Disclaimer

This content is just supportive material for Normalization. various textbooks as well as freely available material from internet sources were consulted for preparing this content. This content may not cover all aspects of learning Normalization, nor these be taken as primary source of information. Students are encouraged to follow the core text books and reference book prescribed by the university.

Example 1

Normalize the given table upto 3NF.

OR

Explain the concept of Normalization by taking any table and Normalize the given table upto 3NF.

EMPLOYEE_ID	NAME	JOB_CODE	JOB	STATE_CODE	HOME_STATE
E001	Alice	J01	Chef	26	Michigan
E001	Alice	J02	Waiter	26	Michigan
E002	Bob	J02	Waiter	56	Wyoming
E002	Bob	J03	Bartender	56	Wyoming
E003	Alice	J01	Chef	56	Wyoming

For 1 NF

A relation is in 1NF if all the contained values should be atomic values.

In above relation all entries are atomic, so it is already in 1NF.

For 2 NF

A relation will be in 2NF if

- ✓ It is in 1NF and
- ✓ All non-prime attributes are fully functional dependent on the key attribute or primary key (There must not be any partial dependency)

Here we {(EMPLOYEE_ID+ JOB_CODE)} considered as composite primary key . Due to these combination we can uniquely identify all the records of the table.

But, here partial dependency exists. All attributes are nor fully functional dependent on primary key.

For example:

EMPLOYEE_ID →NAME

JOB_CODE →JOB

Now, we can remove this partial dependency as follows.

Employee_roles table

EMPLOYEE_ID	JOB_CODE
E001	J01
E001	J02
E002	J02
E002	J03
E003	J01

Employees Table

EMPLOYEE_ID	NAME	STATE_CODE	HOME_STATE
E001	Alice	26	Michigan
E002	Bob	56	Wyoming
E003	Alice	56	Wyoming

Jobs table

JOB_CODE	JOB
J01	Chef
J02	Waiter
J03	Bartender

For 3NF

For any relation to be in 3NF it must satisfied following properties.

- ✓ It is in 2NF.
- ✓ Every non-prime attribute is non-transitively dependent on the primary key. Which means there should not be case that non-prime attribute is functionally dependent on another non-prime attribute.

Finding all functional dependencies in **Employees Table**

EMPLOYEE_ID→STATE_CODE

STATE_CODE→HOME_STATE

Here we find one trival functional dependency.

EMPLOYEE_ID→STATE_CODE →HOME_STATE

So, we have to remove this

Here, **HOME_STATE** which is non-prime attribute depends on another non-prime attribute **STATE_CODE.**

Employee_roles Table

EMPLOYEE_ID	JOB_CODE
E001	J01
E001	J02
E002	J02
E002	J03
E003	J01

Employees Table

EMPLOYEE_ID	NAME	STATE_CODE
E001	Alice	26
E002	Bob	56
E003	Alice	56

Jobs Table

JOB_CODE	JOB
J01	Chef
J02	Waiter
J03	Bartender

states Table

STATE_CODE	HOME_STATE
26	Michigan
56	Wyoming

Now our database is in 3NF.

How will you make a given table std_master with attributes: st_id, st_name, instructor_id, inst_name, course_id1, course_name1, course_id2, course_name2, course_id3, course_name3 in 1st ,2nd and 3rd Normal forms, Write the steps. [PU:2011 fall]

Solution:

UNF

If the relation is in unnormalized form i.e it contains one or more repeating groups, or each row may contain multiple set of values for some columns.

1NF

To take the table into 1NF following criteria must be satisfied.

- ✓ If there are no repeating values in a column (All entries must be atomic)
- ✓ It must not have repeating columns.

Here, course_id1,course_id2,course_id3 are repeating columns, changing it to only course_id and similarly ,course_name1,course_name2,course_name3 into course_name only then,

the 1NF table relation schema will be

stdmaster(st_id,st_name,instructor_id,inst_name,course_id,course_name)

Now this is in 1NF.

2NF

A relation will be in 2NF if

- ✓ It is in 1NF and
- ✓ All non-prime attributes are fully functional dependent on the key attribute or primary key (There must not be any partial dependency)

But here partial dependency exists. All attributes are not fully functional dependent on primary key.

For example:

instructor_id →inst_name

course_id→course_name

Now, to remove these partial dependencies we can decompose the table as follows:

student(st_id,st_name)
instructor(instructor_id,inst_name)
course(course_id,course_name)
teaching_info (st_id, course_id, instructor_id)

3NF

For any relation to be in 3NF it must satisfied following properties.

- ✓ It is in 2NF.
- ✓ Every non-prime attribute is non-transitively dependent on the primary key. Which means there should not be case that non-prime attribute is functionally dependent on another non-prime attribute.

Here, all the above tables are already in 3NF.