Let us assume a table User\_Personal as given below;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***UserID*** | **U\_email** | **Fname** | **Lname** | **City** | **State** | **Zip** |
| *MA12* | Mani@ymail.com | MANISH | JAIN | BILASPUR | CHATISGARH | 458991 |
| *PO45* | Pooja.g@gmail.co | POOJA | MAGG | KACCH | GUJRAT | 832212 |
| *LA33* | Lavle98@jj.com | LAVLEEN | DHALLA | RAIPUR | CHATISGARH | 853578 |
| *CH99* | Cheki9j@ih.com | CHIMAL | BEDI | TRICHY | TAMIL NADU | 632011 |
| *DA74* | Danu58@g.com | DANY | JAMES | TRICHY | TAMIL NADU | 645018 |

* **Is this table in First Normal Form?**

**Yes.** All the attributes contain only atomic values.

* **Is this table in Second Normal Form?**

To verify this property, we need to find all the functional dependencies which are holding in User\_Personal table, and have to identify a Primary key.

Let us do that by using the sample data. This leads to the following set of FDs;

F = { UserID → U\_email Fname Lname City State Zip,   
Zip → City State }

As UserID attribute can uniquely determine all the other attributes, we can have UserID as the Primary key for User\_Personal table.

The next step is to check for the 2NF properties;

***Property 1 – The table should be in 1NF.***

***Property 2 – There should not be any partial key dependencies.***

**Our table is in 1NF, hence property 1 is holding.**

Primary key of our table is UserID and UserID is single simple attribute. **As the key is not composite, there is no chance for partial key dependency to hold.** Hence property 2 is also holding.

User\_Personal table is in 2NF.

* **Is User\_Personal in 3NF?**

To verify this we need to check the 3NF properties;

***Property 1 – Table should be in 2NF.***

***Property 2 – There should not be any Transitive Dependencies in the table.***

**Table User\_Personal is in 2NF, hence property 1 is satisfied.**

User\_Personal table holds the following Transitive dependency;

UserID → Zip, Zip → City State

**Hence, property 2 is not satisfied and the table is not in 3NF.**

**Solution:**

Decompose User\_Personal. For this, we can use the functional dependencies Zip → City State and UserID → U\_email Fname Lname City State Zip.

As a result, we can have the following tables (primary keys are underlined);

User\_Personal (UserID, U\_email, Fname, Lname, Zip)

City (Zip, City, State)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***UserID*** | **U\_email** | **Fname** | **Lname** | **Zip** |
| *MA12* | Mani@ymail.com | MANISH | JAIN | 458991 |
| *PO45* | Pooja.g@gmail.co | POOJA | MAGG | 832212 |
| *LA33* | Lavle98@jj.com | LAVLEEN | DHALLA | 853578 |
| *CH99* | Cheki9j@ih.com | CHIMAL | BEDI | 632011 |
| *DA74* | Danu58@g.com | DANY | JAMES | 645018 |

**Table - User\_Personal**

|  |  |  |
| --- | --- | --- |
| **Zip** | **City** | **State** |
| 458991 | BILASPUR | CHATISGARH |
| 832212 | KACCH | GUJRAT |
| 853578 | RAIPUR | CHATISGARH |
| 632011 | TRICHY | TAMIL NADU |
| 645018 | TRICHY | TAMIL NADU |

**Table – City**

**Both tables are in 3NF.**

**Hence, tables are normalized to Third Normal Form.**