**Technisch Ontwerp**

Webshop NerdyGadgets

|  |  |
| --- | --- |
| Versie: | 1.0 |
| Datum: | 15-12-2020 |

Algemene informatie

Projectorganisatie

|  |  |  |
| --- | --- | --- |
| Naam | Team | Rol: |
| Jasper in ’t Veld | 4 | Teamlid |
| Ivar Post | 4 | Teamlid |
| Anastasia | 4 | Voorzitter |
| Jan Willem Grimme | 4 | Teamlid |
| Roy Schottert | 4 | Teamlid |
|  |  |  |

Versiebeheer

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Datum | Omschrijving | Opmerkingen |
| 0.1 | 10-12-2020 | Begin gemaakt concept | Onze eigen toegevoegde tabellen ingevuld en gebruikers database |
| 0.2 | 12-12-2020 | Gedragsregels | Globaal uitgedacht en veel ingevoerd |
| 1.0 | 15-12-2020 | Laatste loodjes gelegd | PK key archive toegevoegd. Tekst bij kopjes toegevoegd |

Distributielijst

|  |  |  |
| --- | --- | --- |
| **Aan** | **Versie** | Datum |
| Leden groep 4 | 1.0 | 15-12-2020 |
| Wouter Keuning | 1.0 | 16-12-2020 |
| Coördinatoren KBS | 1.1 | 04-01-2021 |

Inhoudsopgave

[1. Inleiding 4](#_Toc58926771)

[1.1. De casus 4](#_Toc58926772)

[1.2. Missie en visie 4](#_Toc58926773)

[1.3. Project 4](#_Toc58926774)

[1.4. Over het document 4](#_Toc58926775)

[2. Huidig ERD 5](#_Toc58926776)

[2.1. Gebruikte tabellen 5](#_Toc58926777)

[3. Aanpassingen aan tabelstructuur 6](#_Toc58926778)

[3.1. Link naar filmpje 6](#_Toc58926779)

[3.2. Afbeeldingen producten 7](#_Toc58926780)

[3.3. Afbeeldingen categorieën 8](#_Toc58926781)

[3.4. Bestelling 9](#_Toc58926782)

[3.5. Archief ColdRoomTemperatures 10](#_Toc58926783)

[3.6. Auto-Increment 11](#_Toc58926784)

[4. Performance 12](#_Toc58926785)

[5. Autorisatie 12](#_Toc58926786)

[5.1. Gebruikers, rollen en privileges 12](#_Toc58926787)

[5.2. Views 12](#_Toc58926788)

[6. Regels in de database 13](#_Toc58926789)

[6.1. Gedragsregels 13](#_Toc58926790)

[6.2. Procedures en functies 13](#_Toc58926791)

[6.3. Triggers 14](#_Toc58926792)

[7. Storage engines 16](#_Toc58926793)

[8. Transacties 16](#_Toc58926794)

# Inleiding

## De casus

NerdyGadgets is een importeur en groothandel die producten levert aan verschillende warenhuizen en supermarkten in de Verenigde Staten. Ook levert NerdyGadgets producten aan weer andere groothandels. Incidenteel verkoopt NerdyGadgets producten rechtstreeks aan consumenten. NerdyGadgets werkt met een groot netwerk aan vertegenwoordigers die het land doortrekken om hun producten in de markt te krijgen. NerdyGadgets heeft ambities om hun activiteiten ook in Europa op te starten vanuit een nieuw hoofdkantoor in Amsterdam, maar de aandeelhouders hebben de opstart steeds tegengehouden omdat ze het een te groot risico vinden om investeringen te doen terwijl belangrijke klanten zijn omgevallen door de hevige concurrentie van online aanbieders.

## Missie en visie

“Wij zijn NerdyGadgets. We zijn trots op onze producten en geloven in kwaliteit. In ieder huis vindt een product van NerdyGadgets zijn thuis. Ondanks onze grootte zien klanten ons als kleinschalig en betrokken. Persoonlijk klantcontact staat bij ons hoog in het vaandel.”

