**Bonus 1**

Normalize the following initial design to 3NF :

PRODUCT (Prod-ID, Description, Price, (Order-id, Order-date, Quantity-ordered), Supplier-id, SupplierName)

The initial design above, PRODUCT, is not in 1FN because there is a repeating group of columns which are Order-id, Order-date and Quantity-ordered.

1. 1st Normal Form

These following rules must be respected for the table to be in 1FN :

* Each column of the table must be single-valued.
* Each column should have a unique name.
* A column should contain the same type of values.
* The order of the data stored does not matter.

PRODUCT1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Prod-ID | Description | Price | Order-id | Order-date | Quantity-ordered | Supplier-id | SupplierName |
| 10101 | PC | 1500$ | 01 | 03/03 | 1 | 50500 | Skytech Gaming |
| 10102 | Headphones | 35$ | 01 | 03/03 | 1 | 50501 | PowerLucus |
| 10103 | Notebook | 2$ | 02 | 05/05 | 5 | 50502 | Five Stars |
| 10104 | Speakers | 85$ | 03 | 06/06 | 2 | 50503 | Edifier |

The primary key is a composite key of Prod-ID and Order-id because they are in a many-to-many relationship : a product can be opted by more than one order, and an order can opt for more than one product.

Now we have the following design in 1FN :

PRODUCT1 (Prod-ID, Description, Price, Order-id, Order-date, Quantity-ordered, Supplier-id, SupplierName)

The design above, PRODUCT1, is not in 2FN because there are partial dependencies.

1. 2nd Normal Form

These following rules must be respected for the table to be in 2FN :

* The table should be in 1st Normal Form.
* The table should not have any partial dependencies.

By observing the table PRODUCT1, we notice 2 partial dependencies :

* The Description only depends on Prod-ID and not necessarily on Order-id.
* The Order-date only depends on Order-id and not necessarily on Prod-ID.

Price, Quantity-ordered, Supplier-id and SupplierName are functionally dependent on the whole primary key (Prod-ID + Order-id). The price of a product varies depending on the product itself and the client’s choice of product(s). The quantity of an order varies depending on the product(s) ordered and also varies depending on the order(s) made. One can decide to order more of a product than another and orders can differ from each other. The supplier varies depending on the product ordered and the client who orders can opt products from different suppliers. It is possible that the supplier can also change for certain products without their product-ID changing, which means products can be overridden (Ex.: Oberlo). A client could order a product with a certain product-ID then another client could order the same product with the same product-ID but the supplier changed. If we were to look for the supplier into the records only by Prod-ID, we would not know which supplier we were looking for, since there would be 2 different suppliers recorded for the same product.

It is impossible to find a specific Price record or a specific Quantity-ordered record or a specific Supplier-id and SupplierName record with Prod-ID only or with Order-id only, which means Price, Quantity-ordered, Supplier-id and SupplierName are not partial dependencies.

We are to divide the table PRODUCT1 in 3 new tables to remove the partial dependencies :

* Product2 (Prod-ID, Description)
* Order2 (Order-id, Order-date)
* Details2 (Prod-ID, Order-id, Price, Quantity-ordered, Supplier-id and SupplierName)

Product2 Order2

|  |  |
| --- | --- |
| Prod-ID | Description |
| 10101 | PC |
| 10102 | Headphones |
| 10103 | Notebook |
| 10104 | Speakers |

|  |  |
| --- | --- |
| Order-id | Order-date |
| 01 | 03/03 |
| 02 | 05/05 |
| 03 | 06/06 |

Details2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Prod-ID | Order-id | Price | Quantity-ordered | Supplier-id | SupplierName |
| 10101 | 01 | 1500$ | 1 | 50500 | Skytech Gaming |
| 10102 | 01 | 35$ | 1 | 50501 | PowerLucus |
| 10103 | 02 | 2$ | 5 | 50502 | Five Stars |
| 10104 | 03 | 85$ | 2 | 50503 | Edifier |

PK in Product2 = Prod-ID

PK in Order2 = Order-id

PK in Details2 = Prod-ID + Order-id (composite key)

FK1 in Details2 = Prod-ID

FK2 in Details2 = Order-id

The following design is not in 3FN because there is a transitive dependency :

Product2 (Prod-ID, Description)

Order2 (Order-id, Order-date)

Details2 (Prod-ID, Order-id, Price, Quantity-ordered, Supplier-id and SupplierName)

1. 3rd Normal Form

These following rules must be respected for the table to be in 3FN :

* The table should be in 2nd Normal Form.
* The table should not have any transitive dependencies.

By observing the table Details2, we notice a transitive dependency :

The SupplierName only depends on Supplier-id and not necessarily on the composite primary key of Prod-ID and Order-id. Supplier-id is not part of the composite primary key in the table Details2.

We are to divide the table Details2 in two tables which results to this following design :

* Product2 (Prod-ID, Description)
* Order2 (Order-id, Order-date)
* Details3 (Prod-ID, Order-id, Price, Quantity-ordered, Supplier-id)
* Supplier3 (Supplier-id, SupplierName)

Product2 Order2

|  |  |
| --- | --- |
| Prod-ID | Description |
| 10101 | PC |
| 10102 | Headphones |
| 10103 | Notebook |
| 10104 | Speakers |

|  |  |
| --- | --- |
| Order-id | Order-date |
| 01 | 03/03 |
| 02 | 05/05 |
| 03 | 06/06 |

Details3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prod-ID | Order-id | Price | Quantity-ordered | Supplier-id |
| 10101 | 01 | 1500$ | 1 | 50500 |
| 10102 | 01 | 35$ | 1 | 50501 |
| 10103 | 02 | 2$ | 5 | 50502 |
| 10104 | 03 | 85$ | 2 | 50503 |

Supplier3

|  |  |
| --- | --- |
| Supplier-id | SupplierName |
| 50500 | Skytech Gaming |
| 50501 | PowerLucus |
| 50502 | Five Stars |
| 50503 | Edifier |

PK in Product2 = Prod-ID

PK in Order2 = Order-id

PK in Supplier3 = Supplier-id

PK in Details3 = Prod-ID + Order-id (composite key)

FK1 in Details3 = Prod-ID

FK2 in Details3 = Order-id

FK3 in Details3 = Supplier-id

Correction : FK2 = Supplier-id

No FK3 in Details3