**Normalisation for Relational Databases – Step by Step from UNF to 3NF**

**Introduction: Data in Un-Normalised Form (UNF)**

Imagine a flat file is used to capture orders from customers making online purchases . The file contains the following data:

| **ORDER ID** | **ORDER DATETIME** | **CUSTOMER NAME** | **CUSTIOMER ADDRESS** | **CUSTOMER PHONE NO.** | **ORDERED ITEMS**  **(Product Name, Product Description, Product Price, Quantity Ordered)** |
| --- | --- | --- | --- | --- | --- |
| 1 | 2019-01-10 11:23 | Kang Poe Chew | 377 Hillcrest Road | 64662255 | [ (Smart Shoe Sensor, Special sensors that detect garish shoes, 100, 5),  (Plumber Robot, Robot that automatically fixes leaks on rooftops, 10000, 1) ] |
| 2 | 2019-05-07 23:45 | Hahman John | 373 Hillcrest Road | 64660033 | [(Plumber Robot, Robot that automatically fixes leaks on rooftops, 10000, 2),  (Agriculture Robot, Robot that automates farming, 5000, 10) ] |

When we instead use a relational database, we need to model the data within relations/tables such that we **avoid anomalies and redundancy**. To do this, we must normalise the data.

In other words, the above data is **unnormalised**; we denote this as UNF.

Unnormalised data will result in insert, delete and update anomalies on the data file. These anomalies will result in a loss of data integrity.

* Insert Anomaly:
  + What happens when the details of a customer need to be stored when the customer did not make any order?
* Delete Anomaly:
  + What happens when an order is cancelled?
* Update Anomaly:
  + What happens when the price of a product is changed?

**First normal form (1NF)**

In order to normalise the UNF data such that it complies with 1NF, it must satisfy the following:

* All values must be **atomic or no repeating groups**
  + This requires us to eliminate repeating groups in each table
  + For example, the ORDERED ITEMS attribute is not atomic; it contains repeating groups; there is more than 1 item listed in each cell of that column
* The resultant relations (i.e., tables) must be **isomorphic ( the rows in the table have the same structure)**
  + Rows need not follow a certain ordering for the relation to be used
  + Columns need not follow a certain ordering for the relation to be used
  + **There are no duplicate rows ie, each row must have a primary key**

We may address these by dividing the repeating groups into multiple columns and rows, as follows:

| **ORDER ID** | **ORDER DATETIME** | **CUSTOMER NAME** | **CUSTIOMER ADDRESS** | **CUSTOMER PHONE NO.** | **PRODUCT NAME** | **PRODUCT DESCRIPTION** | **PRODUCT PRICE** | **QUANTITY ORDERED** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2019-01-10 11:23 | Kang Poe Chow | 377 Hillcrest Road | 64662255 | Plumber Robot | Robot that automatically fixes leaks on rooftops | 10000 | 1 |
| 1 | 2019-01-10 11:23 | Kang Poe Chow | 377 Hillcrest Road | 64662255 | Smart Shoe Sensor | Special sensors that detect garish shoes | 100 | 5 |
| 2 | 2019-05-07 23:45 | Hahman John | 373 Hillcrest Road | 64660033 | Plumber Robot | Robot that automatically fixes leaks on rooftops | 10000 | 2 |
| 2 | 2019-05-07 23:45 | Hahman John | 373 Hillcrest Road | 64660033 | Agriculture Robot | Robot that automates farming | 5000 | 10 |

ORDER DETAIL relation

The table now contains atomic values, and we can uniquely identify each row using a combination of:

* ORDER ID
* PRODUCT NAME (assuming that no two products have the same name)

We may thus state that ORDER ID and PRODUCT NAME now comprise our **primary key.**

Thus, the data is now in first normal form – i.e., 1NF.

**Second normal form (2NF)**

For our data to be in second normal form, it must satisfy the following:

* Be in **1NF**
* Any attribute that is not part of the primary key must be dependent on **all attributes that are part of the primary key.** (applies to primary key that is a composite key)

Attribute Dependency

Let x, y be attributes in a table. We say that attribute y is functionally dependent on attribute x (usually the primary key), if for every valid instance of x , the value of x uniquely determines the value of y ( x→y ).

Let y be an attribute and S be a set of attributes of a table. y is fully dependent on S if all the attributes in S are required to uniquely determine the value of y . If not all the attributes are required, we say that y is partially dependent on S .

PRODUCT DESCRIPTION and PRODUCT PRICE are only dependent on PRODUCT NAME (assuming that no two products have the same name) and NOT on Order ID. This invalidates the requirements for the data to be in second normal form.

