**Storing and Retrieving Data - Project Report**

**Sneakersly – Shoe Shop**

**Group H**

**Lecturers:**

Mijail Naranjo-Zolotov

Nuno Apalhão

Yuri Binev

**Group members:**

Berfin Sakallioglu - m20200545

Henrique Vaz - m20200586

Philipp Metzger - m20201058

Link to Github repository:

<https://github.com/ph1001/NOVA-IMS-SRD-Project>

**Description of our online shop**

Snearkersly is a shoe shop founded in 2020. The aim of our enterprise is to deliver many types of shoes using only an online selling platform. With a vast range of products, *Sneakersly* offers all kinds of sneakers, golf shoes, hiking boots, running shoes, and others.

Our shop relies on a wide range of suppliers that provide us with their high-quality products, allowing us to deliver any shoes at any time.

In order to buy a product in our online shop, the customer needs to register with a valid email address. Upon registration, a unique customer ID is associated with the customer. This customer ID will stay associated with this particular customer for future purchases as well. For ordering products, the customer has to specify the required details for delivery, such as the delivery address, the desired products, and their quantities. During the purchase process, a unique order ID is generated.

**The database**

Sneakerly uses the database *shoeshop* to store their data. The database is defined by the MySQL script ‘database\_and\_data\_creation.sql’.

The same script contains three triggers: Two that are required by task C[[1]](#footnote-1) and one more that ensures that when an order is placed by a customer, the value in EUR per ordered item is documented as an attribute of the respective entity of the entity type ‘customer\_order\_item’. This ensures that later on when any type of analysis is conducted on past purchases or when an invoice is created, these historical values in EUR are still available, even when the price of a respective product has been changed in the meantime.

The online shop has a stock of products. The table ‘stock’ keeps track of all the products that are currently in stock. One entity represents one specific product, such as the pair of shoes with the name ‘Nike\_123’. Each specific product has an available quantity associated with it which is represented by the attribute type ‘available\_quantity’.

Stock is added by ordering and receiving products from our suppliers who are represented by entities in the table ‘supplier’. Each entity of this table represents one of our suppliers.

When we order from our suppliers, an entity of the entity type ‘supplier\_order’ is created to which entities of the entity type ‘supplier\_order\_item’ are associated, which are created in the process of placing the order.

The products *Sneakersly* sells are all from the available stock. Due to the specification ‘unsigned’ of the attribute type ‘available\_quantity’ of table ‘stock’ it is ensured that no order can be processed that contains more items than the ones that are currently available.

Customers are represented by entities of the entity type ‘customer’. When a customer places an order, an entity of the entity type ‘customer\_order’ is created to which entities of the entity type ‘supplier\_order\_items’ are associated, which are created in the process of placing the order.

The purpose of the entity type ‘log\_price’ is to keep track of any changes in the prices of products in our stock. For example, if the price of a certain pair of shoes is lowered from 50 EUR to 40 EUR, this will be automatically documented as a new entity of the entity type ‘log\_price’. This table is required by the second part of task C.

Figure 1 shows the EER diagram of the database of *Sneakersly*.

Diagram

Description automatically generated

Fig 1: EER diagram of the database of *Sneakersly*

When creating the database, it was ensured that all three normal forms are complied with. In the following paragraphs, the three normal forms are described, and it is deliberated how our database complies with them.

The first normal form requires that no column has more than one value stored in a cell. In other words, it should only store atomic values. None of the tables in the database *shoeshop* holdcolumns containing multiple values.

As for the second normal form, it states that the first normal form must be granted (as we’ve said it is) but also no non-prime attribute is dependent on the proper subset of any candidate key of a table. In our tables, there aren´t any redundancies nor separable values disrespecting the second normal form.

Finally, the third normal form requires that the second normal form is fulfilled and that no transitive functional dependency of a non-prime attribute on any super key exists.

Since in all of our database's tables all attribute types are fully functionally dependent on the primary key of their table and all attribute types are atomic and single-valued and there are no transitive dependencies, all tables are in the third normal form.

The tables in Appendix I detail the dependencies between each table’s fields, showing there are no transitive functional dependencies.

**Other relevant files included in this hand in**

Solution for task D: ‘task\_D.ktr’

Readme file for task D: ‘task\_D\_Readme’

Input file for task D: ‘OldSystemCustomers.xls’

Output file for task D: ‘task\_D\_result.txt.sql’

Solution for task G: ‘task\_G.sql’

Solution for task H: ‘task\_H.sql’

**Appendix I**

The addition of ‘(PK)’ indicates that the respective attribute type is a primary key of the respective table and ‘PK Dependency’ denotes a dependency on the primary key of the respective table.

