NORMALIZATION

PREPARED BY: LUIS MEING

OBJECTIVES:

01 **Definition**

02 **Rules**

2.1: First Normal Form
2.2: Second Normal Form
2.3: Third Normal Form

03 **Conclusion**

Definition

- Database Normalization is a technique of organizing the data in the database

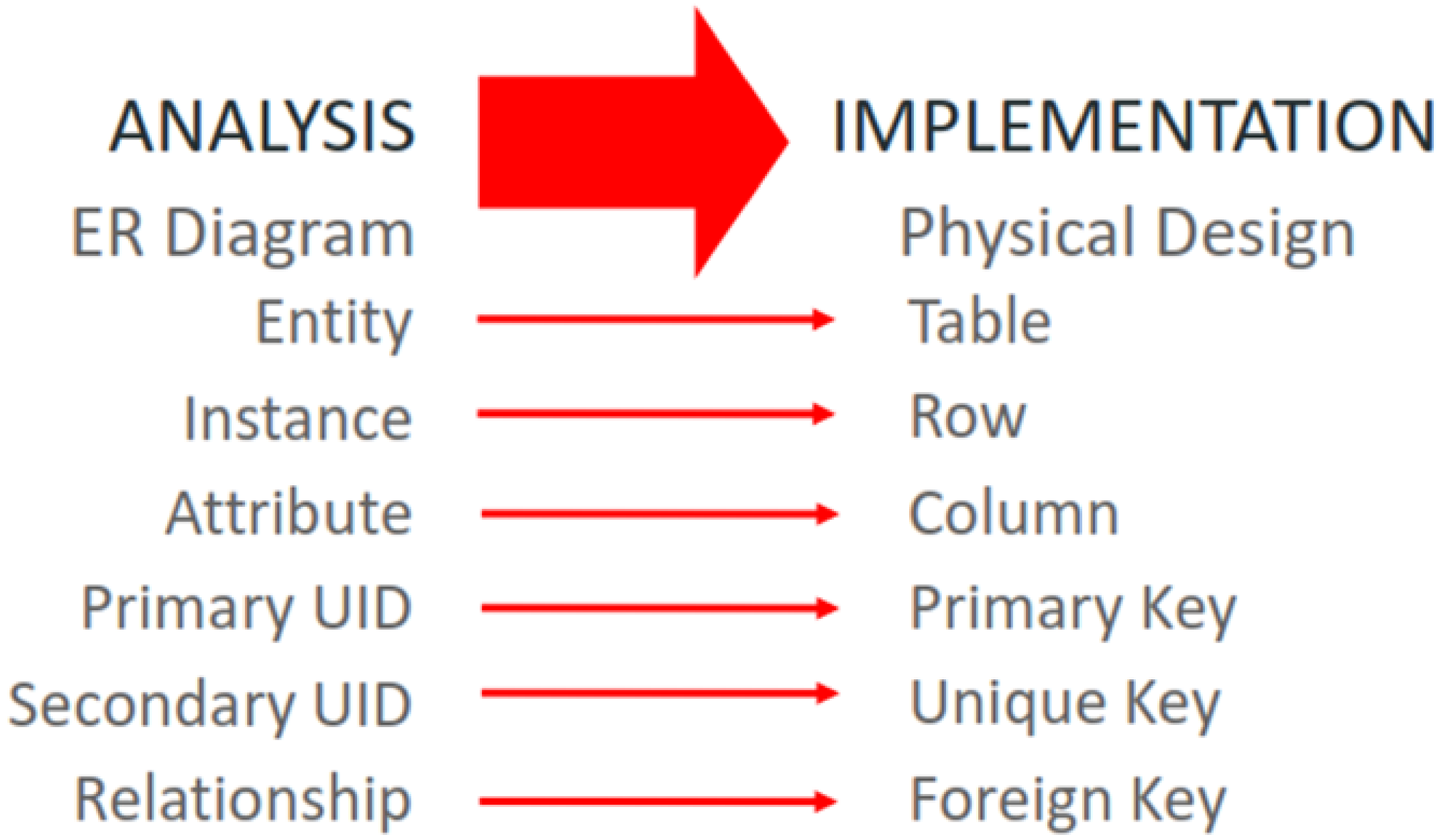
Definition

- is a systematic approach of decomposing tables to eliminate data redundancy(repetition) and undesirable characteristics like Insertion, Update and Deletion Anomalies

Definition

- It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.
- refining the conceptual to transform into physical models

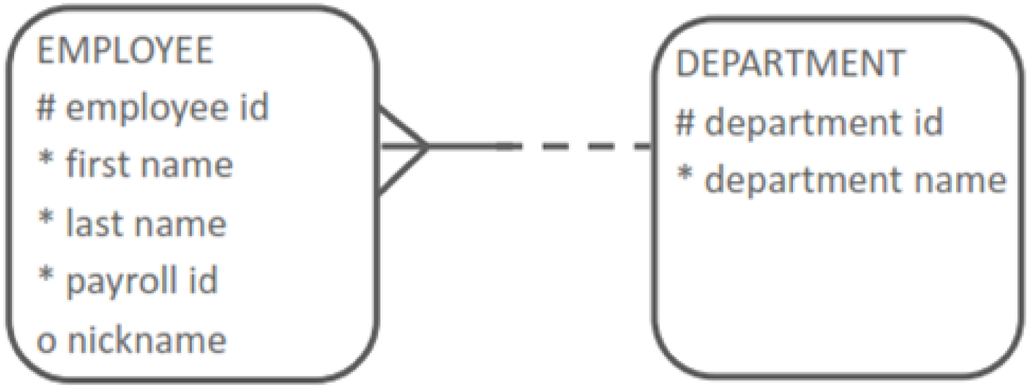
Terminology Mapping



1.1 Definition and Terms

Transforming Conceptual To Physical

Conceptual Model (ERD)



