Database Normalization

Presented by

Rohit Pachlor

(Teacher & Researcher)

Key points

- Normalization
- Functional dependency
- Key attribute and keys in database
- Normal forms (NF)
- Normalization Process
- 1NF
- 2NF
- 3NF
- BCNF (Boyce-codd)

Normalization

- Process of decomposing a complex relation into less complex relations
  - Lossless decomposition
  - Dependency preserving
- Why do we need Normalization
  - To reduce/eliminate redundancy
    - ✓ It causes Modification anomalies
  - To maintain data consistency

Student Relation: S

S_id	S_name	C_id	C_name
S1	Rohit	101	C1
S2	Rahul	101	C1
S3	Rinku	102	C2
S4	Shree	102	C2
S5	Nirvana	103	C3

- Insert anomaly
  - can’t insert information of new course. Or We have to use dummy data to insert information of new course
- Deletion anomaly
  - can’t delete information of sid:S5 because we lost information of C3
- Updation anomaly
  - if we want to update c_name of cid:101, we have to update it at many places. If not, database become inconsistent.

Example: Normalized Relation (NR)

Student Relation: S1

S_id	S_name	C_id
S1	Rohit	101
S2	Rahul	101
S3	Rinku	102
S4	Shree	102
S5	Nirvana	103

Student Relation: S2

C_id	C_name
101	C1
102	C2
103	C3

- While decomposing
  - Define foreign key in old table
  - Refer it new table as primary key
- Is this decomposition lossless?
  - Yes
- Is this decomposition dependency preserving?
  - Yes
- How does redundancy occurs in relation
  - Because of function dependency
  - Because of Multivalued dependency
- Disadvantage of Normalization
  - It increases query processing cost
  - It increases maintenance cost

Functional Dependency (FD)

- Fd: X->Y Where X and Y ∈ relation R
  - Y is functionally determinate by X
  - Or X uniquely determines Y
  - If the value of X replicates, value of Y must also replicates.
  - X is determinant
  - Y is dependent
- Mathematically
  - If T1[X]=T2[X] then T1[Y]=T2[Y]

- Type of Dependencies
  - Trivial
    - ✓ Dependencies which will always hold in a relation.
    - ✓ X->Y is trivial if Y ⊆ X
    - ✓ Example AB->A
  - Non trivial
    - ✓ X->Y is non-trivial if Y is not ⊆ X
    - ✓ Example AB->ABC
- Properties of FD
  - Reflexive **if** Y ⊆ X **then** X-> Y
  - Transitive **if** X->Y and Y->Z **then** X->Z
  - Augmentation **if** X->Y **then** XZ->YZ
  - Union **if** X->Y and X->Z **then** X->YZ
  - Decomposition **if** X->YZ **then** X->Y and X->Z
  - Composition **if** X->Y and Z->W **then** XZ->YW
  - Pseudo transitivity **if** X->Y and YZ->W **then** XZ->W

Keys in database

- Key attribute
  - Any attribute or minimal set of attributes in a relation R, which uniquely identifies every tuple of R or all attributes of R is called key attribute.
  - Simple key attribute
    - ✓ Indivisible key attribute
  - Composite key attribute
    - ✓ Divisible
- Candidate Key Vs. Primary Key Vs. Alternate Key Vs. Super Key Vs. Foreign Key
- Every Candidate key is a super key but vice versa is not true.

- How to find key attribute?
  - Find the closure of attribute
    - ✓ The closure of a attribute ***X*** contains all the attribute which are functionally determine by ***X***.
- FD set for student relation S

  s_id->s_name c_id

  c_id->c_name

  s_id^+ ={s_id,s_name, c_id, c_name}
- So s_id is only candidate key and therefore primary key, no alternate key for this relation.
- And every super set of s_id is super key for this relation.

Normal forms

- 1NF, 2NF, 3NF, BCNF
  - ➢ Reduces redundancy due to FD
  - ➢ If a relation in BCNF then it is free from redundancy due to FDs
- 4NF, 5NF, DKNF, Higher Normal forms
  - ➢ Reduces redundancy due to Multivalued FD.
- A relation in a given normal form is free from certain set of modification anomalies
- To be in a Normal form, a relation must satisfy a condition
  - 1NF=C1
  - 2NF=1NF+C2
  - 3NF=2NF+C3
  - BCNF=3NF+C4

Where
C1: Domain of all attributes must be atomic
C2: No partial FD
C3: No Transitive FD
C4: Every attribute must be dependent on super key only