**3NF Demonstration**

|  |  |  |
| --- | --- | --- |
| Field 1 | Field 2 | Dependency |
| customer\_id (PK) | name | PK Dependency |
| customer\_id (PK) | phone\_number | PK Dependency |
| customer\_id (PK) | phone\_number\_country\_prefix | PK Dependency |
| customer\_id (PK) | email\_address | PK Dependency |
| customer\_id (PK) | home\_address | PK Dependency |
| customer\_id (PK) | spending\_score | PK Dependency |
| customer\_id (PK) | spending\_category | PK Dependency |
| customer\_id (PK) | age | PK Dependency |
| name | phone\_number | None |
| name | phone\_number\_country\_prefix | None |
| name | email\_address | None |
| name | home\_address | None |
| name | spending\_score | None |
| name | spending\_category | None |
| name | age | None |
| phone\_number | phone\_number\_country\_prefix | None |
| phone\_number | email\_address | None |
| phone\_number | home\_address | None |
| phone\_number | spending\_score | None |
| phone\_number | spending\_category | None |
| phone\_number | age | None |
| phone\_number\_country\_prefix | email\_address | None |
| phone\_number\_country\_prefix | home\_address | None |
| phone\_number\_country\_prefix | spending\_score | None |
| phone\_number\_country\_prefix | spending\_category | None |
| phone\_number\_country\_prefix | age | None |
| email\_address | home\_address | None |
| email\_address | spending\_score | None |
| email\_address | spending\_category | None |
| email\_address | age | None |
| home\_address | spending\_score | None |
| home\_address | spending\_category | None |
| home\_address | age | None |
| spending\_score | spending\_category | None |
| spending\_score | age | None |

Table 1: Customer

|  |  |  |
| --- | --- | --- |
| Field 1 | Field 2 | Dependency |
| supplier\_id (PK) | name | PK Dependency |
| supplier\_id (PK) | phone\_number | PK Dependency |
| supplier \_id (PK) | email\_address | PK Dependency |
| supplier\_id (PK) | business\_address | PK Dependency |
| name | phone\_number | None |
| name | email\_address | None |
| name | business\_address | None |
| phone\_number | email\_address | None |
| phone\_number | business\_address | None |
| email\_address | business\_address | None |

Table 2: Supplier

|  |  |  |
| --- | --- | --- |
| Field 1 | Field 2 | Dependency |
| product\_id (PK) | name | PK Dependency |
| product\_id (PK) | type | PK Dependency |
| product\_id (PK) | available\_quantity | PK Dependency |
| product\_id (PK) | unit\_price | PK Dependency |
| name | type | None |
| name | available\_quantity | None |
| name | unit\_price | None |
| type | available\_quantity | None |
| type | unit\_price | None |
| available\_quantity | unit\_price | None |

Table 3: Stock

|  |  |  |
| --- | --- | --- |
| Field 1 | Field 2 | Dependency |
| supplier\_order\_and\_invoice\_ID (PK) | order\_date | PK Dependency |
| supplier\_order\_and\_invoice\_ID (PK) | products\_received | PK Dependency |
| supplier\_order\_and\_invoice\_ID (PK) | supplier\_id | PK Dependency |
| supplier\_order\_and\_invoice\_ID (PK) | invoice\_date | PK Dependency |
| supplier\_order\_and\_invoice\_ID (PK) | paid | PK Dependency |
| order\_date | products\_received | None |
| order\_date | supplier\_id | None |
| order\_date | invoice\_date | None |
| order\_date | paid | None |
| products\_received | supplier\_id | None |
| products\_received | invoice\_date | None |
| products\_received | paid | None |
| supplier\_id | invoice\_date | None |
| supplier\_id | paid | None |
| invoice\_date | paid | None |

Table 4: Supplier\_Order

|  |  |  |
| --- | --- | --- |
| Field 1 | Field 2 | Dependency |
| order\_item\_id (PK) | supplier\_order\_and\_invoice\_ID | PK Dependency |
| order\_item\_id (PK) | product\_id | PK Dependency |
| order\_item\_id (PK) | quantity | PK Dependency |
| supplier\_order\_and\_invoice\_ID | product\_id | None |
| supplier\_order\_and\_invoice\_ID | quantity | None |
| product\_id | quantity | None |

