4.FOR ?

- We can have more than one relationship sets with different meanings between same entity types.

(4) Identifiers of OWNS-E ?

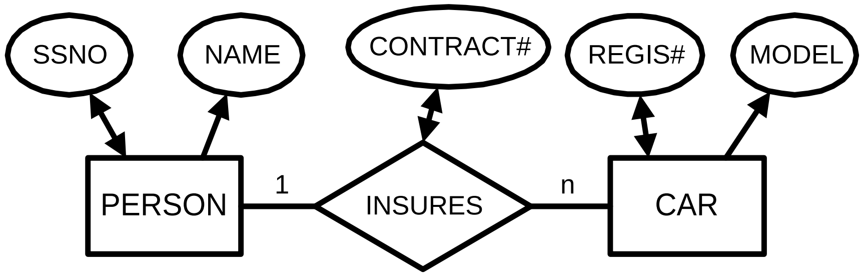
- Even if OWNS-E has an identifier, say O#, how do we know S5.O# ≡ S4.CERT# ?

(5) How to populate R1 and R2 from OWNS?

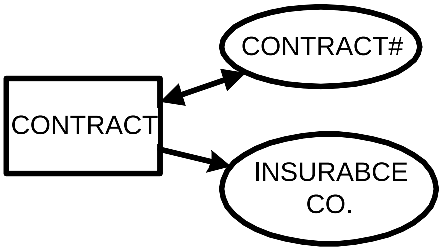
(6) Loss of information if split a relationship set into two or more relationship sets.

- If there exist non-trivial FDs which involve more than 2 attributes, then these FDs may not be preserved when we split the n-ary relationship sets into n binary relationship sets.

Example 2: Integrate S11 and S12



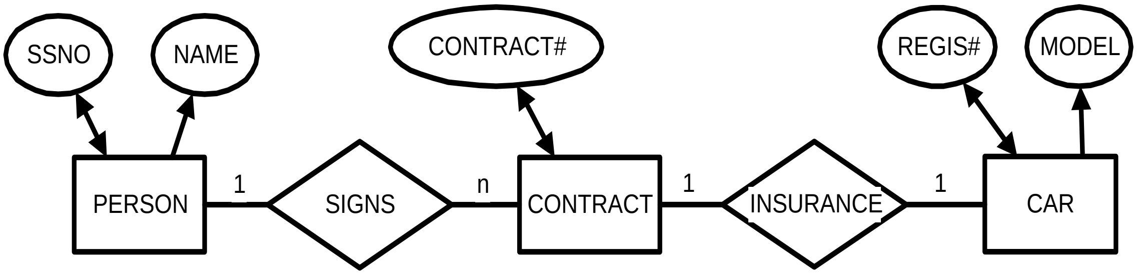
Schema S11



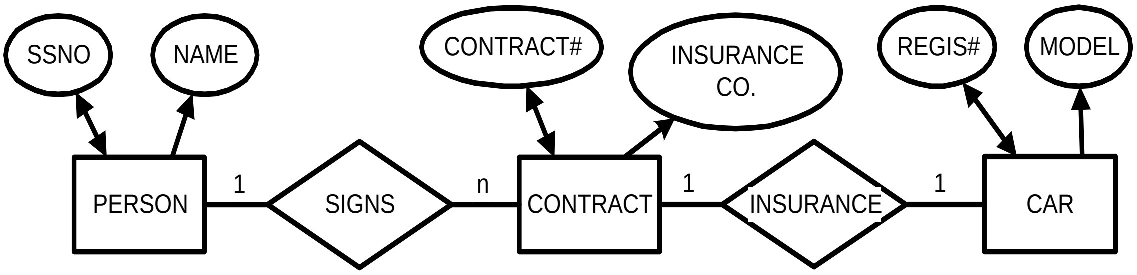
Schema S12

- **[Spac92]'s approach:**

1. Relationship S11.INSURES equivalent to entity type S12.CONTRACT.
2. Transform relationship set S11.INSURES into an entity type CONTRACT with identifier CONTRACT# and binary relationship sets SIGNS and OWNS.
3. Integrate S13 with S12. Final schema is S14.