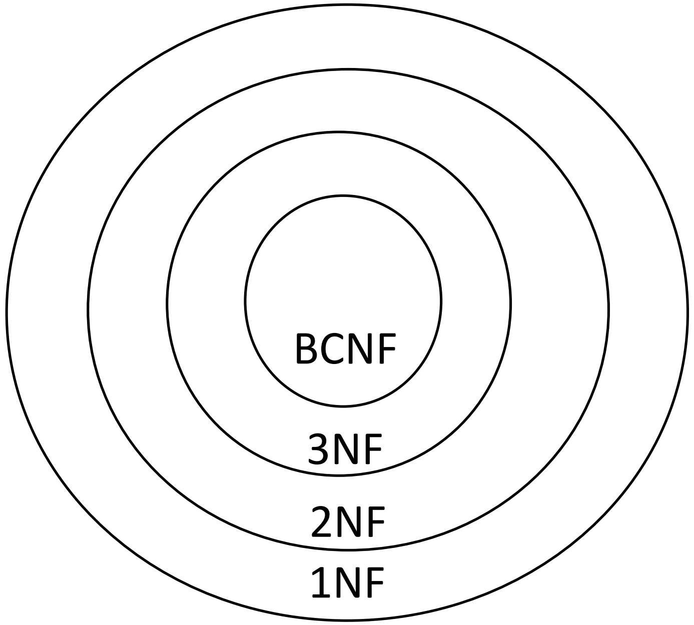

Fig. 1: Normal forms are cumulative

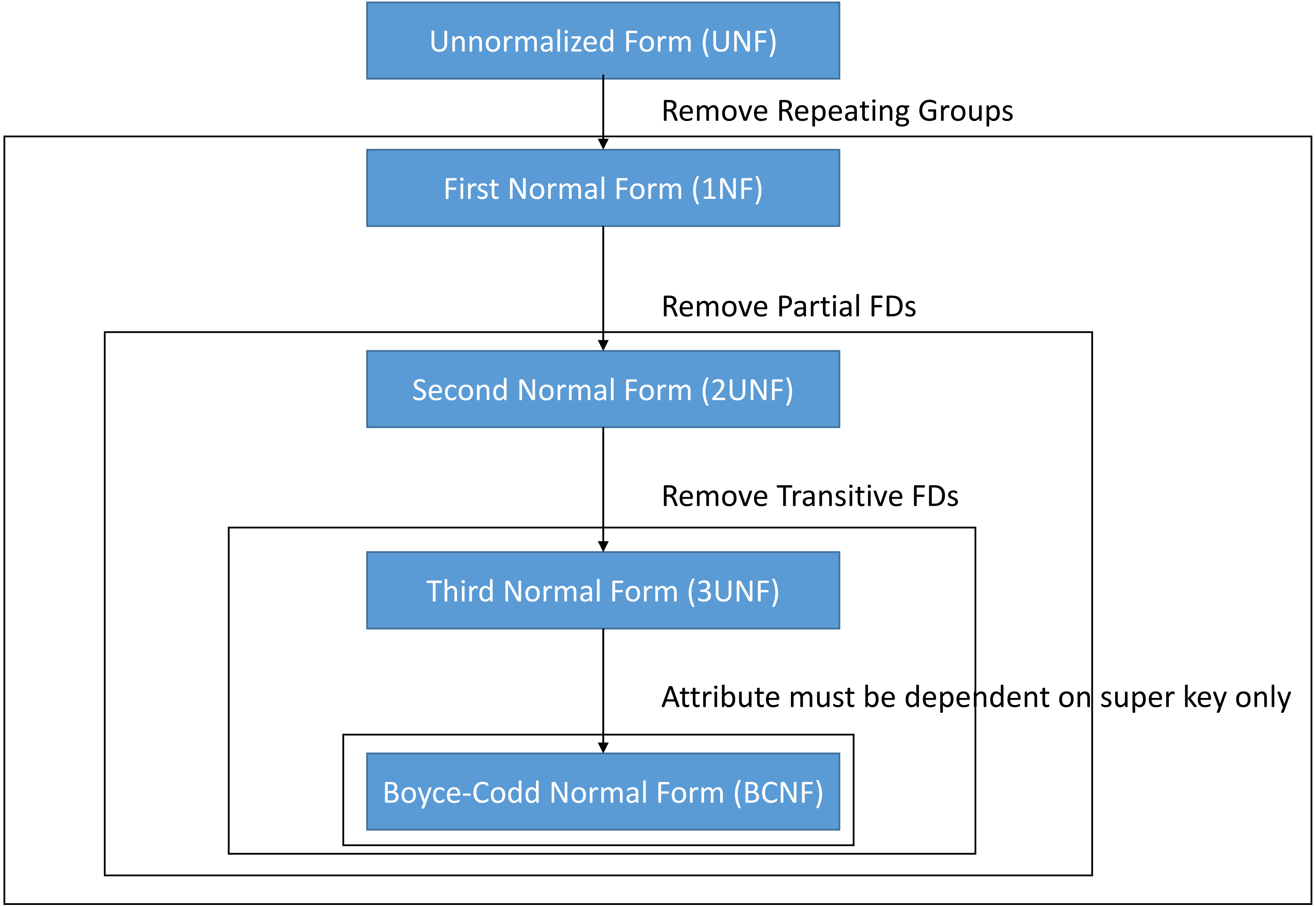

Fig. 2: Normal forms

Normalization Process

- Identify the FD set for relation
- Identify the candidate keys for relation
- Apply the definition of each normal form
- If the relation fails to meet the definition of normal form
  - Decompose it until it meets the definition of desired normal form
- Reapply the definition of normal forms to ensure it meet the definition of normal form.

