

1. Task analysis:

A stationary store and storage where customers can buy various types of products intended mainly for hairdressers. The customers of my database will be people who want to buy hairdressing accessories, for example combs and clippers.

The database users will be workers who want find more information about given product for customers or for their purposes, also storage team, users in this category may require access to the system to see all different types of product, product offers, discounts etc. The purpose of the database is to enable and facilitate employees' access to details about products available in the store. This database allows to check all the necessary details that make working in this store easier.

Possible scenarios of database use:

- The customer wants to know the price of a given product, the employee obtains this information using the database
- The employee wants to know the quantity of a given product in the store.
- Searching for information about promotions for a given product.
- Wanting to check where a product might be in stock.
- The boss would like to check how much a given type of product cost.
- Someone wants who produce given product.
- The employee wants to check until when a given promotion will be active.

Assumptions and limitations of the designed database (e.g. limits of scope):

- one of the assumptions is that product sizes will have a name, i.e. (large, medium, small) etc, which is why we use the size name in the database.
- We assume every product can have only different sizes and colour, nothing else.
- Before add quantity of product to the cart, we have to check if this number is not higher than our quantity on a storage.
- Discount can count to the given product, but also to the cart, for example: 10% for everything.

Inquiries to the database:

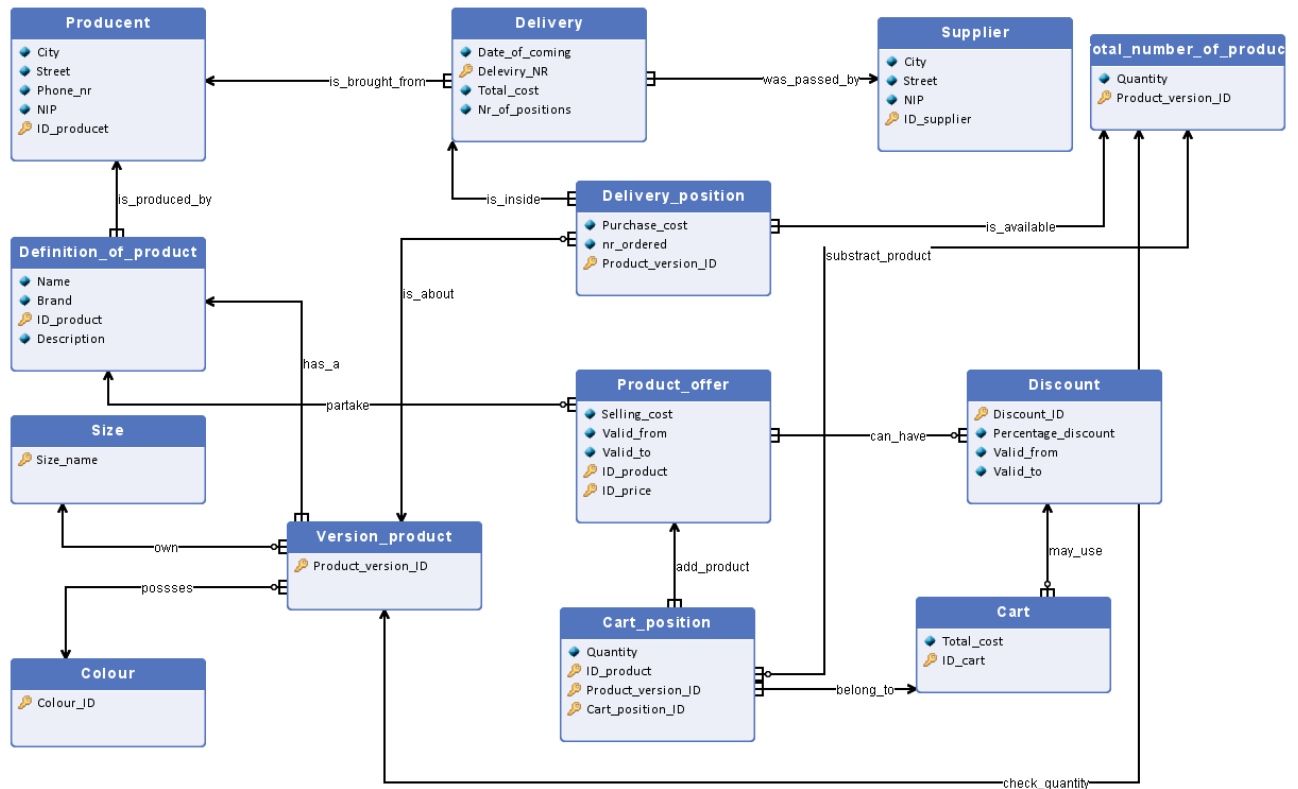
Ability to create a report on how many variants we have.