Schema S13 transformed from S11 by [Spac 92]



Integrated Schema obtained by [Spac92]

* Problems:

(1) Loss of semantics.

- S11. INSURES associates a person who insures a car with a particular contract.
- In S13, this association is split into two relationships: SIGNS associates a person signs a contract, INSURANCE associates an insurance contract with a car.
- Possible to have a contract for a car without having the contract signed by a person.

(2) R(A, B, C) is the lossless join of its projections R1(A, B) & R2(A, C) iff A $\twoheadrightarrow$ B | C holds in R(A, B, C).

INSURES (SSNO, $\underline{\text{REGIS\#}}$, $\underline{\text{CONTRACT\#}}$) split into:
SIGNS (SSNO, $\underline{\text{CONTRACT\#}}$) &
INSURANCE ($\underline{\text{CONTRACT\#}}$, $\underline{\text{RESIGS\#}}$)
Since CONTRACT# $\twoheadrightarrow$ {SSNO, REGIS#}

If we change the cardinality of CONTRACT# or the cardinalities of participating entity types in INSURES, then the MVD may not hold.

OUR APPROACH

* Schema Integration Algorithm

Input: Schemas S1 and S2, and integration assertions IA.
Output: Integrated schema Sm.

1. Resolve structural conflicts
   - Transformation of an attribute into an entity type.

2. Merge schemas.

3. Remove redundant relationship sets.

4. Remove inherited and derived attributes.

5. Create ISA hierarchies.

Transformation of an Attribute into an Entity Type

***** Consider the following factors:

(1) Attribute belongs to entity type or relationship set

(2) Attribute is part of an identifier, a key or a composite attribute.

(3) Cardinality of attribute

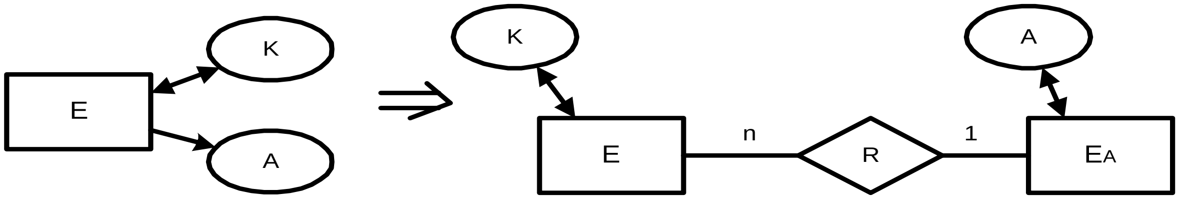
(4) Determine the cardinality of the new entity type (transformed from the attribute) in the relationship set it participates in.

*** Attribute A belongs to an entity type E**

Four basic scenarios when we transform A to an entity type EA with identifier A:

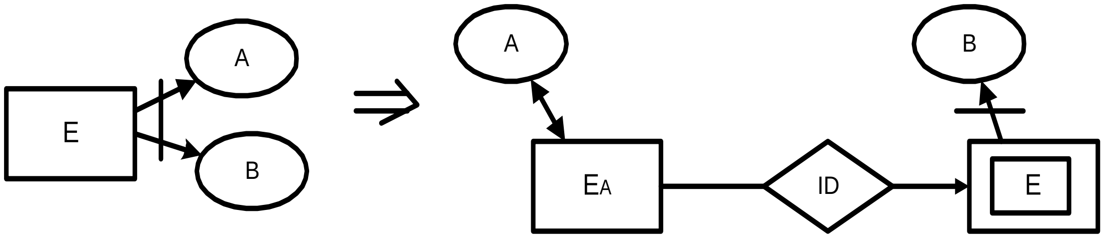
Case 1: A is not part of a key and not part of a composite attribute.

EA connects to E by new relationship set R.



