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] 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'.

[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'.

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*.

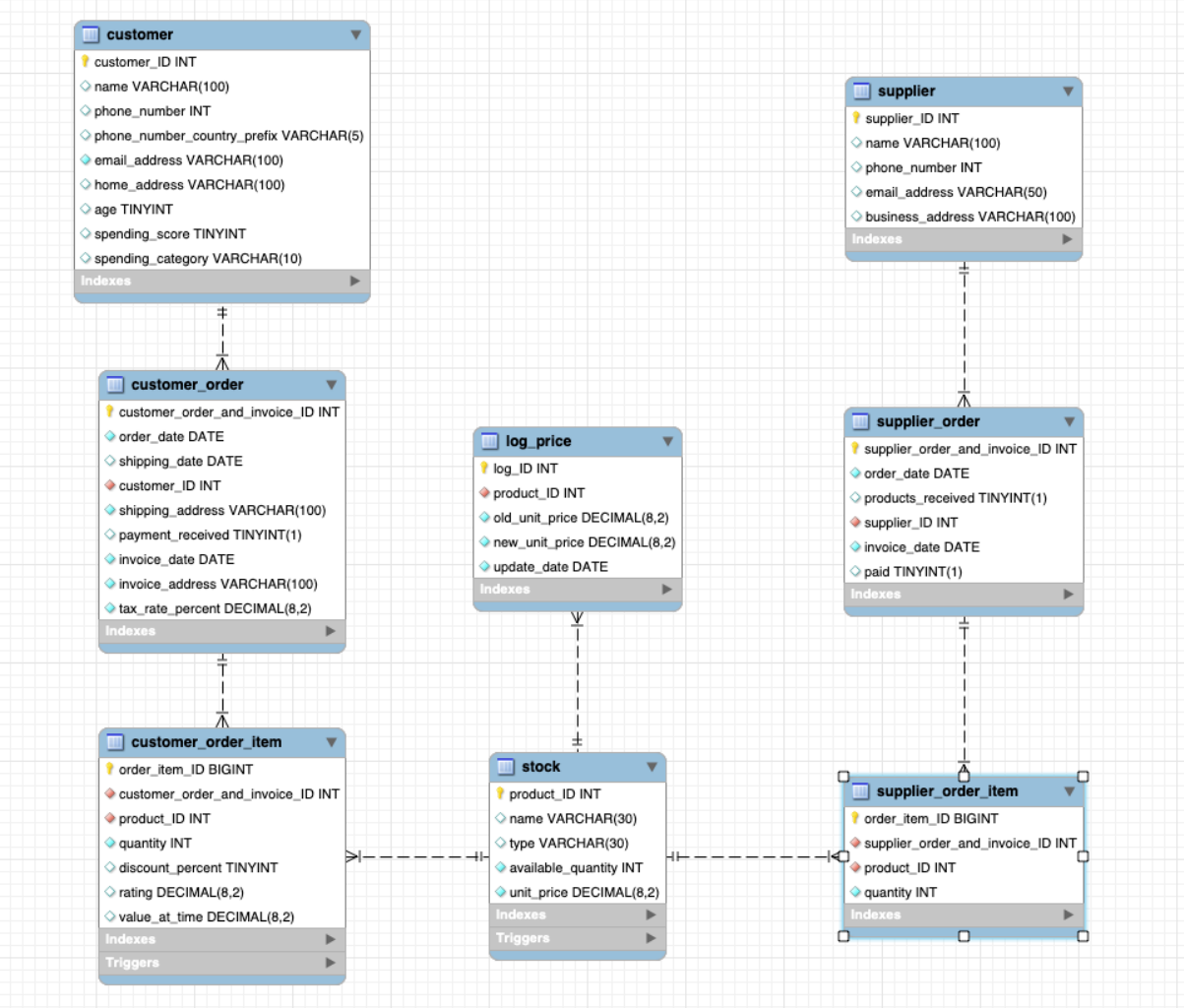

Fig 1: EER diagram of the database of *Sneakersly*

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* hold columns 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