Option to sort deliveries from the newest.

Changing the validity date of discounts for a given product.

Ability to search any product and its details.

2. ERD diagram



3. Description of ERD diagram

Producer- A set of producers of products that are provided to us, another one is added if they start producing something for us and can be removed if there is no longer interest in their services.

~15 entries.

Definition_of_product - to get what a given product is, description, name, which company created it, we need an entity that describes this product, details about this product are later expanded, for each new product, for example a professional cape, a new entry is created, if we no longer interest about given product and there is no quantity of it in our shop, we can easily delete this product.

~20 entries.

Version_product - combines all the physical features of the product. It is a combination of 3 entities (definition_of_product, size and colour). A new one is added if a new color or size or product is added, it is deleted if any of the above no longer exists.

~ max 1000, probably less(different products can have different sizes and colors)

Size - is used to describe how large a given product is, the input is added if necessary for a given product, it can only be removed if a given size is no longer in production.

~5 entries.

Colour - is used to describe what color a given product is, the input is added if necessary for a given product, it can only be removed if a given color is no longer in production.

~ 10 entries.

Delivery - this is a package of different products delivered from producers, each pack creates a new entry with details.

~ 70 entries per month;

Delivery_position – in every delivery, there can be different products, so there exist a need to sort those items, every position is other version of product, number of new entries is number of different product in every delivery.

~ $70 * 10 = 700$ entries per month

Supplier – Supplier that provide to us the given delivery. It is a base of the supplier that we have contract with. New entry is created if new deal is clinch and the supplier deliver something. entry is remove if the relation with supplier does not exist already.

~ 5 entries;

Product_offer - It describes the price for a given product, as well as for what period of time it is valid, a new one is added if the price changes, and then the previous one is deleted.

~ 20 entries (number of products.)

Discount - It allows you to check what promotion is possible for a given product or to the cart, depending on the store and products, entries like a discount codes may be added or removed.

~ 50 entries

Total_number - The set of the total quantity of a given variant in a given store. Is used to check how many are currently in the store, we add the numbr if in delivery there exist given variant, and subtract if somebody buy this variant of the product.

~ max 1000, probably less(different products can have different sizes and colors)

Cart_position – to distinguish the situation when customer add different product, we divide cart into position, every position is different variant of product and entry is also added in that way.

~ $100 * 3 = 300$ entries every day (number of acquisition multiply by the possible mean of differents products in a cart)

Cart – Is a total of all cart position, there we sum up all the shopping, the new entry is added when somebody is buying something.

~ 100 entry every day

Attributes:

Producent:

- City (name of the city where the company's headquarters is located, written only using lowercase letters)
- Street (name of the street where the company's headquarters is located, written using only lowercase letters)
- Phone_nr (telephone number of the person managing the producent company, consisting of 9 digits)
- ID_producent (artificial ID to facilitate searching in the database, consists of 6 randomly selected digits, first letter meaning what function in our company it fullfil, for producent it is number 5)
- NIP (Unique identification number assigned to companies and entrepreneurs in Poland. Consists of 10 digits. The first digit indicates the country to which the number is assigned. In the case of Poland, it is usually the number 9. The next two digits indicate the voivodeship in which the company or entrepreneur is registered. The next seven digits are the unique identifier of the given business entity.)