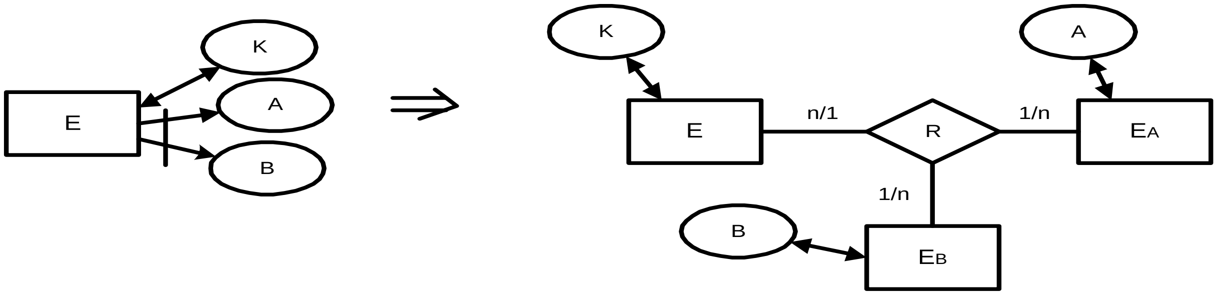
Case 2: A is part of the identifier of E and there is no other key.

E becomes a weak entity type which is identifier dependent on EA.

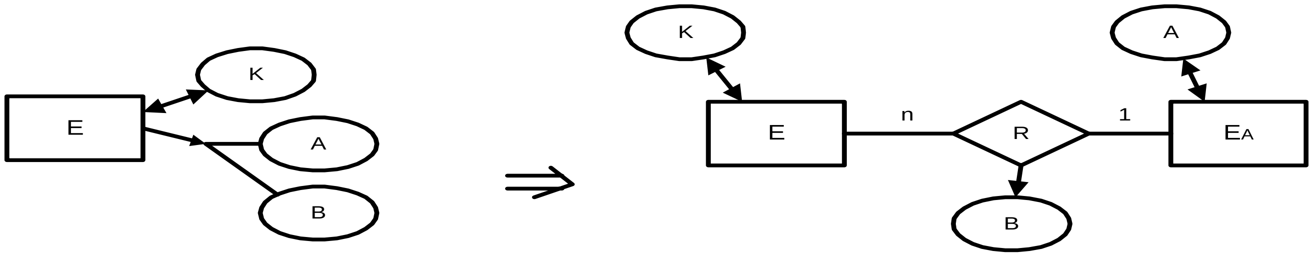


Case 3: A ∪ B is a key of E and there is another key, or
A ∪ B is a composite multivalued attribute.

Transform B to entity type EB.
EA and EB connect to E by a new relationship set R.



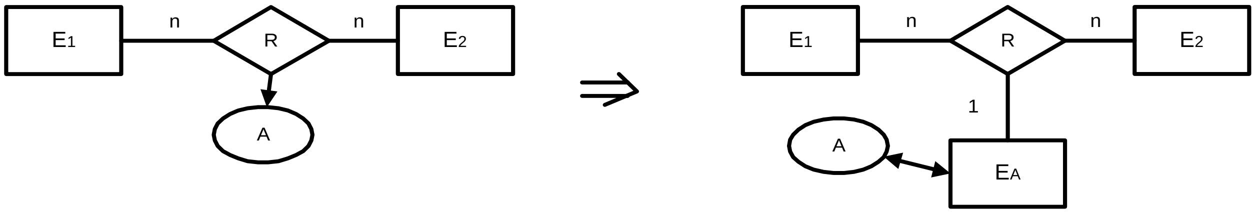
Case 4: A ∪ B is a composite m-1 attribute.
EA connect to E by new relationship set R.
B becomes a m-1 attribute of R.