To fix this, we may divide the data up into the following:

| | **ORDER ID** | **ORDER DATETIME** | **CUSTOMER NAME** | **PRODUCT NAME\*** | **QUANTITY ORDERED** | **CUSTIOMER ADDRESS** | **CUSTOMER PHONE NO.** | | --- | --- | --- | --- | --- | --- | --- | | 1 | 2019-01-10 11:23 | Kang Poe Chow | Plumber Robot | 1 | 377 Hillcrest Road | 64662255 | | 1 | 2019-01-10 11:23 | Kang Poe Chow | Smart Shoe Sensor | 5 | 377 Hillcrest Road | 64662255 | | 2 | 2019-05-07 23:45 | Hahman John | Plumber Robot | 2 | 373 Hillcrest Road | 64660033 | | 2 | 2019-05-07 23:45 | Hahman John | Agriculture Robot | 10 | 373 Hillcrest Road | 64660033 |   ORDER DETAIL relation | | **PRODUCT NAME** | **PRODUCT DESCRIPTION** | **PRODUCT PRICE** | | --- | --- | --- | | Smart Shoe Sensor | Special sensors that detect garish shoes | 100 | | Plumber Robot | Robot that automatically fixes leaks on rooftops | 10000 | | Agriculture Robot | Robot that automates farming | 5000 |   PRODUCT relation |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

The primary key for the new PRODUCT relation is PRODUCT NAME and we now use the PRODUCT NAME on the ORDER DETAIL relation as a **foreign key** to reference the PRODUCT relation. The PRODUCT relation satisfies the criteria for second normal form.

However, the ORDER DETAIL relation still does not satisfy second normal form:

* ORDER DATETIME, CUSTOMER NAME , CUSTOMER ADDRESS and CUSTOMER PHONE NO. is only dependent on ORDER ID

Only QUANTITY ORDERED is dependent on both ORDER ID and PRODUCT NAME

To address this problem, we shall introduce another table: ORDER.

| | **ORDER ID** | **ORDER DATETIME** | **CUSTOMER NAME** | **CUSTIOMER ADDRESS** | **CUSTOMER PHONE NO.** | | --- | --- | --- | --- | --- | | 1 | 2019-01-10 11:23 | Kang Poe Chow | 377 Hillcrest Road | 64662255 | | 2 | 2019-05-07 23:45 | Hahman John | 373 Hillcrest Road | 64660033 |   ORDER relation | | **ORDER ID\*** | **PRODUCT NAME\*** | **QUANTITY ORDERED** | | --- | --- | --- | | 1 | Plumber Robot | 1 | | 1 | Smart Shoe Sensor | 5 | | 2 | Plumber Robot | 2 | | 2 | Agriculture Robot | 10 |   ORDER DETAIL relation |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

The primary key for the new ORDER relation is ORDER ID and we now use the ORDER ID on the ORDER DETAIL relation as a foreign key to reference the ORDER relation.

Now, PRODUCT, ORDER and ORDER DETAIL relations contain non-primary key attributes that are fully dependent on the entire primary key for the respective relation. Thus, the data is now in second normal form – i.e., 2NF.

**Third normal form (3NF)**

For our data to be in second normal form, it must satisfy the following:

* Be in **2NF**
* Any non-primary key attribute **must not be** dependent on any other non-key attribute

In the ORDER relation, notice that CUSTOMER ADDRESS and CUSTOMER PHONE NO. are only dependent on CUSTOMER NAME (assuming that no two customers have the same name), and not dependent on the primary key. This invalidates the requirements for the data to be in third normal form.

To fix this, we introduce another relation, CUSTOMER.

| **CUSTOMER NAME** | **CUSTIOMER ADDRESS** | **CUSTOMER PHONE NO.** |
| --- | --- | --- |
| Kang Poe Chow | 377 Hillcrest Road | 64662255 |
| Hahman John | 373 Hillcrest Road | 64660033 |

The primary key for the new CUSTOMER relation is CUSTOMER NAME and we now use the CUSTOMER NAME on the ORDER relation as a foreign key to reference the CUSTOMER relation.

The final set of relations in 3NF are:

| | **ORDER ID** | **ORDER DATETIME** | **CUSTOMER NAME\*** | | --- | --- | --- | | 1 | 2019-01-10 11:23 | Kang Poe Chow | | 2 | 2019-05-07 23:45 | Hahman John |   ORDER   | **ORDER ID\*** | **PRODUCT NAME\*** | **QUANTITY ORDERED** | | --- | --- | --- | | 1 | Plumber Robot | 1 | | 1 | Smart Shoe Sensor | 5 | | 2 | Plumber Robot | 2 | | 2 | Agriculture Robot | 10 |   ORDER DETAIL | | **CUSTOMER NAME** | **CUSTIOMER ADDRESS** | **CUSTOMER PHONE NO.** | | --- | --- | --- | | Kang Poe Chow | 377 Hillcrest Road | 64662255 | | Hahman John | 373 Hillcrest Road | 64660033 |   CUSTOMER   | **PRODUCT NAME** | **PRODUCT DESCRIPTION** | **PRODUCT PRICE** | | --- | --- | --- | | Smart Shoe Sensor | Special sensors that detect garish shoes | 100 | | Plumber Robot | Robot that automatically fixes leaks on rooftops | 10000 | | Agriculture Robot | Robot that automates farming | 5000 |   PRODUCT |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

This is represented in ERD as:

