Normalization in DBMS

H_OD Name, HOD-Room
H_OD Mobile

Sid	SName	Credits	Dept-Name	Building	Room No
1	Abir	5	CSE	B1	101
2	Rasel	8	CSE	B1	101
3	Sohel	9	EEE	B2	201
4	Arif	8	EEE	B2	202
5	Sazzad	7	Civil	B1	110
6	Mamun	7	ECE	B1	115
7	Mehedi	8	Civil	B1	110
8	Sohan	7	CSE	B1	101
			ME	B1	120

When we are inserting a student information, we repeat Dept-Name, Building, RoomNo. This is lead a redundancy in DBMS. Because we're taking $\boxed{\text{larger Schema}}$. We are talking too much information in one table.

if we take smaller table may be redundancy can be removed.

Redundancy is main or root cause of all the problems.

Data Anomalies

There are threee main problem due to redundancy.

(I) Insertion Anomalies
(II) Updation "
(III) Deletion "

When data is having multiple Copies, and one place we update data in one place but we forget to update that same data in another place and importantly we can't say which data is Correct.

Insertion Anomalies

Suppose we want to insert department Name ME. where Building No is B1 and Rom No is 120. In this case we can't insert these information. Because until one Student enroll in that department we can't insert because Sid is primary key it can't be null.

* Null Null Null ME B1 120

We are taking all information in one table and these anomalies occur.

defn [" When you are inserting information, you've to insert extra information and actually you don't have those extra information]

(II) Updateion Anamalies

Suppose We want to update information of CSE department. CSE dept. has been Shifted from B1 → C1 and Room 101 → 301.

Then We have to update all the tuple or information Where the Dept. is CSE. Why?

Suppose CSE dept has 150 Students. We have to update all the information where dept is CSE.

This is Why redundancy occurs.

[And the main problem is We updated 100 times and forget to update rest of tuples, then We failed to specify which one is Correct!]

Data Inconsistency problem occurs.

(III) Deletion Anomalies

Suppose We want to delete information of Mamun whose id is 7. Then all the data of mamun has been deleted. Suppose, EEE department has one Student, and he might have left the College / University than Mamun as well department will be deleted

[Student may have Shifted or left from University but you can't delete the department information]

Now A question arisen, How can we Solve all these redundancy ?
First we have to decompose the table into two tables

Sid	SName	Credit	Dept. Name FK
1	Abir	5	CSE
2	Rasel	8	CSE
3	Sohel	9	EEE
4	Arif	8	EEE
5	Sazzad	7	Civil
6	Mamun	7	ECE
7	Mehedi	8	Civil
8	Sohan	7	CSE

Student

PK

Dept_Name	Building	Room_No
CSE	B1	101
Civil	B1	110
EEE	B2	~~200~~ 201
ECE	B1	115

Dept

lets see how these redundancy can be removed!
In dept table, the information of CSE dept is stored once while in previous table we stored in multiple times.

When we are separating two tables from larger table or schema, there must have a common attribute $\boxed{\text{dept\_name}}$ to establish relation.

Now we can easily insert (ME, B1, 120). No need extra information what is required in previous.

Insert [When we are updating CSE department building no and Room No, only one tuple is department while in previous case all the tuples have to update.

Delete [Suppose we want to delete information of Mamun and in previous case all the information of Mamun is deleted including dept name. Here it's simple!

Update [Here all the anomalies are removed by decomposing the tables into two small table.

This is the process of Normalization.

Normalization is the process of making the table free from insert, delete and Update anomalies and obviously save space by reducing the redundant or duplicate data.

- → Simplify queries → Searching, Sorting, Creating indexes will be faster.
- → Remove redundancy
- → Save Space
- → Minimize Null Values
- → Important for OLTP System
- → Simplify DB Structures

OLTP → Online Transaction processing

OLAP → Online Analytical Processing

Sometimes we require de-normalized data.

OLTP → require normalized data. should avoid day to day transaction. insert/delete/update anamalies.

OLAP → Data Warehouse [Huge data] complex queries may require one table rather separating multiple tables.

Decomposition is not so easy!

Lossy } Decomposition
Lossless