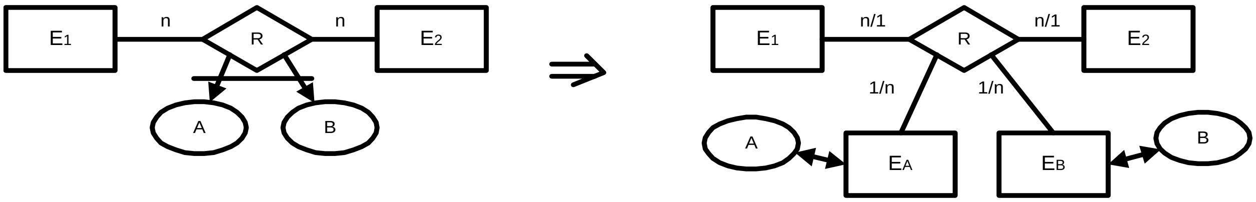
*** Attribute A belongs to a relationship set R.**

Three basic scenarios when we transform A to an entity E_A with identifier A:

Case 1: A is not part of composite attribute of R.
E_A becomes a participating entity type of R.



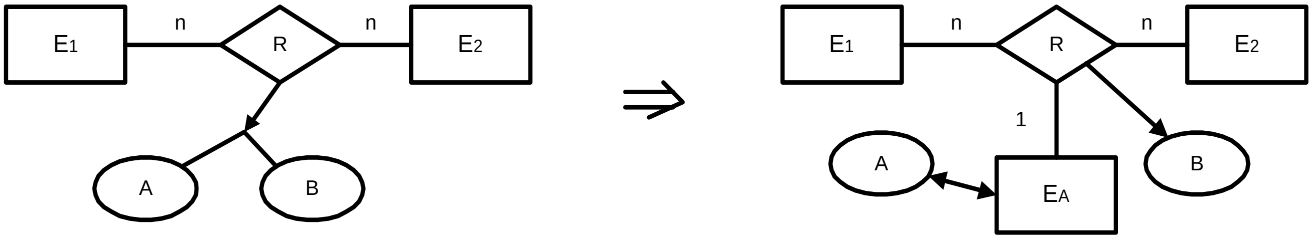
Case 2: A ∪ B is a composite multivalued or 1-1 attribute of R, B a set of attributes.
Transform B to an entity type E_B.
E_A and E_B become participating entity types of R.



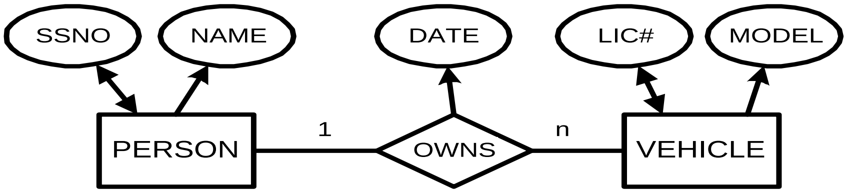
Case 3: A ∪ B is a composite m-1 attribute of R, B is set of attributes.

EA becomes a participating entity type of R.

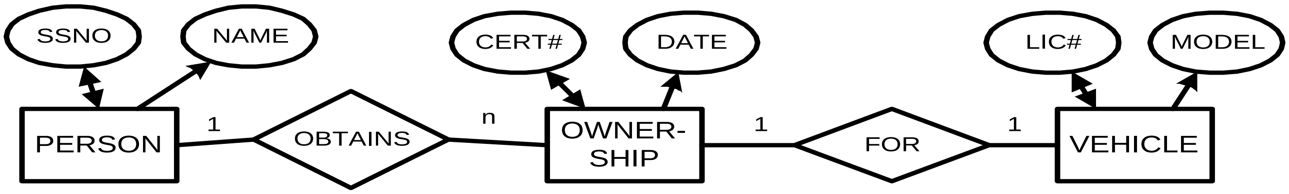
B becomes a m-1 attribute of R.