## Project

Wij als groep hebben besloten ons bezig te houden met het maken van een prototype voor de website. Ook gaan we ons bezig houden om de conversie te verhogen van de website. NerdyGadgets heeft op het moment al een website maar deze heeft een conversie van 0.5%. De vraag was om dit te verhogen tot (op zijn minst) 2%. Daarom zijn wij aangesteld om een nieuwe website te ontwerpen en realiseren.

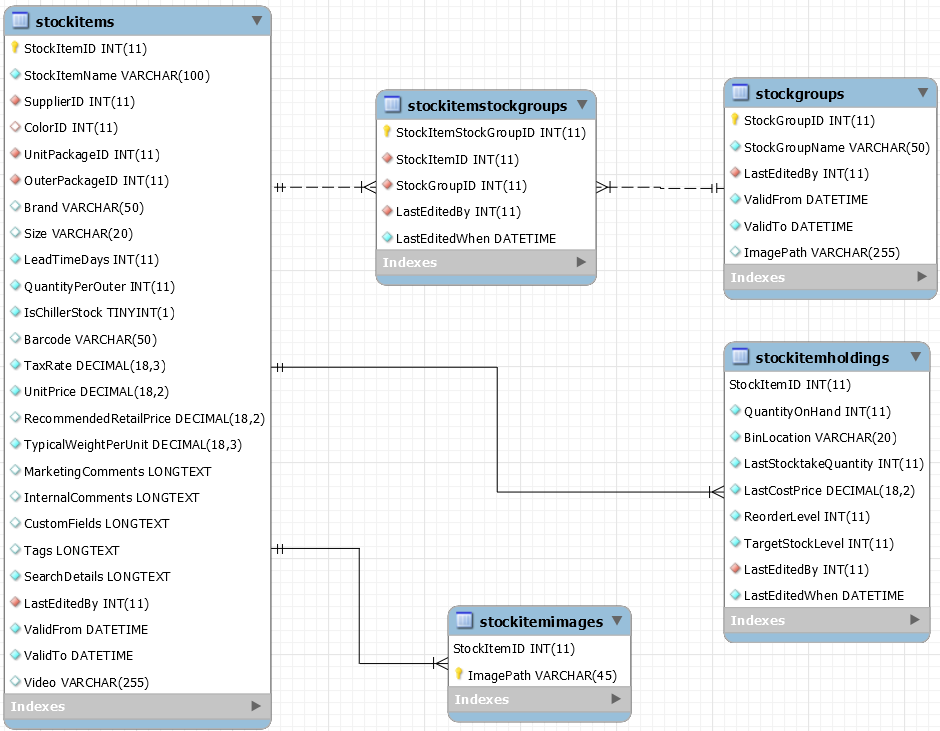
## Over het document

Dit is het Technisch Ontwerp. Hierin wordt de database en de aanpassingen aan de database besproken.

# Huidig ERD

In dit document wordt beschreven welke tabellen uit de actuele database gebruikt worden voor de huidige website van NerdyGadgets.

## Stockitems



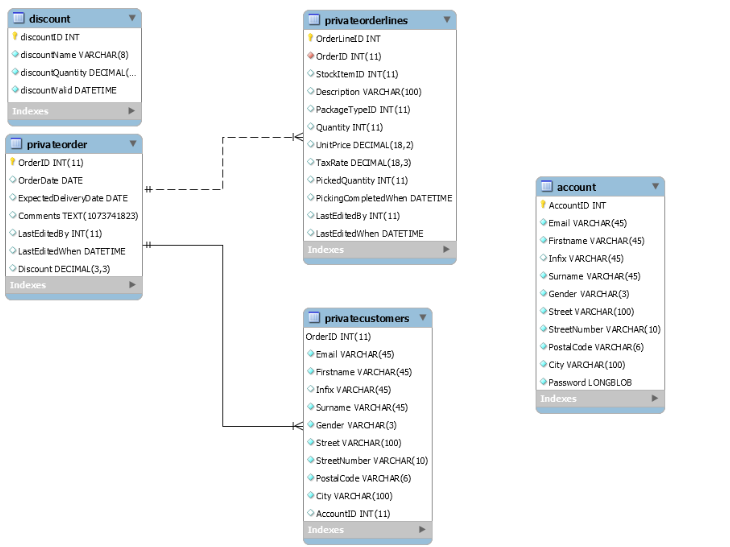
Stockitems zijn de producten.