Delivery:

- Date_of_coming (specifies when the delivery arrived at the company, recorded with dd/mm/yyyy using digits)
- Total_cost (It determines how much we, as a company, paid for the imported goods, recorded in zlotys and groszes after a dot.)
- Delivery_NR (number showing which package a given package is in our company, we use digits, for example the third package will have the value in the key "3", we have to fulfill 5 digits, so others numbers are randomly drawn)
- NR_of_positions (Number that indicates how many differents various products is located in a given delivery, written using digits)

Supplier:

- City (name of the city where the company's headquarters is located, there are only lowercase letters)
- Street (name of the street where the company's headquarters is located, written using only lowercase letters)
- NIP (Unique identification number assigned to companies and entrepreneurs in Poland. consists of 10 digits. The first digit indicates the country to which the number is assigned. In the case of Poland, it is usually the number 9. The next two digits indicate the voivodeship in which the company or entrepreneur is registered. The next seven digits are the unique identifier of the given business entity.)
- ID_supplier (artificial ID to facilitate searching in the database, consists of 6 randomly selected digits, first letter meaning what function in our company it fullfil, for supplier it is number 3)

Definiton_of_product:

- Name (describes the name of a given item in the store, written in lowercase letters)
- Brand (describes the name of the company producing a given product in the store, written in lowercase letters)
- ID_product (Artificial set of 6 digits to facilitate access to information about the product)
- Description (Detailed description of a given product proposed by the manufacturer or one of our employees, written in lowercase letters)

Version_product:

- Product_version_ID (fictitious , randomly choosen 6 digit number)

Size:

- Size_name (type of product size, written in lowercase letters)

Colour:

- Colour_ID (fictious randomly choose 4 digits for every colour to distinguish different models)

Delivery_position:

-Purchase_cost (it shows how we have to paid for given type of product and given time, recorded in zlotys and groszes after a dot.)

Nr_ordered (number indicate how many of a given version of product, we got in a delivery)

Product_version_ID (key from entity Product_version_ID (fictitious , randomly choosen 6 digit number) to distinguish products)

Product_offer :

- Selling_cost (selling cost of a relevant product, written in zlotys and groszes after a dot)
- Valid_from (specifies when the price start to obligate, recorded with dd/mm/yyyy using digits)
- Valid_to (specifies when the price will end to obligate, recorded with dd/mm/yyyy using digit)
- ID_product (key from entity Definition_of_product, Artificial set of 6 digits to facilitate access to information about the product, it is obligate to know about which product this offer is).
- ID_price (artificial random 6-digit number to be assigned to a given price.)

Discount:

- Discount_ID (artificial random 5-digit number to be assigned to a given discount.)
- valid_from(specifies when the discount start to obligate, recorded with dd/mm/yyyy using digits)
- valid_to (specifies when the discount will finish to obligate, recorded with dd/mm/yyyy using digits)
- Percentage_discount (written using numbers, the percentage that is subtracted from a given price of a product offer or cart, 30% promotion means the entered number 30.)

Total_nr_of_product:

- Quantity (a total number that shows how many of a given product variant we actually owned)
- Product_version_ID (key from entity Product_version_ID (fictitious , randomly choosen 6 digit number) to distinguish products)

Cart_position:

- Quantity (number that shows, how many of a given version of a product customer add to the cart)
- ID_product (key from entity Definition_of_product, Artificial set of 6 digits to facilitate access to information about the product, it is obligate to know about which product this offer is)
- Product_version_ID (key from entity Product_version_ID (fictitious , randomly choosen 6 digit number) to distinguish products)
- Cart_position_ID (number showing which product in a given cart we are interesting about)

Cart:

- Total_cost (total cost of a relevant cart, written in zlotys and groszes after a dot)
- ID_cart (artificial ID to facilitate searching the cart in the database, consists of 6 randomly selected digits)

Definition of relationships between entities:

is_brought_by (1 : 1..n) - relationship between Producent and Delivery entities, represent who produce given products from a delivery, each Producent can have 1 or more deliveries, but each deliver can be linked only to one producent.