Table 5: Supplier\_Order\_Item

|  |  |  |
| --- | --- | --- |
| Field 1 | Field 2 | Dependency |
| customer\_order\_and\_invoice\_ID (PK) | order\_date | PK Dependency |
| customer\_order\_and\_invoice\_ID (PK) | shipping\_date | PK Dependency |
| customer\_order\_and\_invoice\_ID (PK) | customer\_id | PK Dependency |
| customer\_order\_and\_invoice\_ID (PK) | shipping\_adress | PK Dependency |
| customer\_order\_and\_invoice\_ID (PK) | payment\_received | PK Dependency |
| customer\_order\_and\_invoice\_ID (PK) | invoice\_date | PK Dependency |
| customer\_order\_and\_invoice\_ID (PK) | invoice\_address | PK Dependency |
| customer\_order\_and\_invoice\_ID (PK) | tax\_percent\_rate | PK Dependency |
| order\_date | shipping\_date | None |
| order\_date | customer\_id | None |
| order\_date | shipping\_adress | None |
| order\_date | payment\_received | None |
| order\_date | invoice\_date | None |
| order\_date | invoice\_address | None |
| order\_date | tax\_percent\_rate | None |
| shipping\_date | customer\_id | None |
| shipping\_date | shipping\_adress | None |
| shipping\_date | payment\_received | None |
| shipping\_date | invoice\_date | None |
| shipping\_date | invoice\_address | None |
| shipping\_date | tax\_percent\_rate | None |
| customer\_id | shipping\_adress | None |
| customer\_id | payment\_received | None |
| customer\_id | invoice\_date | None |
| customer\_id | invoice\_address | None |
| customer\_id | tax\_percent\_rate | None |
| shipping\_adress | payment\_received | None |
| shipping\_adress | invoice\_date | None |
| shipping\_adress | invoice\_address | None |
| shipping\_adress | tax\_percent\_rate | None |
| payment\_received | invoice\_date | None |
| payment\_received | invoice\_address | None |
| payment\_received | tax\_percent\_rate | None |
| invoice\_date | invoice\_address | None |
| invoice\_date | tax\_percent\_rate | None |
| invoice\_address | tax\_percent\_rate | None |

Table 6: Customer\_Order

|  |  |  |
| --- | --- | --- |
| Field 1 | Field 2 | Dependency |
| order\_item\_id (PK) | customer\_order\_and\_invoice\_ID | PK Dependency |
| order\_item\_id (PK) | product\_id | PK Dependency |
| order\_item\_id (PK) | quantity | PK Dependency |
| order\_item\_id (PK) | discount\_percent | PK Dependency |
| order\_item\_id (PK) | rating | PK Dependency |
| order\_item\_id (PK) | value\_at\_time | PK Dependency |
| customer\_order\_and\_invoice\_ID | product\_id | None |
| customer\_order\_and\_invoice\_ID | quantity | None |
| customer\_order\_and\_invoice\_ID | discount\_percent | None |
| customer\_order\_and\_invoice\_ID | rating | None |
| customer\_order\_and\_invoice\_ID | value\_at\_time | None |
| product\_id | quantity | None |
| product\_id | discount\_percent | None |
| product\_id | rating | None |
| product\_id | value\_at\_time | None |
| quantity | discount\_percent | None |
| quantity | rating | None |
| quantity | value\_at\_time | None |
| discount\_percent | rating | None |
| discount\_percent | value\_at\_time | None |
| rating | value\_at\_time | None |

Table 7: Customer\_Order\_Item

|  |  |  |
| --- | --- | --- |
| Field 1 | Field 2 | Dependency |
| log\_id (PK) | product\_id | PK Dependency |
| log\_id (PK) | old\_unit\_price | PK Dependency |
| log\_id (PK) | new\_unit\_price | PK Dependency |
| log\_id (PK) | update\_date | PK Dependency |
| product\_id | old\_unit\_price | None |
| product\_id | new\_unit\_price | None |
| product\_id | update\_date | None |
| old\_unit\_price | new\_unit\_price | None |
| old\_unit\_price | update\_date | None |
| new\_unit\_price | update\_date | None |

Table 8: Log\_Price

1. The functionality of the two triggers required by task C can be demonstrated by running the code of the MySQL script ‘demonstrate\_trigger\_behaviour.sql’. [↑](#footnote-ref-1)