In de tabel stockitemholdings staat hoeveel voorraad er nog is (QuantityOnHand).

In de tabel stockgroups staan alle categorieën waarin producten kunnen zitten.

In de tabel stockitemstockgroups staan de koppelingen tussen producten en categorieën. Eén product kan in een of meerdere categorieën zitten.

## Bestelling & Account



Deze tabellen worden gebruikt bij het plaatsen van een bestelling op de webshop.

In privateorder wordt het ID van de bestelling genoteerd samen met de algemene informatie van de bestelling.

In privateorderlines wordt per product uit privateorder de algemene informatie en prijs weergegeven. De tabel is via OrderID aan privateorder verbonden. Een bestelling in privateorder kan bij één of meer privateorderlines horen.

In privatecustomers wordt per bestelling de informatie van de klant opgeslagen. Als de klant voor de bestelling ingelogd was met een account van NerdyGadgets dan wordt het AccountID ook opgeslagen.

De tabel account bevat de accountsinformatie van de accounts die op NerdyGadgets bestaan. Wachtwoorden worden gehasht opgeslagen zodat deze niet afgelezen kunnen worden.

De tabel discount bevat alle geldige kortingscodes van de webshop, inclusief de datum tot wanneer deze geldig zijn.

# Aanpassingen aan tabelstructuur

De originele database voldeed niet voor de webshop. Daarom hebben we de volgende aanpassingen gedaan.

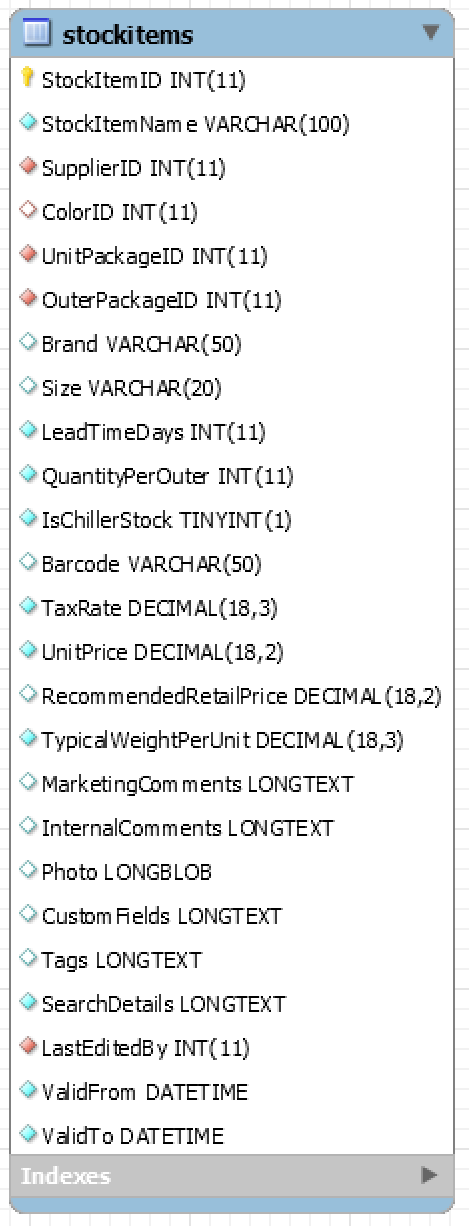
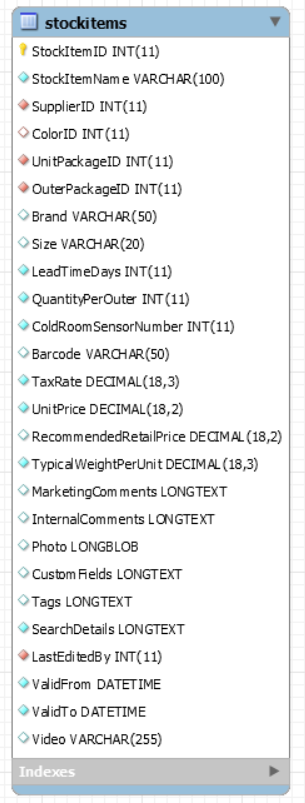
## Link naar filmpje

**Waarom de huidige tabelstructuur niet voldoet**

Voor onze user story bekijken producten moet het mogelijk zijn dat er voor een product een video laat spelen. Deze informatie was niet beschikbaar in de database.