Example 1: Integrate S3 and S4



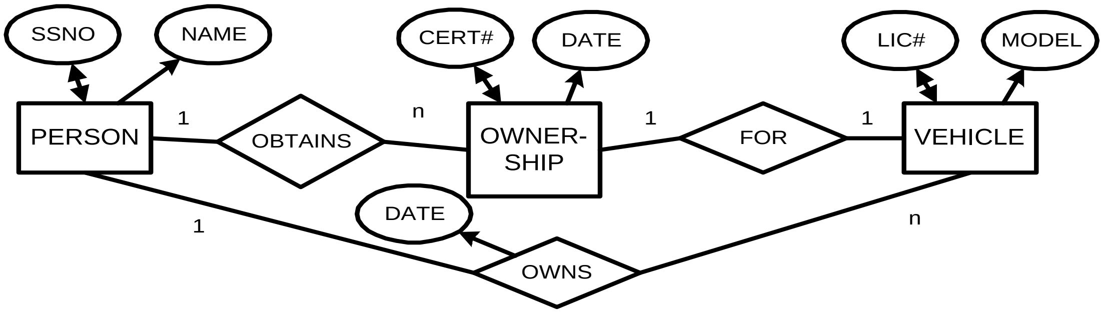
Schema S3



Schema S4

- **Our Approach:**

1. No structural conflict.
2. Merge S3 and S4.



Schema S6: Schema obtained by merging S3 and S4 by our approach

3. Check for cycles.

```
Is OWNS a derived relationship set ?
IF  yes   Then     OWNS is redundant.
                   DATE is a derived attribute
                   Remove from schema
                   Integrated schema is S4.
ELSE      S3.DATE & S4.DATE are homonyms.
          Integrated schema is S6.
```

* We obtain the same integrated schema as [Bati84], but our approach does not have the problems faced by [Bati84].

Attribute of Relationship Set vs Attribute of Entity Type

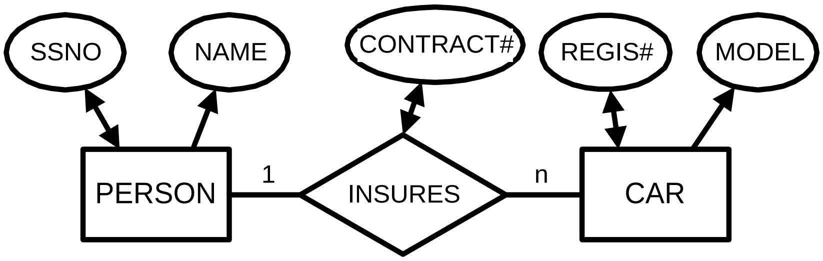
- Semantics of these two types of attributes inherently different.

- Attribute of entity type.
  - Property of the entity type
  - Not related to other entity types or relationship sets
  - SSNO and NAME are properties of PERSON

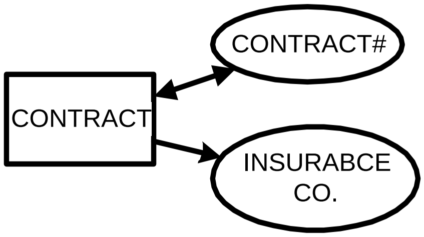
- Attribute of relationship set.
  - Meaningful only when associated with all the participating entity types in the relationship set.

  - S3.DATE:
    a. Models the date on which a person owns a particular vehicle.
    b. Meaningful only when associated with SSNO and LIC# together.
    c. Value of DATE need to be updated whenever value of LIC# is updated.