Transformation process

Physical Implementation: Relational Database

EMPLOYEES (EPE)		
Key type	Optionality	Column name
pk	*	employee_id
uk	*	payroll_id
	*	last_name
	*	first_name
	o	nickname
fk	*	department_id

DEPARTMENTS (DPT)		
Key type	Optionality	Column name
pk	*	department_id
	*	department_name

Purposes

- Eliminating redundant(useless) data.
- Ensuring data dependencies make sense (i.e. data is logically stored)

First Normal Form

- Requires that no multivalued attribute must exist.
- it just seems like common sense, make sure you don’t store the same data twice in the model, and that you store data in the correct place

First Normal Form

- data with the same context should be stored once in the entire ERD

First Normal Form Example

- Think about your instructor and the subject/s they are teaching. They have an ID, a name, a SSS number, subject/s to teach, and title.

First Normal Form Example

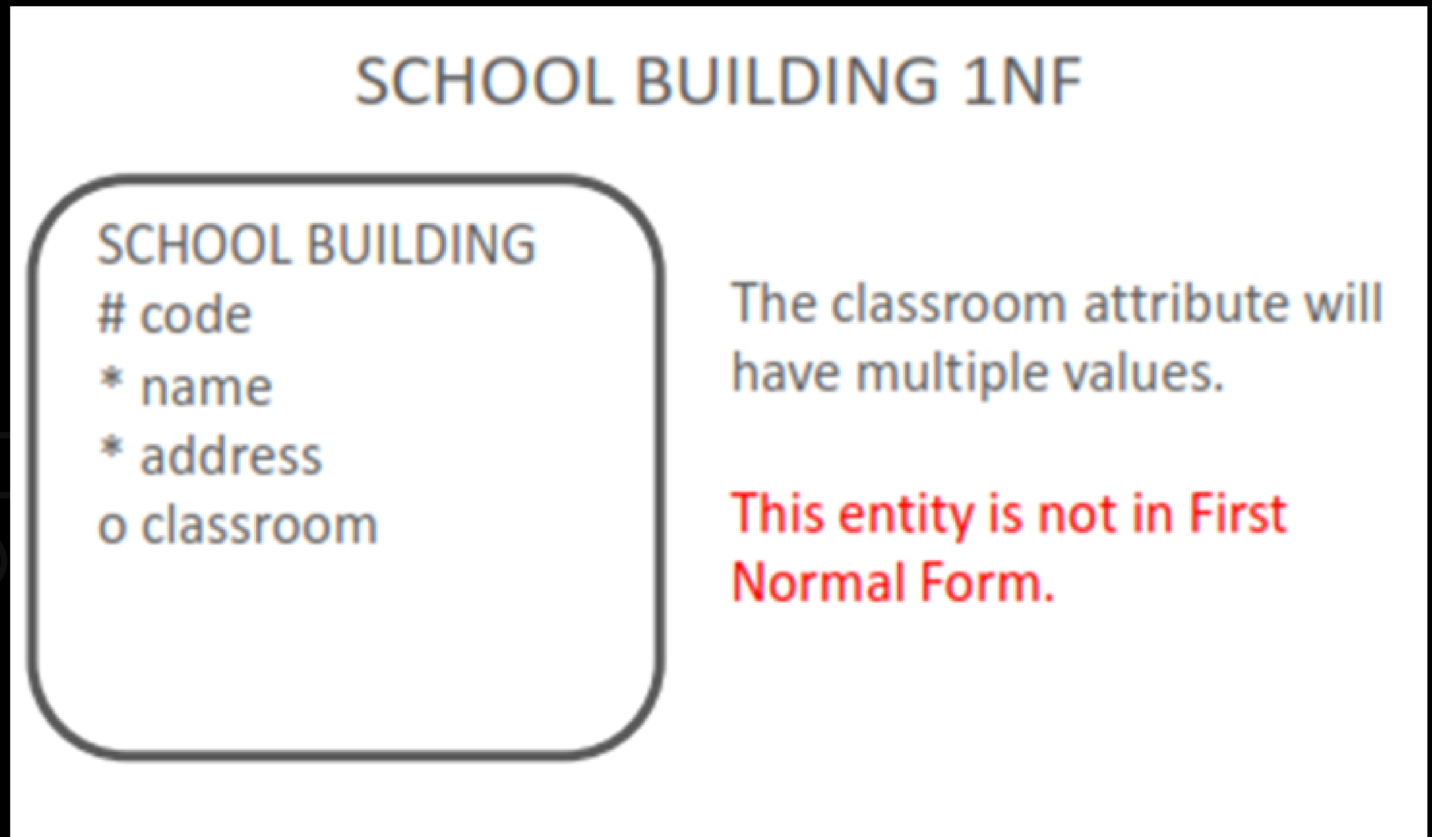
- What happens if one instace happens, and they are teaching multiple subjects?
- What happes if there are multiple titles/jobs an educator may have?

First Normal Form Step 1

- Evaluate if One Attribute Occurs Every One Instance
    - note that some attributes might be different entities themselves

First Normal Form Step 1

- Example:



First Normal Form Step 1

- Example:
  - One code, one name, and one address exist for the school building, but not one classroom.
  - Since many classrooms exist in a school building, classroom is multi-valued and violates 1NF (First Normal Form)

First Normal Form Step 1

- Evaluate if One Attribute Occurs Every One Instance
    - These rules are applied to your entities to validate it if there are attribute/s that have multiple values.

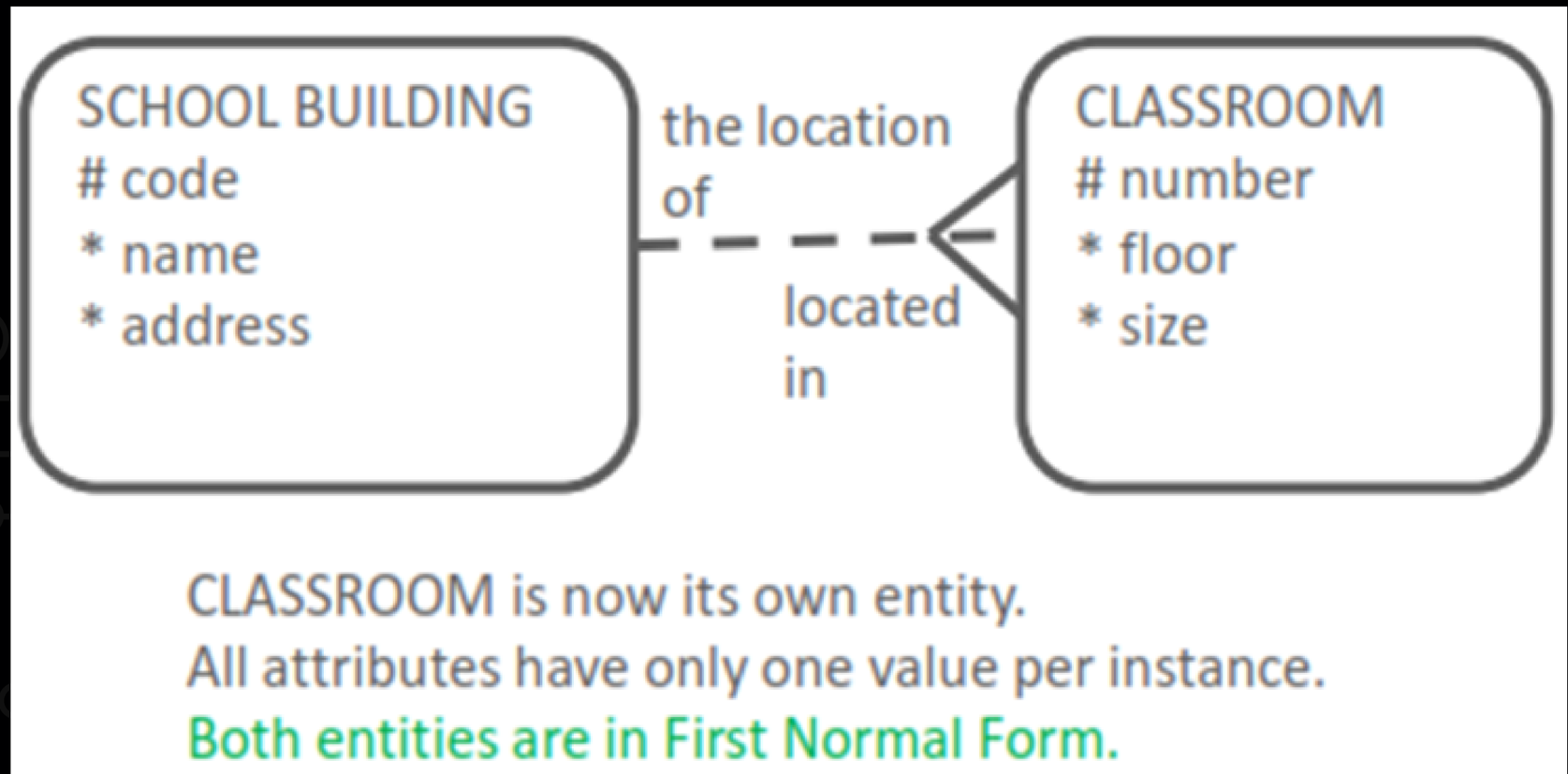
First Normal Form Step 2

- Expand
  - we ‘normalize’ now

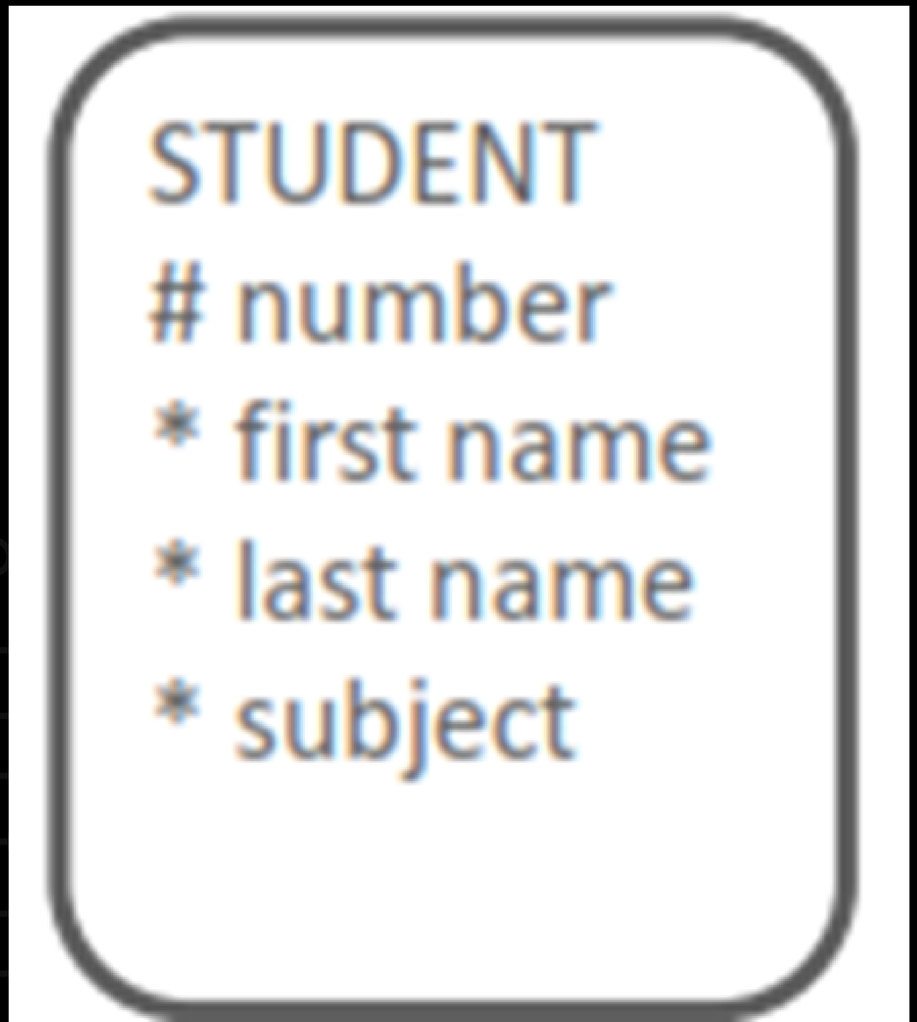
  - we have evaluated that some components might be missing or incorrect and have created a separate container, now add it

First Normal Form Step 2

- Example:



First Normal Form Example



- evaluate
- expand

Second Normal Form

- Requires that any non-UID attribute be dependent on the entire UID.

Second Normal Form

- Data must be stored in one logical place

- “store information in one place only and in the best possible place”

Second Normal Form Example

- Going back to the example of the educator, if the titles are going to change, how can the data stay consistent? If an instructor achieves a Doctorate degree and becomes a professor, can we change it? How about multiple instructors?

Second Normal Form Step 1

- Evaluate if the UIDs should belong to a single entity
  - know that each instance created might have redundant attributes or attributes that might change overtime
  - these changes must depend on the UIDs

Second Normal Form Step 1

- Evaluate if the UIDs should belong to a single entity
  - Thus the term, ‘that any non-UID attribute be dependent on the entire UID’

Second Normal Form Step 1

- Example:

Second Normal Form Step 1

- Example:
  - If one supplier supplies 5 different products, then 5 different instances are created:

Second Normal Form Step 1

- Example:

PRODUCT SUPPLIER			
supplier_number	product_number	purchase_price	supplier_name
001	1	10	Evangel
001	2	5	Evangel
001	3	2	Evangel
001	4	12	Evangel
001	5	5	Evangel

Second Normal Form Step 1

- Example:
  - What happens if the supplier name changes?
  - What if some of them were changed but not the others?
  - How would users know which name is the correct name?

Second Normal Form Step 1

- Example:
  - What happens if the supplier name changes?
  - What if some of them were changed but not the others?
  - How would users know which name is the correct name?

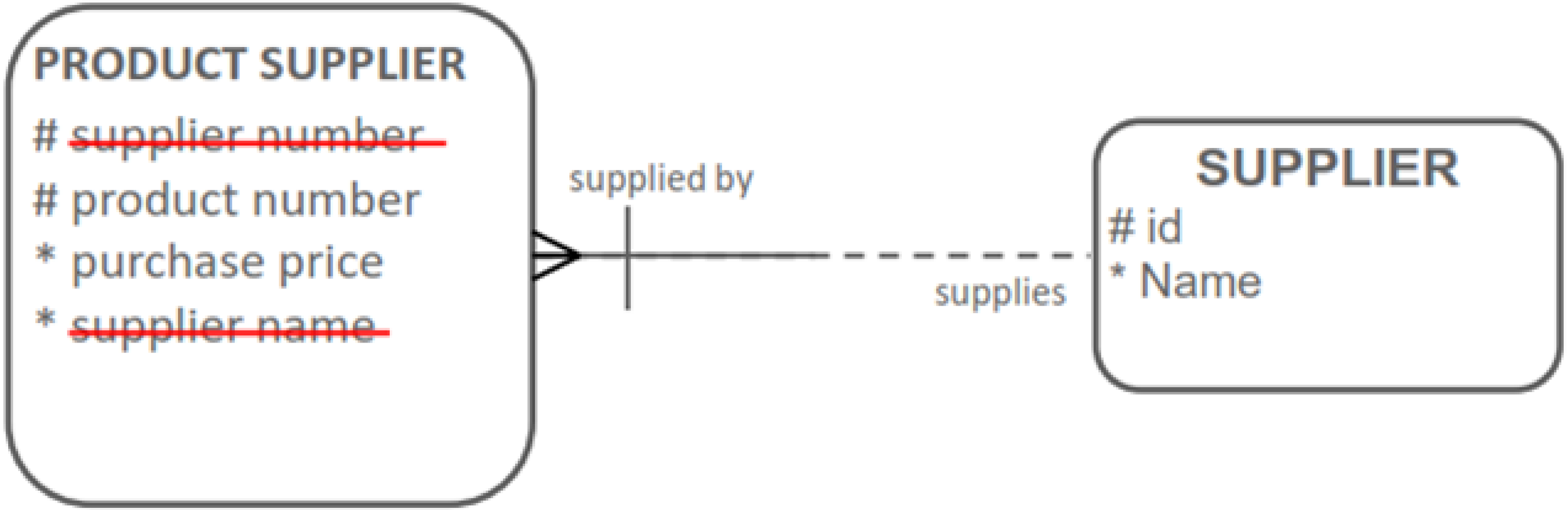
Second Normal Form Step 2

- Expand:
  - An entity is in 2NF if the UID of the entity is simple (only 1 UID).

  - Always remember this for 2NF: “Every non-UID attribute must be dependent on the whole UID” (not just part of it).

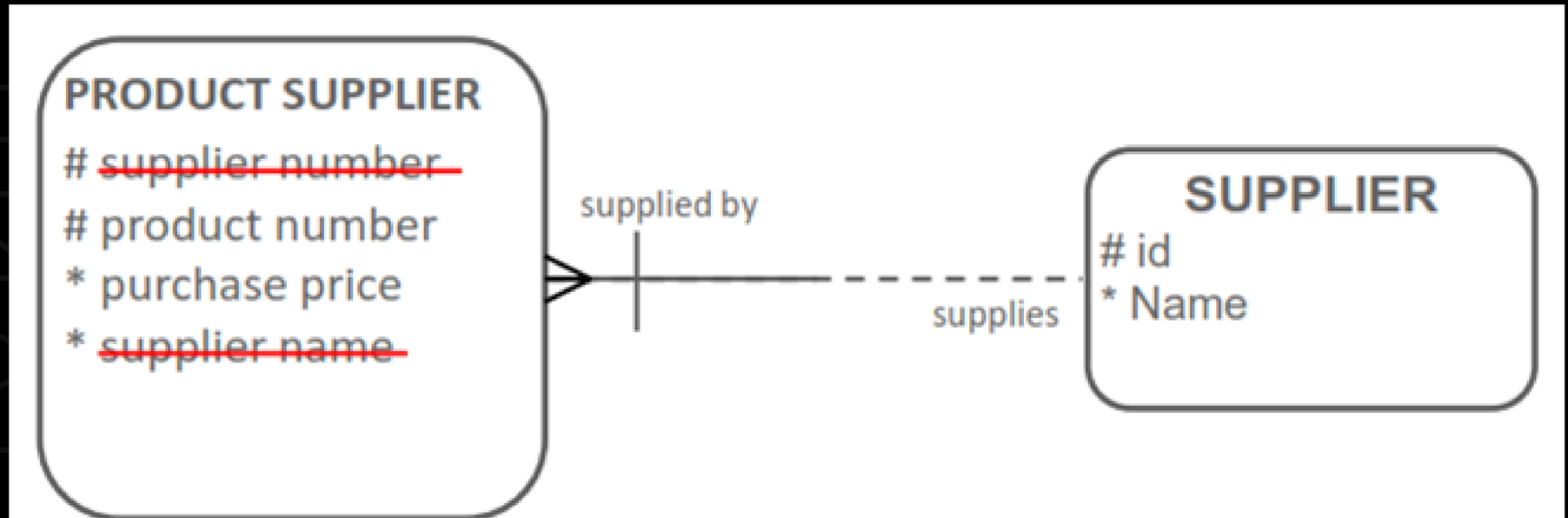
Second Normal Form Step 2

- Example:

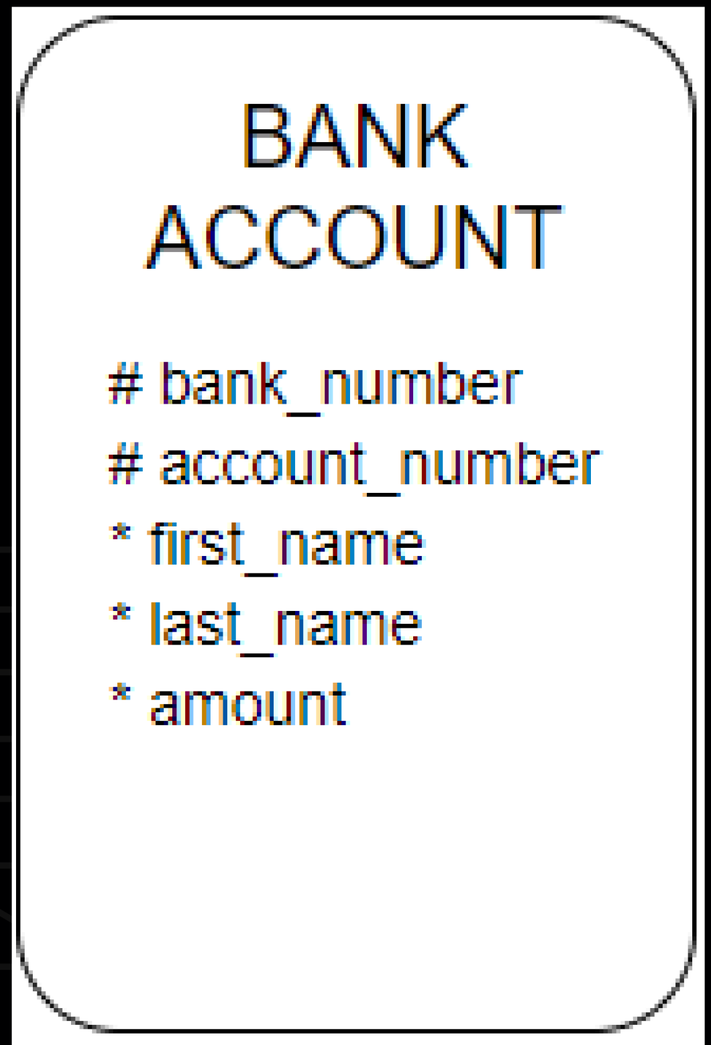


Second Normal Form Step 2

- Example:
  - Note that it also uses Barred Relationship, meaning the PRODUCT SUPPLIER would not exist if there is no SUPPLIER and the UID of the SUPPLIER is represented by the Barred relationship.



Second Normal Form Example



- evaluate
- expand

Third Normal Form

- The rule of Third Normal Form (3NF) states that no non-UID attribute can be dependent on another non-UID attribute

- All attributes must be based on the UID, not another non-UID attribute

Third Normal Form

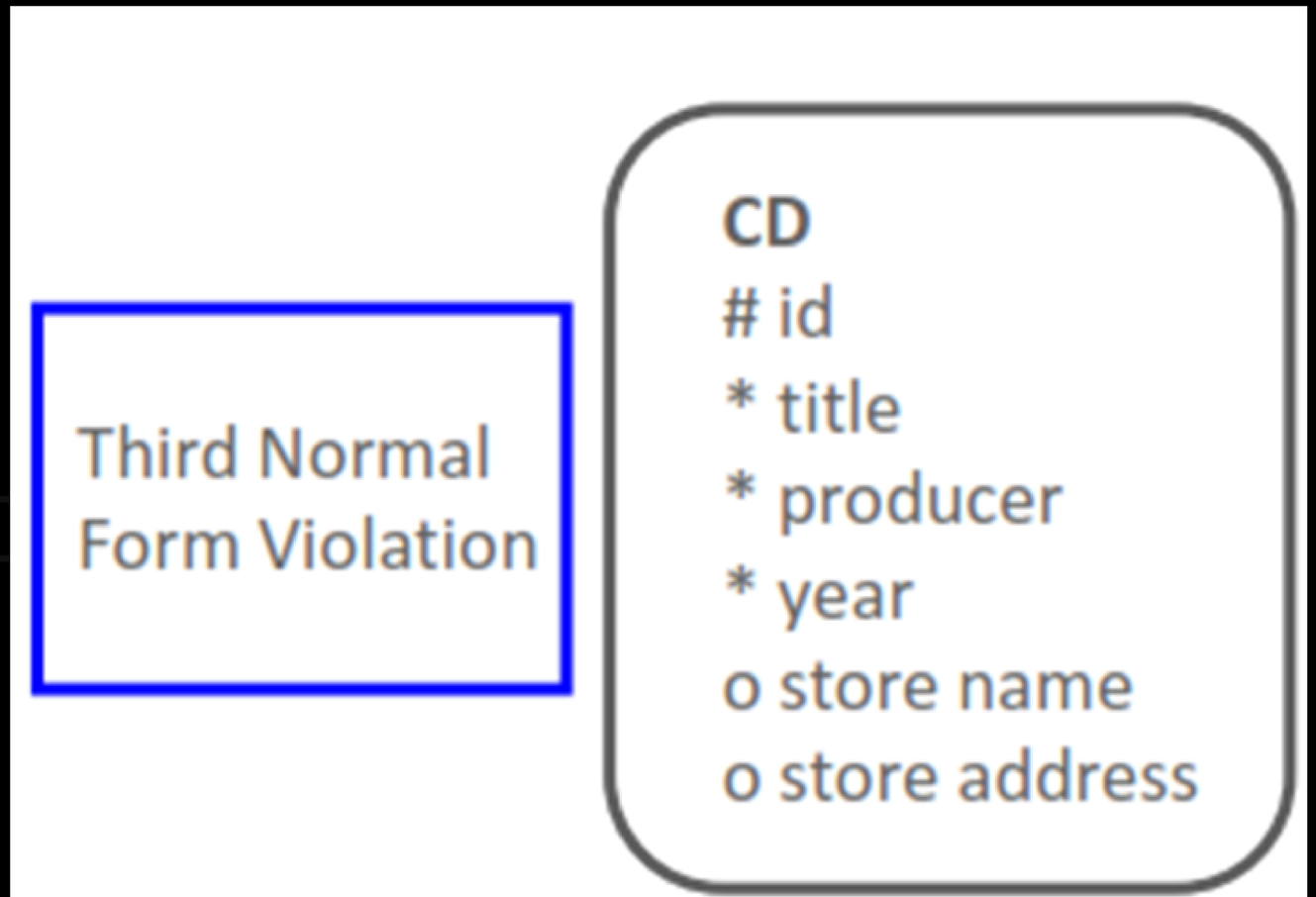
- Third Normal Form prohibits transitive dependencies.
  - A transitive dependency exists when any attribute in an entity is dependent on any other non-UID attribute in that entity.

