EIU
TRƯỜNG ĐẠI HỌC
QUỐC TẾ
MIỀN ĐÔNG
EASTERN
INTERNATIONAL
UNIVERSITY

CSE301 – DATABASE

Normalization

Normalization

What is Normalization?

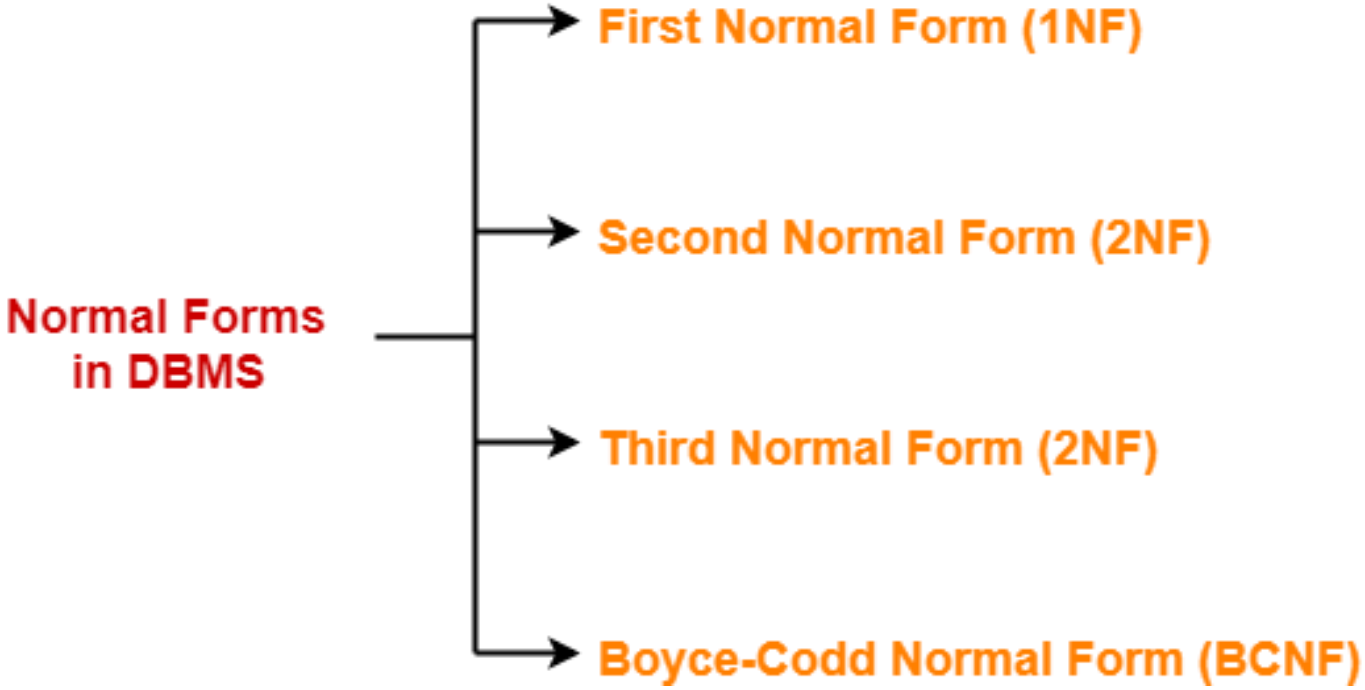
- Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies.
- It is a process of organizing data in the database
  - ✓ Minimize redundancy in relation.
  - ✓ Divides larger tables into smaller tables and links them using relationships.

What is redundancy?

- When some data is stored multiple times unnecessarily in a database.

Student_id	Name	Age	Dept_Code	Dept_Name	Hod_Name
1	A	18	101	SE	XYZ
2	B	19	101	SE	XYZ
3	C	18	101	SE	XYZ
4	D	20	102	CNDC	PQR
5	E	21	102	CNDC	PQR
6	F	19	103	ECE	KLM

Normal Forms



- There exists several other normal forms even after BCNF but generally we normalize till BCNF only.

First Normal Form (1NF)

- A given relation is called in First Normal Form (1NF) :
  - if each cell of the table contains only an atomic value.

    Or

    if the attribute of every tuple is either single valued or a null value.
  - Order of rows & order of columns are irrelevant.
  - In every column all value must be in same domain.
  - Every column should have a unique name
- Note:
  - ✓ By default, every relation is in 1NF.

