# Introduction

# Requirements

Scenarios

Purchasing from manufactorer

1. The wholesaler places an order with the manufactorer supplying a reference number and enters the order in ManufactorerOrders.
2. The manufactorer sends an order confirmation to the wholesaler who verifies it and enters it in the ManufactorerOrderConfirmations.
3. The manufacturer sends an invoice to the Wholesaler who verifies it and enters it in the ManufactorerInvoices.
4. When the order is received the shipping manifest is compared to the order and the products are added to the ManufactorerReceptions
5. When the invoice is to be paid the payment is made and the ManufactorerInvoices is updated to reflect that the payment is done.

Customer purchasing product

1. The customer places an order with the web-shop and an entry is added to the CustomerOrders.
   1. If the customer pays with a credit card a NETS interface is used to validate the card and an ID is generated. This ID is used later when the money is to be transferred (may also be used in case the order is returned.
2. An order confirmation is generated and sent to the customer.
3. The order is packaged and shipped and the shipping manifest is added to CustomerDeliveries
4. An invoice is sent to the customer and it is entered in the CustomerInvoices
5. If the customer paid via credit card the money is withdrawn from the customer’s accont and the CustomerInvoices is updated accordingly.
6. If the customer pays the invoice manually the CustomerInvoices is updated when the money is received.

Design

Multiple references:

When we look at e.g. the customer purchase scenario we can see that there are several entities involved:



All of these refer to a customer, but they also refer to each other. In a completed order the Customer (C) possibly leads to a Customer Order (CO) leads to a Customer Order Confirmation (COC) and possibly a NETS Payment (NP). The COC leads to a Customer Invoice (CI) and a Customer Deliveries (CD). And from the Customer Deliveries there is a possible link to the NP (so the money may be drawn from the account after delivery). This gives the functional dependencies:

1. C -> CO
2. CO -> COC
3. CO -> NP
4. COC -> CI
5. COC -> CD
6. CD -> NP

This is the logic dependencies. However as a customer may have more than one order, but an order always refer to exactly one customer, it is more logical to create a CO -> C dependency with respect to relations.

Alternatively it is possible to design this in a star configuration by saying

1. C -> CO
2. CO -> C

When one has a Customer Order, one may have a

ER-diagram

Functional Dependenciesh

Performance and indexes

Transactions