Course Name: **A Level (1st Sem)** Subject : **Introduction to DBMS**

Topic: **DB Normalization – Exercise Practice on 2NF (Part 5)** Date: **19-May-2020**

Database Normalization – Exercise practices on 2NF

Que 1:

Suppose a following stu_proj relational schema:

stu_id	proj_id	stu_name	proj_name	proj_lang_used
s101	p001	Rakesh	Online Chatting	python
s102	p001	Kritika	Online Chatting	python
s102	p002	Kritika	Text Editor	Java
s103	p002	Mahesh	Text Editor	Java
s104	p002	Ram	Text Editor	Java
s104	p003	Ram	Online Shopping	PHP

Identify functional dependencies in above table and find out that relation is in 2NF or not? If not decompose it in 2NF.

(A student can work on many projects and a project can have many students associated with it.)

Solution:

The table is in 1NF because each attribute in the table have atomic (single) value.

The following FDs are identified based on the value and descriptions given about table:

stu_id → **stu_name** (Student name can be determined by student id)

proj_id → **proj_name, proj_lang_used** (Project name and language used can be determined by project id)

Since (stu_id, proj_id) uniquely identifies each record in the table, and the closure of (stu_id, proj_id) also has all the attributes of table.

So, **(stu_id, proj_id) is the candidate key** in table.

Prime attributes - stu_id, proj_id (because these are the part of the candidate key)

Non prime attributes - stu_name, proj_name, proj_lang_used (because these are not part of the candidate key)

Now, the partial dependency exists in the table, it means violation of rule of 2NF because non prime attribute is dependent on part of the candidate key.

stu_id → stu_name	both FD violating rule of 2NF, since
proj_id → proj_name, proj_lang_used	(stu_id, proj_id) is the candidate key)

Therefore, to convert the relation in 2NF, It is divided into three relations:

student (stu_id, stu_name)	Since stu_id → stu_name
project (proj_id, proj_name, proj_lang_use(	Since proj_id → proj_name, pro_lang_used
stu_proj_alloc (stu_id, proj_id)	Since stu_id, proj_id is candidate key

Above three tables (student, project, stu_proj_alloc) are following the rules of 2NF.

See, how the data redundancy is minimized after converting table into 2NF (compare it with original table):

table1: **student**

stu_id	stu_name
s101	Rakesh
s102	Kritika
s103	Mahesh
s104	Ram

table2: **project**

proj_id	proj_name	proj_lang_used
p001	Online Chatting	python
p002	Text Editor	Java
p003	Online Shopping	PHP

table3: **stu_proj_alloc**

stu_id	proj_id
s101	p001
s102	p001
s102	p002
s103	p002
s104	p002
s104	p003

Exercise:

1. Suppose a relational schema R (A B C), and

   FDs: AB → C

   B → C

Check out the relation R is in 2NF or not? If not decompose it in 2NF.