2/22

Saturday 01
Day 001-364

Normalization

→ Normalization is a technique which is been used to remove the redundancy (or we can say the duplicates) from a table.

→ If we are not able to remove the duplicacy completely from a table, so it gives us the feasibility to reduce that duplicacy to an extent.

* There are basically two types of duplicacy in a table →

(1) Row level
(2) Column level

Sunday 02

So, lets start with row level duplicacy, in which we will take data for a student →

03 Day 003-362

Primary key

Ex →

SID	Sname	Age
1	Rutvi	22
2	Ram	20
3	Ravi	19
1	Rutvi	22

So, in the above table we can see that the data of row 1 and row 4 have the exact same values. i.e., duplicacy.

So, in order to remove this row duplicacy we have a concept of "Primary key", in which we set any one attribute of our table as Primary key.

So, here we can make 'SId' as primary key.

and Properties of Primary Key are that it should be - (Unique + Not Null)

(Unique → values won't be repeated) + (Not Null → neither we can leave this empty)

So, this is how it will take care of the row level duplicacy.

*Now, let's talk about Column level duplicacy.

Ex→ [SID, Sname, Cid, Cname, Fid, Fname, Salary] Columns of our table.

Where, sid → student id
Sname → Student name
Cid → Course id
Cname → Course name
Fid → Faculty id
Fname → Faculty name
Salary → Salary

05 Day 005-360

SID	Sname	CID	Cname	Fid	Fname	Salary
1	Rutvi	C1	DBMS	F1	John	30000
2	Ram	C2	JAVA	F2	Bob	30000
3	Ravi	C1	DBMS	F1	John	30000
4	Monal	C1	DBMS	F1	John	30000

- As, it is quite evident from the table above that there is no row duplicacy, since we have already removed it, by setting 'SID' as our Primary key.
- But, we can see that there are colums which are quite similar to each other.
- Now, as we know that when the data is been added to a table it is # in thousands, lakhs, and what

if there are columns with similar data, how does it even create a problem?

• So, their are 3 types of Problems which can arise in case of column duplicacy.

(a) Insertion Anomaly
(b) Deletion Anomaly
(c) Updation Anomaly

(These three are the operations used in database)

[Anomaly is basically a type of problem, but a problem which arises at some special occassions.]

• So, let's begin with insertion anomaly →

Suppose, I want to enter the data of a new student in my database, it will done easily.

Eg →

SID	Sname
1	
⋮	⋮
5	Varun

07 Day 007-358

Now, suppose their was a new course introduced in the university whose,

Sid	Sname	CourseID	Cname
		C10	computer

So, in here this data can't be directly inserted into the table since, we haven't talked about SId and Sname.

When their is no specific SID for this where this will be inserted, and since SID is a primary key we can't keep it null and it must be unique as well.

- So, in case of deletion anomly, simply we are asked to delete the data for a student whose roll no (SID) is '1'.

So, by simply writing a query for deletion we can do that →

2022 January • Saturday 08 Day 008-357

(Table name)

"Delete from Student where SID = 1 ;"

(Column name)

So, this will completely remove the whole row where SID = 1.

So, now in case of updation anomly suppose I want to update Sname from Monal to Monal Thakkar where SID is 4.

So the query for this would be →

"Update Student set Sname = "Monal Thakkar" where SID = 4."

This would work fine for the updation.

Now, lets take another scenario for updation.

Sunday 09

Suppose we wont to update the salary of the faculty whose FID = F1

We wont to change it from 30,000 to 40,000.

M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	FEB 2022
														14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	•	•	•	•	•	•	

So, now when try to update the salary all the values of FID, whose value is f_1 will be updated.

And this is not exactly what we wanted

So, now the solution for this is →

Divide the entire table into three/multiple different tables →

I st

SID	Sname

This is Primary key

II nd → This is P.K.

CID	Cname

III rd → This is PK

FID	Fname	Salary

So, now everything becomes easy with this, new course can be launched and

inserted into table IInd without affecting the data of other tables.

Similarly, if we delete a student's record the other records like for course and faculty still remains unaffected.

And same happens at the time of updation since FID is the Primary key in Table IIIrd, no entries would be repeated and salary can be easily changed.