

DBMS Lab Assignment 6

B M Kalpajeet

19BCS117

Team 4

1. Convert the table to 1NF

i)

Id	Name	Age	Location	Course
1	Sachin	22	Delhi	OS, DBMS
2	Ram	22	Jamshedpur	DAA, DBMS
3	Mike	23	Chennai	ML, OS
4	Sameer	21	Bengaluru	DAA, ML
5	Vijay	22	Mumbai	ML,DSMS

Answer:

For 1NF:

- 1) All attributes should have atomic values.
- 2) All columns must have unique names.
- 3) All values must be from the same domain.
- 4) Order of data does not matter.

In the above table, all attributes are unique, and the values for an attribute are from the same domain. However, is it not in 1NF as the Course attribute has more than one value for a single row. As per 1NF, they need to be atomic values, and so we must convert it. As it is not in 1NF, it is also not in 2NF and 3NF.

Primary key: Id

Candidate key: (Id, Name)

Prime attributes: Id, Name

Non-prime attributes: Location, Age, Courses

Converted 1NF table:

Id	Name	Age	Location	Course
1	Sachin	22	Delhi	OS
1	Sachin	22	Delhi	DBMS
2	Ram	22	Jamshedpur	DAA
2	Ram	22	Jamshedpur	DBMS
3	Mike	23	Chennai	ML
3	Mike	23	Chennai	OS
4	Sameer	21	Bengaluru	DAA
4	Sameer	21	Bengaluru	ML
5	Vijay	22	Mumbai	ML
5	Vijay	22	Mumbai	DBMS

Dependency:

Functional dependency: Id -> Name, Id -> Location, Id -> Age

ii)

ID	Name	Phone	State	Country
1	Kailley	9716245698	Karnataka	INDIA
2	Janet	9876543261	Maharashtra	INDIA
3	Robert	9456735678	Andra Pradesh	INDIA
4	Thomas	9966744381	Kerala	INDIA

Answer:

This table is already in 1NF as the attributes are unique, atomic and from the same respective domain.

Primary key: ID

Candidate key: (ID, Name)

Prime attributes: ID, Name

Non-prime attributes: Phone, State, Country

2. Convert the table to 2NF

i)

Emp_ID	Duty_shift_ID	Name	Age	Duty_shift
101	1	Arun	26	Morning
102	2	Bobby	28	Afternoon
103	3	Suresh	32	Night
104	1	Sita	24	Morning

Answer:

For 2NF,

- 1) It should be in 1NF.
- 2) It should not have any partial dependency (an attribute will depend on only a part of the primary key and not on the whole key)

The above table is not in 2NF as it is not totally functionally independent. We can see that:

Emp_ID -> Name, Empd_ID -> Age

Emp_ID, Duty_shift_ID -> Duty_shift

Since there is partial dependency, we need to remove it, and convert into 2NF.
Since it is not in 2NF, hence it is not in 3NF.

Primary key: Emp_ID

Candidate key: (Emp_ID, Duty_shift_ID)

Prime attributes: Emp_ID, Duty_shift_ID

Non-prime attributes: Name, Age, Duty_shift

Converted 2NF tables:

i) For Emp_ID

Emp_ID	Name	Age
101	Arun	26
102	Bobby	28
103	Suresh	32
104	Sita	24

ii) For Duty_shift_ID

Emp_ID	Duty_shift_ID	Duty_shift
101	1	Morning
102	2	Afternoon
103	3	Night
104	1	Morning

ii)

Emp_ID	Project_ID	Name	Proj_Name	No_of_hours
123	Prj_21	Ajay	Speech_system	10
321	Prj_45	Charu	HR System	15
546	Prj_24	Rajesh	Automate Tickets	23
765	Prj_11	Abhishek	NLP	16

Answer:

This table is in 2NF as it already is in 1NF and no partial dependency. Non-prime key attribute is fully functionally dependent on candidate key.

Primary key: Emp_ID

Candidate key: Emp_ID

Prime attributes: Emp_ID

Non-prime attributes: Project_ID, Name, Proj_Name, No_of_hours

If there was any same values in Emp_ID and Project_ID, then it won't be in 2NF and the following will be applicable:

This table is not in 2NF, as there is partial dependency between Proj_Name and Project_ID. We can see that:

Emp_ID -> Name

Emp_ID, Project_ID -> Proj_Name, No_of_hours

Since there is partial dependency, we need to remove it, and convert into 2NF. Since it is not in 2NF, hence it is not in 3NF.

Primary key: Emp_ID

Candidate key: (Emp_ID, Project_ID)

Prime attributes: Emp_ID, Project_ID

Non-prime attributes: Name, Proj_Name, No_of_hours

Converted 2NF tables:

i) For Emp_ID

Emp_ID (PK)	Name
123	Ajay
321	Charu
546	Rajesh
765	Abhishek

ii) For Project_ID

Emp_ID	Project_ID (PK)	Proj_Name	No_of_hours
123	Prj_21	Speech_system	10
321	Prj_45	HR system	15
546	Prj_24	Automate Tickets	23
765	Prj_11	NLP	16

3. Convert the table to 3NF

i)

Cust_ID	Cust_name	Cust_postcode	Cust_address	Cust_loc
25	Dell	560037	Whitefield	Bangalore
45	Lenovo	560046	Marathahalli	Bangalore
89	Acer	210067	Bandra	Mumbai
90	Samsung	4500078	Delhi Central	Delhi

Answer:

For 3NF,

- 1) It should be in 2NF.
- 2) There should not be any transitive dependency (a non-prime attribute depends on other non-prime attributes instead of depending on the prime attributes or primary key)

We can see that:

Cust_ID -> Cust_Loc -> Cust_Address

(As location depends on the address and postcode of the place)

Primary key: Cust_ID

Candidate key: (Cust_ID, Cust_Name), Cust_postcode, Cust_address

Prime attributes: (Cust_ID, Cust_Name), Cust_postcode, Cust_address

Non-prime attributes: Cust_loc

Converted 3NF tables:

i) For Cust_ID

Cust_ID (PK)	Cust_name	Cust_loc	Cust_address
25	Dell	Bangalore	Whitefield
45	Lenovo	Bangalore	Marathalli
89	Acer	Mumbai	Bandra
90	Samsung	Delhi	Delhi Central

ii) For Cust_postcode

Cust_address	Cust_postcode
Whitefield	560037
Marathalli	560046
Bandra	210067
Delhi-Central	4500078

ii)

Building	Contractor	Builder	Fee
B_2156	Taylor	Prestige	2567891
B_8765	Sandeep	Hiranandani	3567356
B_4567	vishaka	Tata	4567990

Answer:

This table is already in 3NF, as it is in 2NF and has no transitive dependencies.

Primary key: Building

Candidate key: Building

Prime attributes: Building

Non-prime attributes: Contractor, Builder, Fee