

Schema Refinement (or) Normalization

Why normalization is required?

In the case of duplicated data in multiple tables (or) single table they can decompose or divide the tables into multiple table until duplications is removed

Problems with redundancy

(i) Wastage of memory and storage

(ii) It increases access time

(iii) Insert Anomaly

To insert the data in one table it will effect the same duplicated data in other tables

(iv) Delete Anomaly

To Delete the Data from one table it requires to perform deletion (or) other tables where duplication of data existed else it leads to inconsistency data

(v) Update Anomaly

To update the data in one table the data is perfected on other tables having old values which is resulting inconsistency data

Decomposition

Divide the table into multiple table to restore into the duplications but it is also resolves the incorrect data

Normalization

It is the process of decomposing the tables into multiple tables by refining the schema using diff

There are ⁴ practical Normal forms

1NF, 2NF, 3NF, BCNF

1NF :- It is used to eliminate multiple values (or) composite attributes

A relation is in 1st ^{normal} order form it must satisfy all the attributes have single or atomic value

std id	course
1.	C, Java
2.	Python
3.	C++

std id	course
1	C
1	Java
2	Python
3	C++

2NF :- A relation to be in 2NF it must satisfy the following rules

(i) A relation must be in 1st NF

(ii) It mustn't contain any partial dependencies that is all non prime attributes are fully functional dependent on primary key

$x \rightarrow y$

By this functionally dependent on x where x is a primary key

In case of partial dependency i.e., some non prime attributes are dependent on other key apart from primary key.

This is to be eliminated by decomposing (or) dividing the table into multiple tables.

Primary Key	std id	Sname	course id	cname
	1	x	10	xx
	2	y	20	yy

\Rightarrow Partial FD

stdid	sname

courseid	cname

fully FD

3NF :- In 3NF it is used to eliminate transitive dependency that is $x \rightarrow y, y \rightarrow z, x \rightarrow z$

A relation to be in 3NF it must satisfy the following rules

(i) The relation must be in 2NF

(ii) No non prime attribute is transitively depend on primary key

(iii) for each functional dependency $x \rightarrow z$ atleast one of the conditions hold :

* x is a super key of table minimal no. of attribute of

* z is prime attribute of table

* .

Empcode	Ename	Empcity	pincode



Empcode	Ename	Empcity

Pincode	Empcity

Transitive dependency relation to be converted into 3NF by satisfying the rules the process of decomposing the table into multiple table