

# Team Everyone



**MD  
RUFSANJANI  
SHANTO**  
221-35-1064



**MD MAHFUZ  
RAHMAN  
NIROB**  
221-35-981



**PIJUS  
CHANDRO  
ROY**  
221-35-1050



**SASONJOY  
CHANDRA  
COACH**  
221-35-841



**AHIDUL  
ISLAM**  
221-35-931





**MD SUHAG ALI**  
**LECTURER**  
**DEPT. OF SOFTWARE ENGINEERING**  
**DAFFODIL INTERNATIONAL**  
**UNIVERSITY**

**SUBMITTED BY**  
**MD SUHAG ALI**

# FIRST NORMAL FORM (1NF)



## WHAT IS FIRST NORMAL FORM (1NF)

**The official qualifications for 1NF are:**

- Each attribute must have a unique name.
- Each attribute must have a single value.
- Row cannot be duplicated.
- There is no repeating groups.

**Additional:**

- Choose a primary key. The primary key can be an attribute or combined attributes

## **Advantages of 1NF**

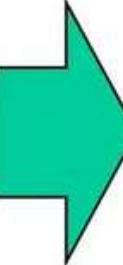
- Minimization of Data Redundancy: By having a unique primary key for each table and atomic values in each column, 1NF helps to eliminate data redundancy and ensure that data is stored only once. This helps to improve the efficiency of the database and reduce the risk of data inconsistencies.
- Improved Data Integrity: 1NF helps to ensure that data is accurate and consistent across all rows in the table. This helps to ensure that data is reliable and can be used effectively for reporting and analysis.
- Simplified Data Access and Updates: By adhering to the rules of 1NF, data is organized in a logical and consistent way, making it easy to access and update. This helps to improve the performance of the database and reduce the risk of errors.
- Ease of Maintenance: With a 1NF structure, it is easier to maintain the database and make changes as needed. It also makes it easier to identify errors and inconsistencies, which can be corrected quickly.

## Disadvantages of 1NF

- Increased Number of Tables: The strict rules of 1NF can lead to an increased number of tables in the database, which can make it more difficult to manage and maintain.
- Limited Flexibility: The strict rules of 1NF can make it difficult to adapt the database to changing requirements. For example, adding a new column to a table can require the creation of a new table, which can be a time-consuming and complex process.
- Increased Storage Space: 1NF can lead to increased storage space requirements, as data is stored only once, and the same data may be repeated in multiple tables.
- Limited scalability: 1NF databases can handle a limited amount of data and can become slow and inefficient when the data volume increases.

Name	DOB	Course		Payment
Sok	11/5/1990	IT		450 Dollars
Sao	4/4/1989	Mgt		400 Dollars
Chan	7/7/1991	IT	Mgt	IT: 450 Dollars Mgt: 400 Dollars
Sok	11/5/1990	Mgt		400 Dollars
Sao	4/4/1989	Tour		1) 200 Dollars 2) 200 Dollars

1. Each attribute has unique name -> Good
2. The *Payment* has multi data type (currency & string) -> Bad
3. All rows are not duplicated -> Good
4. The Course and Payment have repeating groups -> Bad



Name	DOB	Course	Payment (\$)
Sok	11/5/1990	IT	450
Sao	4/4/1989	Mgt	400
Chan	7/7/1991	IT	450
Chan	7/7/1991	Mgt	400
Sok	11/5/1990	Mgt	400
Sao	4/4/1989	Tour	200
Sao	4/4/1989	Tour	200

All correct? Not yet. Choose a primary key.

Name? No. Name has duplicated values.

Or DOB, or Course or Payment? No. Each one has duplicated values.

Name and DOB? No. They still have duplicated values.

Name and DOB and Course? No. Still duplicated.

Combine all attribute? Still no. The last two rows are duplicated.

So what else we can do?

Of course, there is a way. Add a new attribute to be a primary key. So let's call it **ID**.



ID	Name	DOB	Course	Payment
1	Sok	11/5/1990	IT	450
2	Sao	4/4/1989	Mgt	400
3	Chan	7/7/1991	IT	450
4	Chan	7/7/1991	Mgt	400
5	Sok	11/5/1990	Mgt	300
6	Sao	4/4/1989	Tour	200
7	Sao	4/4/1989	Tour	200

Now it is completely in 1NF.

*Next, check it if it is not in 2NF.*

**THANK YOU**

