****

**CZ2007 Lab 3 Report**

**Group 2**

**Date:** 9th October 2020

**SHOPS(Shops\_name)**

Keys: Shops\_name

Primary Key: Shops\_name

The relation is in 3NF.

**USERS(Users\_ID, Users\_name)**

Keys: Users\_ID

Primary Key: Users\_ID

FD: Users\_ID → Users\_name

The relation is in 3NF.

**EMPLOYEES(Employees\_ID, Employees\_name, Salary)**

Keys: Employees\_ID

Primary Key: Employees\_ID

FD: Employees\_ID → Employees\_name and Salary

The relation is in 3NF.

**ORDERS(Orders\_ID, Shipping-address)**

Keys: Orders\_ID

Primary Key: Orders\_ID

FD: Orders\_ID → Shipping-address

The relation is in 3NF.

**PRODUCTS(Products\_name, Maker, Category)**

Keys: Products\_name

Primary Key: Products\_name

FD: Products\_name → Maker and Category

The relation is 3NF. Each product has a unique name, as such the Products\_name should determine the maker and the category.

**PRODUCTS-IN-ORDER(Shops\_name, Products\_name, Orders\_ID, Status, Delivery-date, PIO\_quantity, PIO\_price)**

Keys: Shops\_name, Products\_name, and Orders\_ID

Primary Key: Shops\_name, Products\_name, and Orders\_ID

FDs: Shops\_name, Products\_name and Orders\_ID → PIO\_price, PIO\_quantity, Status, and Delivery-date

The relation is in 3NF. Orders\_ID does not have the name of the product, and from which shop. Thus, all three are required as key to show PIO\_price and PIO\_quantity. Also, Status and Delivery-date are for different products and from different shops, which again requires Product\_name and Shops\_name. Therefore, only Orders\_ID is not capable of showing everything in Products\_in\_Order. With Products\_name and Shops\_name, the keys can show it all.

**PRODUCTS-IN-SHOPS(Shops\_name, Products\_name, PIS\_price, PIS\_quantity)**

Keys: Shops\_name and Products\_name

Primary Key: Shops\_name and Products\_name

FD: Shops\_name and Products\_name → PIS\_price and PIS\_quantity

This relation is in 3NF. It is a subclass of Products, therefore it takes the key of Product. Also, Shops\_name is imperative, as seen in Appendix A that multiple shops can sell one product. Products\_name and Shops\_name together will find out the PIS\_price and PIS\_quantity.

**PRICE-HISTORY(Shops\_name, Products\_name, PH\_price, Start-date, End-date)**

Keys: Shops\_name, Products\_name

Primary Key: Shops\_name, Products\_name

FD: Shops\_name, Products\_name → PH\_price, Start-date, and End-date

This relation is in 3NF. Price-History is a weak entity of Products-in-Shop, thus its own key is the same as Products-in-Shop.

**COMPLAINTS-ON-ORDERS(Complaints\_ID, Orders\_ID, Products\_name)**

Keys: Complaints\_ID

Primary Key: Complaints\_ID

FD: Complaints\_ID → Orders\_ID and Products\_name

This relation is in 3NF. It is possible to be unhappy about a particular product, but still be happy about the entire order. Therefore, Products\_name is included as part of the table, so that customers can complain about products specifically.

**COMPLAINTS-ON-SHOPS(Complaints\_ID, Shops\_name)**

Keys: Complaints\_ID

Primary Key: Complaints\_ID

FD: Complaints\_ID → Shops\_name

This relation is in 3NF.

**COMPLAINTS( Complaints\_ID, Employee\_ID, Text, Status, and Filed-date-time)**

Keys: Complaints\_ID

Primary Key: Complaints\_ID

FD: Complaints\_ID → Text, Status, Filed-date-time and Employee\_ID

This relation is in 3NF. The Complaints\_ID should give the ID of the employee that handled it, and the text, status, and filed-date-time of the complaint.

**IN(Orders\_ID, Products\_name, Date-time)**

Keys: Orders\_ID and Products\_name

Primary Key: Orders\_ID and Products\_name

FD: Orders\_ID and Products\_name → Date-time

This relation is in 3NF. It is a Many-to-Many relationship, as such it requires the keys of both entities as its own. Therefore, it has Orders\_ID and Products\_name, as the Many-to-Many is between Order and Product.

**IN(Shops\_name, Products\_name, Price-variation, Start-date, End-date)**

Keys: Shops\_name and Products\_name

Primary Key: Shops\_name and Products\_name

FD: Shops\_name and Products\_name → Price-variation, Start-date, and End-date

This relation is in 3NF. It is a Many-to-Many relationship, as such it requires the keys of both entities as its own. Therefore, it has Shops\_name and Products\_name, as the Many-to-Many is between Shop and Product. This table allows us to store the price-variation, start and end date of the products being in the shop.

**FEEDBACK(Users\_ID, Shops\_name, Products\_name, Orders\_ID, Rating, Comment, Date-time)**

Keys: Users\_ID, Shops\_name, Products\_name, and Orders\_ID

Primary Key: Users\_ID, Shops\_name, Products\_name, and Orders\_ID

FD: Users\_ID, Shops\_name, Products\_name, and Orders\_ID → Rating, Comment, Date-time

This relation is in 3NF. As stated in Appendix A, users are allowed to only rate and comment after they have purchased a product. This means that they would have to place an order for a product. As such the key of feedback should include Users\_ID, Shops\_name, Products\_name, and Orders\_ID.