1NF

- A relation R is said to be in 1NF, iff
  - The domain of each attribute must be atomic
    - ✓ A domain is said to be atomic if the values of the domain are indivisible unit and not set of values .
    - ✓ In other words, repeated groups are not allowed in 1NF
    - ✓ Multivalued and composite attribute are not allowed in 1NF

Relation S

S_id	S_Name	C_Name
S1	XYZ	C/C++
S2	YXZ	C++/Java
S3	YZX	Java/C
S4	XZY	C
S5	ZXY	Java

Relation S

S_id	S_Name	C_Name
S1	XYZ	C
S1	XYZ	C++
S2	YXZ	C++
S2	YXZ	Java
S3	YZX	Java
S3	YZX	C
S4	XZY	C
S5	ZXY	Java

Relation S1

S_id	S_Name
S1	XYZ
S2	YXZ
S3	YZX
S4	XZY
S5	ZXY

Relation S2

S_id	C_Name
S1	C
S1	C++
S2	C++
S2	Java
S3	Java
S3	C
S4	C
S5	Java

2NF

- A relation is said to be in 2NF iff,
  - It is in 1NF
  - No partial FDs are present in it
- A FD (X->Y) is said to partial dependency (PD) iff
  - X is prime attribute
  - Y is non-prime attribute
  - Prime attribute is part of composite candidate key
  - non-prime attribute are not part of candidate key
- Mathematically

$X\text{->}Y$ is PD $\Leftrightarrow$ X is not $\subset$ CK $\wedge$ Y:NPA

FIRST (supplier_no, status, city, part_no, quantity)

- **FD Set**

(supplier_no, part_no) -> quantity

supplier_no -> status

supplier_no -> city

city -> status

- **Candidate key(s) (CK)**

(Supplier_no, part_no)

- **Prime attribute (PA)**

supplier_no, part_no

- **Non prime attribute (NPA)**

status, city, quantity

- **2NF Decompostion**

SECOND (supplier_no, status, city)

SUPPLIER_PART (supplier_no, part_no, quantity)

3NF

- A relation is said to be in 3NF iff,
  - It is in 2NF
  - No Transitive FDs are present in it.
  - A non-prime attribute must not transitively dependent on key attribute
- Mathematically

$\forall\ FD_{NT}: X\text{->}Y \Longleftrightarrow [X: SK \vee Y{:}PA]$

- SECOND (supplier_no, status, city)
- FD Set

supplier_no -> status

supplier_no -> city

city -> status

- CK

Supplier_no

- 3NF Decomposition

SUPPLIER_CITY (supplier_no, city)

CITY_STATUS (city, status)

BCNF

- A relation is said to be in BCNF iff,
  - It is in 3NF
  - if and only if every determinant is a super key
- Mathematically
- $\forall$ FD_{NT}: X->Y $\Longleftrightarrow$X:SK

- SUPPLIER_PART (supplier_no, supplier_name, part_no, quantity)
- **FD set**

(supplier_no, part_no) -> quantity

(supplier_no, part_no) -> supplier_name

(supplier_name, part_no) -> quantity

(supplier_name, part_no) -> supplier_no

supplier_name -> supplier_no

supplier_no -> supplier_name

- **CK:**

(supplier_no, part_no) and (supplier_name, part_no)

- **Decomposition:**
- SUPPLIER_ID (supplier_no, supplier_name)
- SUPPLIER_PARTS (supplier_no, part_no, quantity)

References

- Ramez Elmasri , Shamkant Navathe, Fundamentals of Database Systems, Addison-Wesley Publishing Company, 2010
- Henry F. Korth , Abraham Silberschatz, Database system concepts, McGraw-Hill, Inc., New York, NY, 1986
- https://www.geeksforgeeks.org/database-normalization-normal-forms/
- http://www2.york.psu.edu/~lxn/IST_210/normal_form_definitions.html
- YouTube| KNOWLEDGE GATE| DBMS
- YouTube| Gate Instructors| Database Management System – DBMS
- YouTube| Gate Lectures by Ravindrababu Ravula