Topic : Normalization:

YouTube video link:

https://www.youtube.com/watch?v=UrYLYV7WSHM

Summary:

Through this class I learnt the normalization techniques of database management. Database normalization is a method of organizing the structure of database. This entails building tables and establishing the relationship between these tables in accordance with the principles to protect the data and make the database more flexible by eliminating data redundancy and inconsistent dependency.

Types of Normalization:

1NF (1st Normal Form):

A relation is said to be in 1NF if it contains atomic value. This Normal Form addresses the atomicity issue. Atomicity here denotes that the values in the table shouldn't be subdivided further. In simple terms, one cell cannot contain more than one value. A table can violate the First Normal Form if it has a composite or multivalued attribute.

Employee id	Employee name	Phone number	Salary
1EDU001	Harry	+919876543210 +918632456820	65000
1EDU002	Alia	+919865327410	74000
1EDU003	John	+919632587310	51000

In the above table, we can notice that the Phone Number column has two values. Therefore, it broke the first NF. Now, the outcome of applying the first NF on the above table is seen in the table below.

Employee id	Employee name	Phone number	Salary
1EDU001	Harry	+919876543210	65000
1EDU001	Harry	+918632456820	65000
1EDU002	Alia	+919865327410	74000
1EDU003	John	+919632587310	51000

By this, atomicity is achieved and also each and every column have unique values.

2NF (2nd Normal Form):

The table must appear in the first NF as the first requirement for the relation to in the second NF. Partial dependencies should not be present in the table either. In this case, partial dependency means that a non-prime attribute is determined by the right subset of the candidate key. Let's have a look at the example below to help you understand.

Employee id	Department id	Office Location

1EDU001	ED-T1	Pune
1EDU002	ED-S2	Bangalore
1EDU003	ED-M1	Delhi
1EDU004	ED-T3	Mumbai

An employee ID and department ID composite primary key is present in this table. The office location characteristic is a non-key one. In this instance, the only component of the primary key on which Office Location depends is Department ID. As a result, the second Normal Form is not satisfied by this table.

To bring this table to Second Normal Form, we need to break the table into two parts.

Department id	Office Location
ED-T1	Pune
ED-S2	Bengaluru
ED-M1	Delhi
ED-T3	Mumbai

We can see, we removed the partial functional dependency that we initially had. Now, in the table, the column Office Location is fully dependent on the primary key of that table, which is Department ID.

3NF (3rd Normal Form):

For the relation to be in 3NF, the table has to be in 2NF before proceeding to 2NF. The absence of transitive reliance for non-prime properties is the second requirement. Accordingly, non-prime qualities in a particular table that do not serve as a candidate key shouldn't be dependent on other non-prime attributes. Therefore, a transitive dependency is a functional dependency in which X Z (X causes Z) indirectly, because of X Y and Y Z (where it is not true that Y X).

For instance,

Student id	Student name	Subject id	Subject	Address
1DT15ENG1	Harry	15CS6	MySQL	Goa
1DT15ENG2	Alia	15CS7	Python	Bangalore
1DT15ENG3	John	15CS8	Java	Delhi
1DT15ENG4	Alex	15CS9	C	Kochi

Student ID determines Subject ID in the table above, and Subject ID decides Subject. So, using Subject ID, Student ID determines Subject. As a result, the third normal form is not satisfied by this structure, which suggests that we have a transitive functional dependency.