First Normal Form (1NF)

- Example:
  - ✓ The following relation is not in 1NF:

Student_id	Name	Subjects
100	Dat	Computer Networks, Data Structure
101	Duc	Database, Java, Software Engineering
102	Phuc	Database

First Normal Form (1NF)

- ✓ This relation can be brought into 1NF.
- ✓ This can be done by rewriting the relation such that each cell of the table contains only one value.

Student_id	Name	Subjects
100	Dat	Computer Networks
100	Dat	Data Structure
101	Duc	Database
101	Duc	Java
101	Duc	Software Engineering
102	Phuc	Database

Second Normal Form (2NF)

- A given relation is called in Second Normal Form (2NF) if and only if
  - ✓ Relation already exists in 1NF.
  - ✓ No partial dependency exists in the relation ([prime → non-prime] should not be present)

Second Normal Form (2NF)

- Example: The following relation is not in 2NF:
- R(ABCD), FD: {AB → D, B → C}
- Find candidate key =?
  - $\{AB\}^+ = AB = ABCD = R$ So, Candidate key is = AB
- Prime attributes: A, B (part of candidate key)
- Non-prime attributes: C, D
- Now check all FD's. Is there any (Prime → Non-prime) exists or not?
- B → C is Prime → Non-prime. So this relation is not in 2NF.

Second Normal Form (2NF)

- Example: How to decompose this relation into 2NF?
- If a partial dependency exists,
- we remove the partially dependent attribute(s) from the relation by placing them in a new relation along with a copy of their determinant (Create a new table for each partial dependency).
- R(ABCD), FD: {AB → D, B → C} will be,
  - R_1(<u>AB</u>D), FD: {AB → D} here AB is key
  - R_2(<u>B</u>C), FD: {B → C} here B is key

Second Normal Form (2NF)

- Example: The following relation is in 2NF:
- R (V , W , X , Y , Z) FD: {VW → XY, Y → V, WX → YZ}
- The possible candidate keys for this relation are: VW , WX , WY
- From here, Prime attributes = { V , W , X , Y }, Non-prime attributes = { Z }
- Now, if we observe the given dependencies, there is no partial dependency.
- This is because there exists no (prime → non-prime) dependency where incomplete candidate key determines any non-prime attribute.
- Thus, we conclude that the given relation is in 2NF.

Third Normal Form (3NF)

- A given relation is called in Third Normal Form (3NF) if and only if
  - ✓ Relation already exists in 2NF.
  - ✓ No transitive dependency exists for non-prime attributes.

Third Normal Form (3NF)

- Example: The following relation is not in 3NF:
- R(ABC), FD: {A → B, B → C}
- Find candidate key =?
  - = $\{A\}^+$ = AB = ABC = R So, Candidate key is = A
- Prime attributes: A (part of candidate key)
- Non-prime attributes: B, C
- Now check all FD's. Is there any (Non-prime → Non-prime) exists or not?
- B → C is Non-prime → Non-prime. So this relation is not in 3NF.

Third Normal Form (3NF)

- Example: How to decompose this relation into 3NF?
- If a partial dependency exists, first we remove the partially dependent attribute(s) from the relation by creating a new table for each partial dependency.
- If a transitive dependency exists, we remove the transitively dependent attribute(s) from the relation by placing the attribute(s) in a new relation along with a copy of the determinant. (Create a new table for each transitive dependency).
- R(ABC), FD: {A → B, B → C} will be,
  - R_1(<u>A</u>B), FD: {A → B} here A is key
  - R_2(<u>B</u>C), FD: {B → C} here B is key

Third Normal Form (3NF)

- Example: The following relation is in 3NF:
- R (A , B , C , D , E) FD: {A → BC, CD → E, B → D, E → A}
- Find candidate key =?
  - So, Candidate keys are A , E , CD , BC.
- Prime attributes: A , B, C, D, E (part of candidate key)
- Non-prime attributes: There are no non-prime attributes
- Now check all FD's. Is there any (Non-prime → Non-prime) exists or not?
- There is no Non-prime → Non-prime. So this relation is in 3NF.

Find the Normal Forms: Exercises

- R(ABCDE), FD: {CE → D, D → B, C → A}
  - ✓ CE is candidate key
  - ✓ It is in 1NF. Because C → A is a partial dependency.
- R(ABCDEF), FD: {AB → C, DC → AE, E → F}
  - ✓ ABD, BCD are candidate keys
  - ✓ It is in 1NF. DC → AE {DC → A, DC → E}, Because DC → E is a partial dependency
- R(ABCDE), FD: {AB → CD, D → A, BC → DE}
  - ✓ AB, BD, BC are candidate keys
  - ✓ It is in 3NF. Because all LHS are not super key, as well as no partial or transitive FD's.