Third Normal Form Step 1

- Evaluate if all attributes depend on the UID
  - see if all attributes belong in a single entity, check if one non-UID attribute is dependent to another non-UID attribute

Third Normal Form Step 1

- Example:



Third Normal Form Step 1

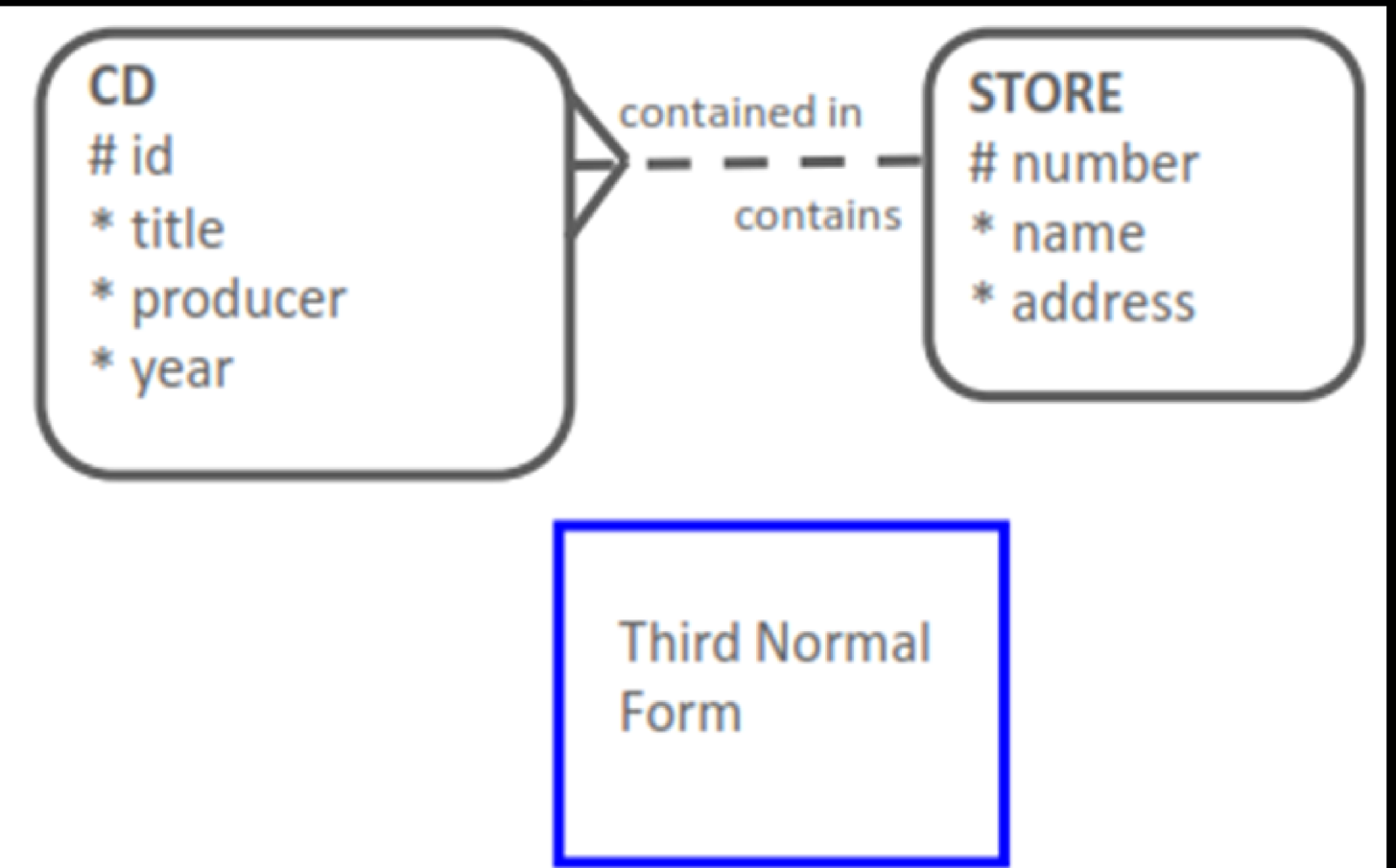
- Evaluate if all attributes depend on the UID
  - The store address is dependent on the CD number, which is the UID of the CD entity. So this entity is in 1NF and 2NF.
  - But store address is also dependent on store name, which is a non-UID attribute.

