NORMALIZATION

IN DBMS, database normalization is a process of making the database consistent by :

- **Reducing the redundancies**
- **Ensuring the integrity of data through lossless decomposition**

Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion, and updation anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.

1. First Normal Form -

If a relation contains a composite or multi-valued attribute, it violates the first normal form or the relation is in the first normal form if it does not contain any composite or multi-valued attribute. A relation is in first normal form if every attribute in that relation is singled valued attribute.

Example 1 - Relation STUDENT in table 1 is not in 1NF because of multivalued attribute STUD_PHONE. Its decomposition into 1NF has been shown in table 2

Stud_no	Stud_name	Stud_phone	Stud_state	Stud_country
1	AYUSH	9746575757 9822164646	Haryana	India
2	AYUSH	9503194847	Jaipur	India
3	RAJ		Jaipur	India

TABLE 1
After conversion to first normal form refer table 2

Stud_no	Stud_name	Stud_phone	Stud_state	Stud_country
1	AYUSH	9746575757	Haryana	
1	AYUSH	9822164646	Haryana	India
2	AYUSH	9503194847	Jaipur	India
3	RAJ		Jaipur	India

TABLE 2: 1NF

2.Second Normal Form

To be in second normal form, a relation must be in first normal form and the relation must not contain any partial dependency. A relation is in 2NF if it has **No Partial Dependency**, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.

Partial Dependency - If the proper subset of candidate key determines non-prime attribute, it is called partial dependency.

Example 2 -

STUD_NO	COURSE_NO	COURSE_FEE
1	c1	1000
2	c2	1500
1	c4	2000
4	c3	1000
4	c1	1000
2	c5	2000

{Note that, there are many courses having the same course fee. }

Here, COURSE_FEE cannot alone decide the value of COURSE_NO or STUD_NO;

COURSE_FEE together with STUD_NO cannot decide the value of COURSE_NO;

COURSE_FEE together with COURSE_NO cannot decide the value of STUD_NO;

Hence,
COURSE_FEE would be a non-prime attribute, as it does not belong to the one only candidate key {STUD_NO, COURSE_NO} ;
But, COURSE_NO -> COURSE_FEE, i.e., COURSE_FEE is dependent on COURSE_NO, which is a proper subset of the candidate key. Non-prime attribute COURSE_FEE is dependent on a proper subset of the candidate key,
which is a partial dependency and so this relation is not in 2NF.
To convert the above relation to 2NF,
we need to split the table into two tables such as :
Table 1: STUD_NO, COURSE_NO
Table 2: COURSE_NO, COURSE_FEE

TABLE 1

STUD_NO	COURSE_NO
1	c1
2	c2
1	c4
4	c3
4	c1
2	c5

TABLE 2

COURSE_NO	CO
c1	1000
c2	1500
c3	1000
c4	2000
c5	2000

NOTE: 2NF tries to reduce the redundant data getting stored in memory. For instance, if there are 100 students taking the C1 course, we don't need to store its Fee as 1000 for all the 100 records, instead, once we can store it in the second table as the course fee for C1 is 1000.

3.Third Normal Form

A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.
A relation is in 3NF if at least one of the following condition holds in every non-trivial function dependency X --> Y

0. X is a super key.
1. Y is a prime attribute (each element of Y is part of some candidate key).

Stud_no	Stud_name	Stud_state	Stud_country	Stud_age
1	AYUSH	Nagpur	India	25
2	AYUSH	Jaipur	India	21
3	RAJ	Jaipur	India	23

Table 4

Transitive dependency - If A->B and B->C are two FDs then A->C is called transitive dependency.

Example 1 - In relation STUDENT given in Table 4,
FD set: {STUD_NO -> STUD_NAME, STUD_NO -> STUD_STATE, STUD_STATE ->

STUD_COUNTRY, STUD_NO -> STUD_AGE}
Candidate Key: {STUD_NO}
For this relation in table 4, STUD_NO -> STUD_STATE and STUD_STATE -> STUD_COUNTRY are true. So STUD_COUNTRY is transitively dependent on STUD_NO.
It violates the third normal form.
To convert it in third normal form, we will decompose the relation STUDENT (STUD_NO, STUD_NAME, STUD_PHONE, STUD_STATE, STUD_COUNTRY_STUD_AGE) as: STUDENT (STUD_NO, STUD_NAME, STUD_PHONE, STUD_STATE, STUD_AGE) STATE_COUNTRY (STATE, COUNTRY)

4. Boyce-Codd Normal Form (BCNF)

A relation R is in BCNF if R is in Third Normal Form and for every FD, LHS is super key. A relation is in BCNF iff in every non-trivial functional dependency X --> Y, X is a super key.

Example 1 - Find the highest normal form of a relation R(A,B,C,D,E) with FD set as {BC->D, AC->BE, B->E}

Step 1. As we can see, (AC)+ ={A,C,B,E,D} but none of its subset can determine all attribute of relation, So AC will be candidate key. A or C can't be derived from any other attribute of the relation, so there will be only 1 candidate key {AC}.

Step 2. Prime attributes are those attribute which are part of candidate key {A, C} in this example and others will be non-prime {B, D, E} in this example.