was_passed_by (1..n : 1) – relationship between Delivery and Supplier entities, showing who pass the deliver, every deliver can be passed only by one supplier, but each supplier can passed 1 or more deliveries.

is_inside (1 : 1..n) – relationship between Delivery and Delivery_position, it represent that in every delivery there can be different products, and every position is different version of a product to help in distinguish. Every delivery can have 1 or more delivery positions, but each position is only linked to one deliver.

is_produced_by (1 : 1..n)- relationship between Producent and Definition_of_product entities, represent who produce given product, each Producent can produce 1 or more products, but every product can produce only one producer.

has_a (1 : 1..n) - relationship between Definition_of_product and Version_product entities, represent how many version of a product can have given product, every product can have one or more versions, but every version can be linked only to one product.

own (1 : 0..n) – relationship between Size and Version_product entities, represent possible sizes of a product's version, every version of a product is linked to only one size, but each size can be linked to 0 or more versions.

posses (1 : 0..n) - relationship between Colour and Version_product entities, represent possible colours of a product's version, every version of a product is linked to only one colour, but each colour can be linked to 0 or more versions.

Is_about (1 : 0..n) – relationship between Version_product and Delivery_position, it represent which version of product is located in a given delivery position, every version of a product can be linked to 0 or more positions, but every position is only about one version.

partake (1 : 0..n) – relationship between Definition_of_product and Product_offer , that represent about which product the offer is. Every product can possible have 0 or more offers, but every offer is only about one product.

can_have (1..n : 0..n) – relationship between Product_offer and Discount entities, it showing that products offer can have discounts. Every product offer can have 0 or more possible discounts and each discount can be used to 1 or more offers.

add_product (1..n : 1) – relationship between Cart_position and Product_offer entities, it represent that client can add to the cart given version of a product using product offer, every product offer can be add different cart positions (different customer), but every cart position is only about one product.

Belong_to (1..n : 1) – relationship between Cart_position and Cart entities, showing the whole cart is divided into smaller positions, every position is linked only to one Cart entities, but each Cart can have one or more Cart positions.

may_use (1 : 0..n) – relationship between Discount and Cart entities, it represent that every client can use discount for the cart, every cart can use 1 discounts for a given cart and the given discount can be used to 0 or more carts.

Substract_products (1 : 0..n) – relationship between Total_number_of_product and Cart_position entites, represent that after client buy something it has to be subtracted for the whole version product quantity, each cart_position can be linked only to one total number, because is only about one version, but each total number can be linked to 0 or more cart positions.

Is_available (1..n : 1)– relationship between Delivery_position and Total_number_of_product entites, represent that from the delivery products are added to the total number, there can be one or more deliveries, but each delivery position can be linked only to one total number.

Check_quantity (1 : 1) – relationship between Version_product and Total_number_of_product entites, represent that every total number is linked to one version of a product, and each version of a product can only one entity responsible showing the quantity available.

4. Relational database schema

Producent (City, Street, Phone_nr, NIP, ID_producent)

Delivery (Date_of_coming, Delivery_NR, Total_cost, NR_of_positions, ID_producent REF Producent, ID_supplier REF Supplier)

Definition_of_product (Name, Brand, ID_product, Description, ID_producent REF Producent)

Version_product (Product_version_ID, ID_product REF Definition_of_product, Size_name REF Size, Colour_ID REF Colour)

Size (Size_name)

Colour (Colour_ID)

Delivery_position(Purchase_cost, nr_ordered, Product_version_ID REF Product_version, Delivery_NR REF Delivery)

Product_offer(Selling_cost, Valid_from, Valid_to, ID_price, ID_product REF Definition_of_product)

Discount(Discount_ID, Percentage_discount, Valid_from, Valid_to)

possession (Offer REF Product_offer, Disc REF Discount)

Supplier (City, Street, NIP, ID_supplier)

Cart_position (Quantity, ID_product REF Product_offer , Product_version_ID REF Total_number_product, Cart_position_ID)

Cart(Total_cost, ID_cart, Discount_ID REF Discount)

Total_number_of_product(Quantity, Product_version_ID REF Product_version)