Third Normal Form Step 2

- Expand
    - build on the logic of the attributes and create another entity

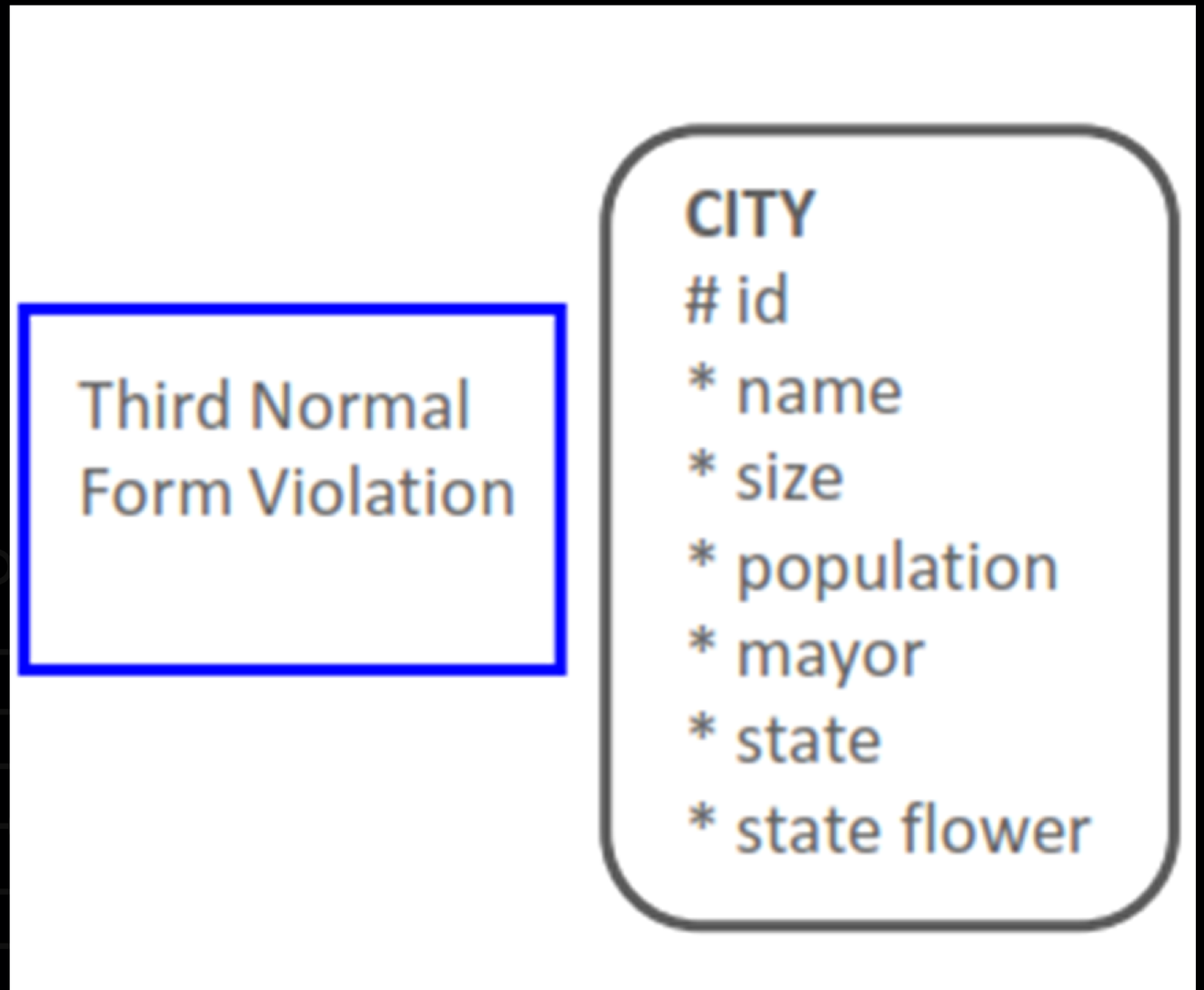
Third Normal Form Step 2

- Example:



- create a second entity STORE, with a relationship to CD.

Third Normal Form Example



- evaluate
- expand

Conclusion

- 1NF is about making sure that attributes have only one value for all instances of one entity

Conclusion

- 2NF is making sure single UIDs are the masters of all attributes in the entity

Conclusion

- 3NF is making sure that non-UID attributes (mandatory and optional attributes) are not dependent on another non-UID attribute