Step 3. The relation R is in 1st normal form as a relational DBMS does not allow multi-valued or composite attributes.
The relation is in 2nd normal form because BC->D is in 2nd normal form (BC is not a proper subset of candidate key AC) and AC-> BE is in 2nd normal form (AC is candidate key) and B-> E is in 2nd normal form (B is not a proper subset of candidate key AC).
The relation is not in 3rd normal form because in BC->D (neither BC is a super key nor D is a prime attribute) and in B-> E (neither B is a super key nor E is a prime attribute) but to satisfy 3rd normal for, either LHS of an FD should be super key or RHS should be a prime attribute.
So the highest normal form of relation will be the 2nd Normal form.

4.Fourth Normal Form (4NF)

The Fourth Normal Form (4NF) is a level of database normalization where there are no non-trivial multivalued dependencies other than a candidate key. It builds on the first three normal forms (1NF, 2NF, and 3NF) and the Boyce-Codd Normal Form (BCNF). It states that, in addition to a database meeting the requirements of BCNF, it must not contain more than one multivalued dependency.

Properties

A relation R is in 4NF if and only if the following conditions are satisfied:
1. It should be in the Boyce-Codd Normal Form (BCNF).
2. The table should not have any Multi-valued Dependency.

A table with a multivalued dependency violates the normalization standard of the Fourth Normal Form (4NF) because it creates unnecessary redundancies and can contribute to inconsistent data. To bring this up to 4NF, it is necessary to break this information into two tables.

Example:

Table R1

Company	product
c1	Laptop
c1	headphone
c2	tv
c2	tv

Table R2

Agent	Company
Solid snake	c1
Solid snake	c2
Cipher	c1

Table R3

agent	product
Solid snake	Laptop
Solid snake	headphone
Solid snake	tv
Cipher	tv

Table R1⋈R2⋈R3

Company	Product	Agent
c1	Laptop	Solid snake
c1	headphone	Solid snake
c2	tv	Cipher
c1	tv	Solid snake

5.Fifth Normal Form/Projected Normal Form (5NF)

A relation R is in Fifth Normal Form if and only if everyone joins dependency in R is implied by the candidate keys of R. A relation decomposed into two relations must have lossless join Property, which ensures that no spurious or extra tuples are generated when relations are reunited through a natural join.

Properties
A relation R is in 5NF if and only if it satisfies the following conditions:
1. R should be already in 4NF.
2. It cannot be further non loss decomposed (join dependency).

Example – Consider the above schema, with a case as “if a company makes a product and an agent is an agent for that company, then he always sells that product for the company”. Under these circumstances, the ACP table is shown as:

Table ACP

Agent	Company	Product
A1	Parle	BIscuit
A1	Parle	BIscuit
A1	Dea	FENTANYL
A1	Dea	FENTANYL
A2	Parle	BIscuit

The relation ACP is again decomposed into 3 relations. Now, the natural Join of all three relations will be shown as:

Table R1

Agent	Company
A1	Parle
A1	Dea
A2	Parle

Table R2

Agent	Product
A1	BIscuit
A1	FENTANYL
A2	BIscuit

Table R3

Company	Product
Parle	BIscuit
Parle	FENTANYL
Dea	BIscuit
Dea	FENTANYL

The result of the Natural Join of R1 and R3 over ‘Company’ and then the Natural Join of R13 and R2 over ‘Agent And ‘Product’ will be Table ACP.

Hence, in this example, all the redundancies are eliminated, and the decomposition of ACP is a lossless join decomposition. Therefore, the relation is in 5NF as it does not violate the property of lossless join.

Summary

1NF: It is known as the first normal form and is the simplest type of normalization that you can implement in a database. A table to be in its first normal form should satisfy the following conditions:

- Every column must have a single value and should be atomic.
- Duplicate columns from the same table should be removed.
- Separate tables should be created for each group of related data and each row should be identified with a unique column.

2NF: It is known as the second normal form. A table to be in its second normal form should satisfy the following conditions:

- The table should be in its 1NF i.e. satisfy all the conditions of 1NF.
- Every non-prime attribute of the table should be fully functionally dependent on the primary key i.e. every non-key attribute should be dependent on the primary key in such a way that if any key element is deleted then even the non_key element will be saved in the database.

3NF: It is known as the third normal form. A table to be in its third normal form should satisfy the following conditions:

- The table should be in its 2NF i.e. satisfy all the conditions of 2NF.
- There is no transitive functional dependency of one attribute on any attribute in the same table.

BCNF: BCNF stands for Boyce-Codd Normal Form and is an advanced form of 3NF. It is also referred to as 3.5NF for the same reason. A table to be in its BCNF normal form should satisfy the following conditions:

- The table should be in its 3NF i.e. satisfy all the conditions of 3NF.
- For every functional dependency of any attribute A on B (A->B), A should be the super key of the table. It simply implies that A can’t be a non-prime attribute if B is a prime attribute.

SOURCE

- College notes / pdfs
- https://www.geeksforgeeks.org/
- https://github.com/riti2409/DBMS_SQL-Notes?tab=readme-ov-file

BY AYUSH RAJ SINGH

(No gpt were harmed or misused in this doc)