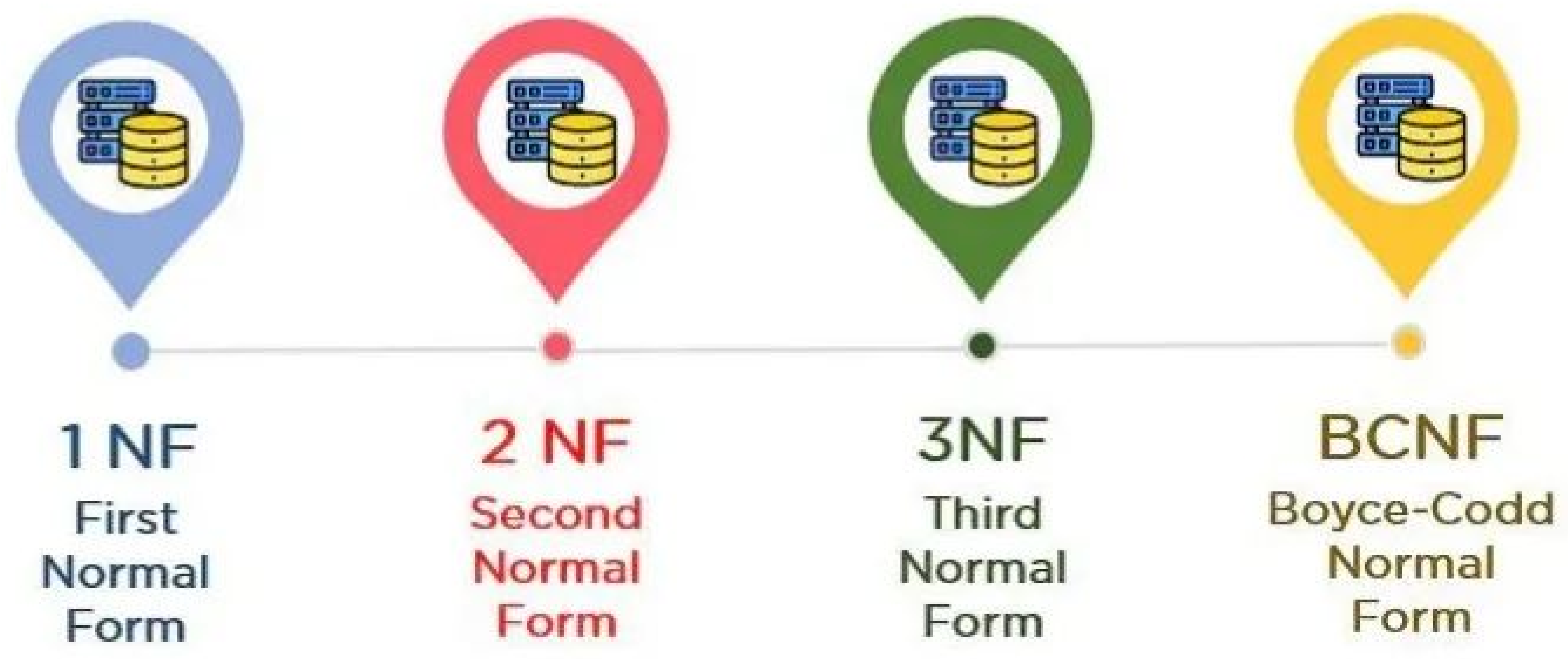
What is Normalization in SQL?

Normalization is the process to eliminate data redundancy (repetition) and enhance data integrity (consistency and accuracy) in the table.

It is a multi-step process that sets the data into tabular form and removes the duplicated data from the relational tables.

Normalization of a Database table is achieved by following a set of rules called "Forms" or "Normal forms" in creating the database table.



1st Normal Form (1NF)

- A table is referred to as being in its First Normal Form if atomicity of the table is 1.
- Here, atomicity states that a single cell cannot hold multiple values. It must hold only a single-valued attribute.

Id	Name	Course	Age
101	Alex	C / C++	21
102	Rohan	Java	20
103	Ritu	Python	19
104	Sam	C / C++	22

Id	Name	Course	Age
101	Alex	C	21
101	Alex	C++	21
102	Rohan	Java	20
103	Ritu	Python	19
104	Sam	C	22
104	Sam	C++	22

Second Normal Form (2NF)

- The entity should be considered already in 1NF.
- The table should not possess partial dependency. The partial dependency here means the proper subset of the candidate key should give a non-prime attribute.

cust_Id	store_id	store_loc
1	D1	Delhi
2	D3	Noida
3	T1	Pune
4	F2	Mumbai
5	H3	Gujrat

The above table possesses a composite primary key cust_id, store_id. The non-key attribute is store_loc. In this case, store_loc only depends on store_id, which is a part of the primary key. Hence, this table does not fulfill the second normal form.

To bring the table to Second Normal Form, you need to split the table into two parts.

cust_Id	store_id
1	D1
2	D3
3	T1
4	F2
5	H3

store_id	store_loc
D1	Delhi
D3	Noida
T1	Pune
F2	Mumbai
H3	Gujrat

As you have removed the partial functional dependency from the table, the column store_loc entirely depends on the primary key of that table, store_id.

Third Normal Form (3NF)

- Rule 1- Be in 2NF
- Rule 2- Has no transitive functional dependencies for non prime attributes.

Transitive dependency is a functional dependency in which A → C (A determines C) indirectly, because of A → B and B → C (where it is not the case that B → A).

stu_Id	name	sub_id	sub	address
1	Arun	11	SQL	Delhi
2	Varun	12	Java	Noida
3	Harsh	13	C++	Pune
4	Keshav	12	Java	Mumbai

stu_Id	name	sub_id	address
1	Arun	11	Delhi
2	Varun	12	Noida
3	Harsh	13	Pune
4	Keshav	12	Mumbai

sub_Id	sub
11	SQL
12	Java
13	C++

Boyce CoddNormal Form (BCNF)

- Rule 1- Be in 3NF
  - Rule 2- Every Right-Hand Side (RHS) attribute of the functional dependencies should depend on the super key of that particular table.

For example :
You have a functional dependency X → Y. In the particular functional dependency, X has to be the part of the super key of the provided table.

- Even when a database is in 3rd Normal Form, still there would be anomalies resulted if it has more than one Candidate Key.
- Sometimes is BCNF is also referred as 3.5 Normal Form.

stu_Id	subject	Professor
1	SQL	prof. Mishra
2	Java	prof. Anand
2	C++	prof. Kanth
3	Java	prof. James
4	Python	prof. Lokesh

prof_Id	subject	Professor
101	SQL	prof. Mishra
102	Java	prof. Anand
103	C++	prof. Kanth
104	Java	prof. James
105	Python	prof. Lokesh

stu_Id	prof_Id
1	101
2	102
2	103
3	104
4	105

@dynamic.coding

Thank you!

Did you find
this post helpful?

Follow us for more
related content and
projects.