Example 2: Integrate S11 and S12



Schema S11



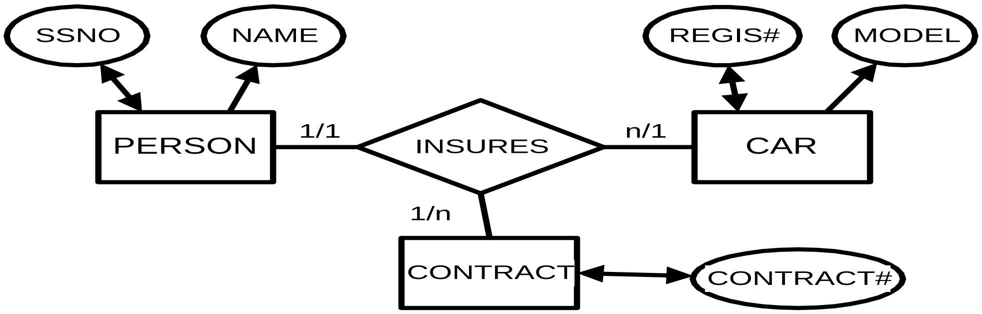
Schema S12

- **Our Approach:**

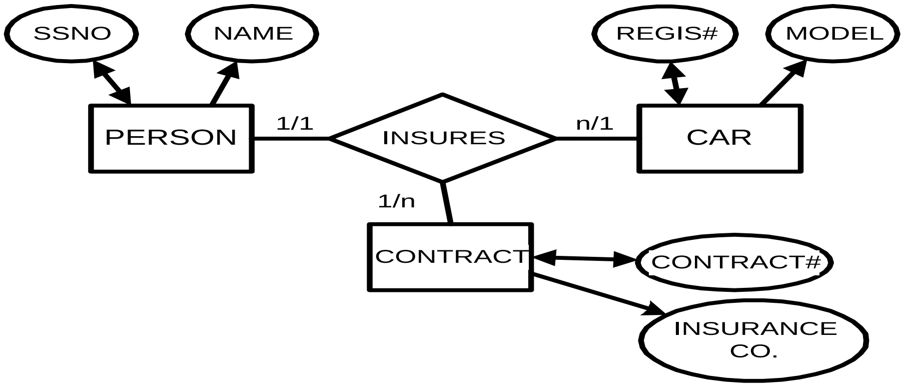
1. Structural conflict:
   S11.CONTRACT# ≡ S12.CONTRACT.

   Transform the attribute S11.CONTRACT# into an entity type CONTRACT which becomes a participating entity type in INSURES.

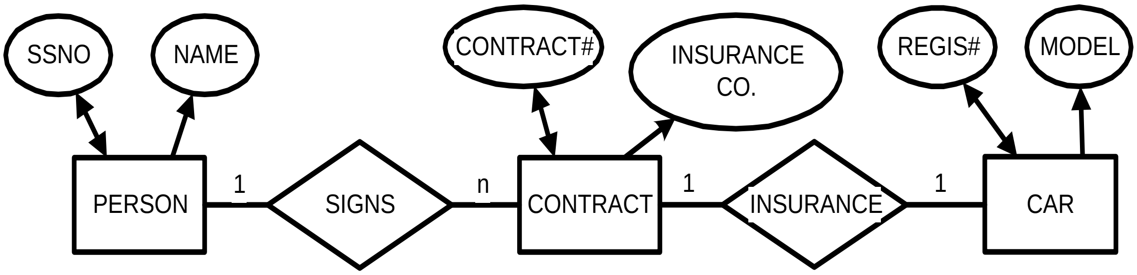
2. Merge S12 and S15. Final schema is S16.



Schema S15 obtained by transforming the attribute
CONTRACT# in S11 into an enity type CONTRACT#
by our approach.



Integrated schema S16 obtained by our approach.



Integrated Schema obtained by [Spac92]

- No loss of semantics because ternary relationship set INSURES still maintains the association of a person who insures a car with a particular contract.

CONCLUSION

1. Different approach to integrate ER schemas.

2. Resolve only structural conflict between an entity type and an attribute, and the other structural conflicts are automatically resolved.

3. Algorithm transform attribute into equivalent entity type without loss of semantics or FDs.

References

[Bati84] Batini, C. and Lenzerini, M., A Methodology for Data Schema Integration in the Entity-Relationship Model, IEEE Trans. Software Engineering, Vol 10, 6, 1984.

[Spac92] Spaccapietra, S., Parent, C., and Dupont, Y., Model independent assertions for integration of heterogeneous schemas, VLDB Journal, (1), 1992.

[Lee&Ling95] Lee M L and Ling T W, Resolving Structural Conflicts in the Integration of Entity Relationship Schemas, 14th Int Conf on Object-Oriented Entity Relationship Modeling, Gold Coast, Australia, Dec 12-15, 1995

? Find rules for doing schema integration for relational, network, and hierarchical models.