Example for 1NF (First Normal Form)

|  |  |  |  |
| --- | --- | --- | --- |
| Emp no | First  Name | Salary | Degrees |
| 111 | Joe | 29,000 | BSc,MSc |
| 200 | April | 41,000 | BA,MA |
| 205 | Peter | 33,000 | BEng |

In this example, each column contains atomic values, and there are no repeating groups. This table is already in 1NF.

Example for 2NF (Second Normal Form)

|  |  |  |
| --- | --- | --- |
| STUDENT NO | COURSE NO | COURSE FEE |
| 1 | 1 | 1000 |
| 2 | 2 | 1500 |
| 1 | 1 | 2000 |
| 4 | 4 | 1000 |

Table 1(Student)

|  |  |
| --- | --- |
| STUDENT NO | COURSE NO |
| 1 | 1 |
| 2 | 2 |
| 1 | 1 |
| 4 | 4 |

Table 2(Course)

|  |  |
| --- | --- |
| COURSE NO | COURSE FEE |
| 1 | 1000 |
| 2 | 1500 |
| 1 | 2000 |
| 4 | 1000 |

In this example, each column contains atomic values, and there are no repeating groups. This table is already in 2NF.

Example for 3Nf (Third Normal Form)

|  |  |  |  |
| --- | --- | --- | --- |
| Book ID | Genre ID | Genre Type | Price |
| 1 | 1 | Gardening | 25 |
| 2 | 2 | Sports | 14 |
| 3 | 1 | Gardening | 10 |
| 4 | 3 | Travel | 12 |
| 5 | 2 | Sports | 17 |

In the table able, [Book ID] determines [Genre ID], and [Genre ID] determines [Genre Type]. Therefore, [Book ID] determines [Genre Type] via [Genre ID] and we have transitive functional dependency, and this structure does not satisfy third normal form.

TABLE\_BOOK

|  |  |  |
| --- | --- | --- |
| Book ID | Genre ID | Price |
| 1 | 1 | 25 |
| 2 | 2 | 14 |
| 3 | 1 | 10 |
| 4 | 3 | 12 |
| 5 | 2 | 17 |

**TABLE\_GENRE**

|  |  |
| --- | --- |
| Genre ID | Genre Type |
| 1 | Gardening |
| 2 | Sports |
| 3 | Travel |

**Example for ACID Properties:**

ACID properties are essential in database transactions. Let's say we have a banking system where a user transfers money from one account to another:

**Atomicity:**

Either the entire transaction of transferring money occurs, or none of it does. For example, if a user transfers $100 from Account A to Account B, either the full $100 is deducted from Account A and added to Account B, or no money is transferred at all.

**-Consistency:** The database remains in a consistent state before and after the transaction. For instance, after the money transfer, the total balance in both accounts must reflect the correct values.

**Isolation:** Transactions occur independently without interference from other transactions. For example, if another transaction is querying Account A while the money transfer transaction is ongoing, it should not see an inconsistent state where the money is deducted from Account A but not yet added to Account B.

**Durability:** Once a transaction is committed, its changes persist even in the event of system failure. For example, after the money transfer transaction is completed and confirmed, the changes to the account balances must be saved to permanent storage and remain intact, even if the system crashes afterwards