**Nieuwe situatie**

In de ‘stockitems’ tabel wordt de niet verplichte kolom Video geplaatst, dit is een VARCHAR met een maximum van 255 karakters.

**Oude situatie Nieuwe situatie**

## Afbeeldingen producten

**Waarom de huidige tabelstructuur niet voldoet**

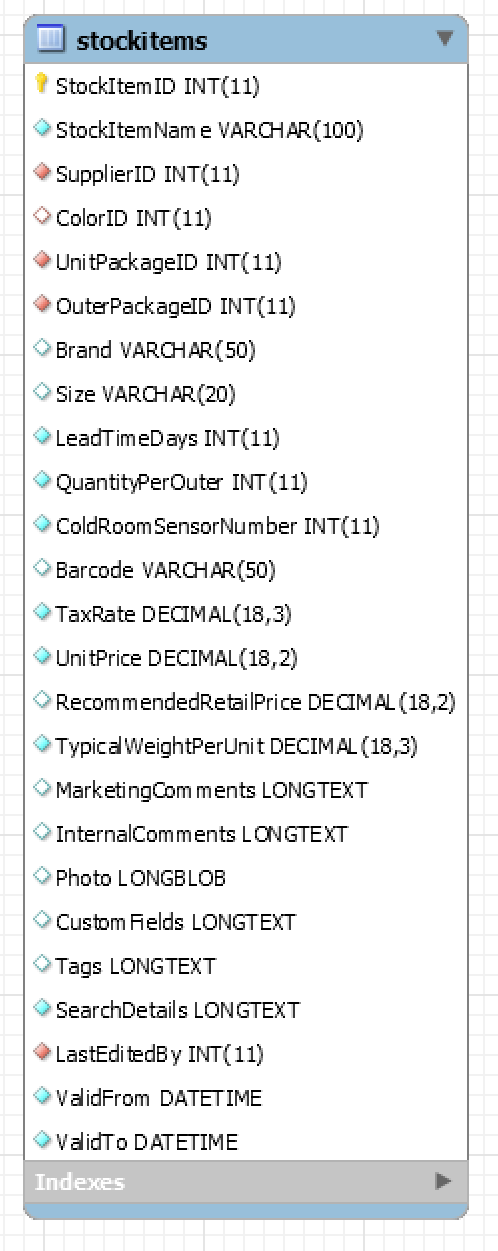
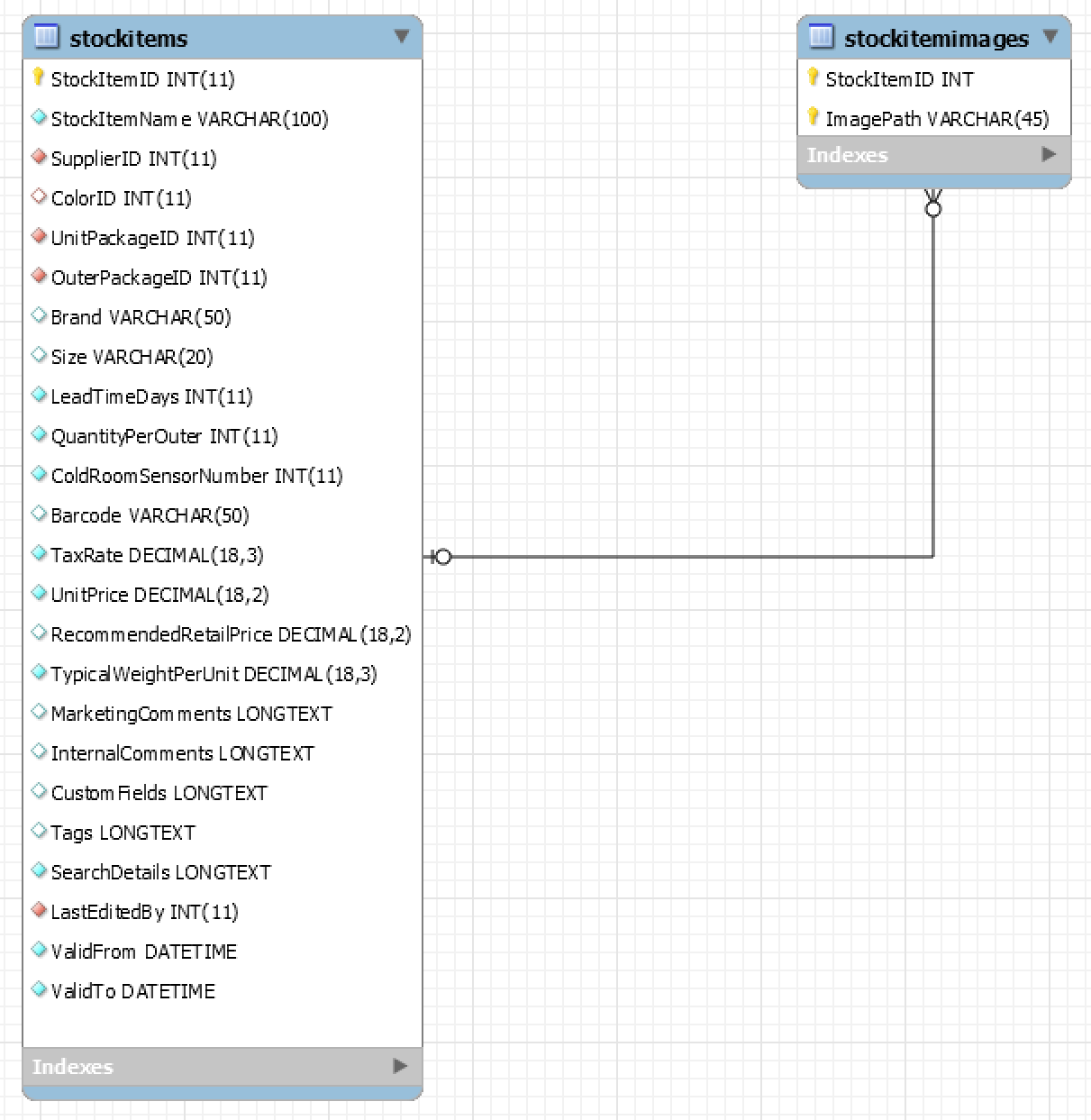
Er kan per product maar één foto toegevoegd worden. Voor de User story Bekijken producten moeten we echter meerdere afbeeldingen laten zien.

