

The Need & Rationale

- *Spreadsheet syndrome*
- Goal:
 - reduce redundancies and inconsistencies
 - efficient updates
 - eliminate anomalies
- Normalization solves these problems

Normal Forms

- Criteria for safety against anomalies & inconsistencies
- Usually 3 to 7 normal forms (1NF $\rightarrow$ 7NF)
 - First 3: how non-key attributes relate to key
 - 4th & 5th: many-to-one, many-to-many
 - Must be satisfied progressively
 - May combine several in one step
- Highest Normal Form (**HNF**)
- Applied to individual tables

First Normal Form

- A relation *is* a relation:
 - **no repetition in a tuple**
 - one (at least) unique key
 - no *nullable* attribute (optional)
- Atomic attribute values
 - For e.g., decompose *multi-valued* attributes
- Atomicity can be extreme: *date, strings*

Second Normal Form

- Conditions:
 - Relation is in 1NF
 - No non-prime attributes functionally depend on subset of PK

Second Normal Form

- Conditions:
 - Relation is in 1NF
 - No non-prime attributes functionally depend on subset of PK
- Not in 2NF:
 - *books*(book_name, author_name, review)
 - *skills*(employee_name, skill, address)

Second Normal Form

- Conditions:
 - Relation is in 1NF
 - No non-prime attributes functionally depend on subset of PK
- Not in 2NF:
 - *books*(book_name, author_name, review)
 - *skills*(employee_name, skill, address)
- Solution: separate into multiple relations

Second Normal Form

- Conditions:
 - Relation is in 1NF
 - No non-prime attributes functionally depend on subset of PK
- Not in 2NF:
 - *books*(*book_name*, *author_name*, *review*)
 - *skills*(*employee_name*, *skill*, *address*)
- Solution: separate into multiple relations
 - For e.g.: *employee_name*, *skill* and *employee_name*, *address*
 - Both are in 2NF

Second Normal Form

- Conditions:
 - Relation is in 1NF
 - No non-prime attributes functionally depend on subset of PK
- Not in 2NF:
 - *books*(*book_name*, *author_name*, *review*)
 - *skills*(*employee_name*, *skill*, *address*)
- Solution: separate into multiple relations
 - For e.g.: *employee_name*, *skill* and *employee_name*, *address*
 - Both are in 2NF
- single primary key $\Rightarrow$ 2NF

Second Normal Form

- Conditions:
 - Relation is in 1NF
 - No non-prime attributes functionally depend on subset of PK
- Not in 2NF:
 - *books*(*book_name*, *author_name*, *review*)
 - *skills*(*employee_name*, *skill*, *address*)
- Solution: separate into multiple relations
 - For e.g.: *employee_name*, *skill* and *employee_name*, *address*
 - Both are in 2NF
- single primary key $\Rightarrow$ 2NF
- Can have anomalies in 2NF

AuthorAwards(*award_name*, *award_year*, *author_name*, *date_of_birth*)

Third Normal Form

- Conditions:

- Relation is in 2NF
- Every non-key is directly dependent on the key
 - “every non-key attribute must provide a fact about the key, the whole key, and nothing but the key.”

Third Normal Form

■ Conditions:

- Relation is in 2NF
- Every non-key is directly dependent on the key
 - “every non-key attribute must provide a fact about the key, the whole key, and nothing but the key.”

AuthorAwards(*award_name, award_year*, *author_name, date_of_birth*)

Third Normal Form

■ Conditions:

- Relation is in 2NF
- Every non-key is directly dependent on the key
 - “every non-key attribute must provide a fact about the key, the whole key, and nothing but the key.”

AuthorAwards(award_name, award_year, author_name, date_of_birth)

■ *date_of_birth* intransitively dependent on key

- $(author_name, award_year) \rightarrow author_name \rightarrow date_of_birth$

■ Solution: separate into multiple relations

- *AuthorAwards(award_name, award_year, author_name)* &
Author(author_name, date_of_birth)

■ Other examples:

- *Employees(Empl_id, Empl_name, Dept_name, Dept_floor)* :

Third Normal Form

■ Conditions:

- Relation is in 2NF
- Every non-key is directly dependent on the key
 - “every non-key attribute must provide a fact about the key, the whole key, and nothing but the key.”

AuthorAwards(*award_name*, *award_year*, *author_name*, *date_of_birth*)

■ *date_of_birth* intransitively dependent on key

- $(author_name, award_year) \rightarrow author_name \rightarrow date_of_birth$

■ Solution: separate into multiple relations

- *AuthorAwards*(*award_name*, *award_year*, *author_name*) &
Author(*author_name*, *date_of_birth*)

■ Other examples:

- *Employees*(*Empl_id*, *Empl_name*, *Dept_name*, *Dept_floor*) :
Employees(*Empl_id*, *Empl_name*, *Dept_id*) and
Department(*Dept_id*, *Dept_name*, *Dept_floor*)

■ 3NF: free of update, insertion, and deletion anomalies

Fourth Normal Form

- Only **many-to-one** and **many-to-many**