**Nieuwe situatie**

Er wordt een tabel ‘stockitemimages’ toevoegd en daarin twee primaire sleutels, StockItemID VARCHAR ImagePath. Beide variabelen zijn verplicht. De verbinding tussen de tabellen is een primary key waarin er nul of meerdere plaatjes per artikel kunnen zijn.

De kolom Photo wordt uit de tabel ‘stockitems’ verwijderd.

**Oude situatie Nieuwe situatie**



## Afbeeldingen categorieën

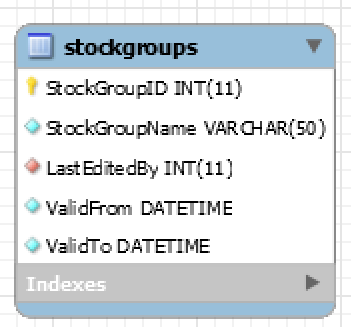
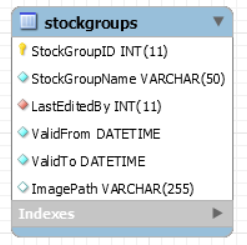
**Waarom de huidige tabelstructuur niet voldoet**

Voor onze User stories Bladeren door producten en Bekijken producten wordt de afbeelding van de categorie gebruikt. Doordat deze nog niet beschikbaar is kunnen deze requirements niet opgeleverd worden.

**Nieuwe situatie**

Voor de afbeeldingen van de categorieën voegen we een kolom toe in de ‘stockgroups’ tabel. Deze kolom is van het type VARCHAR en is niet verplicht.

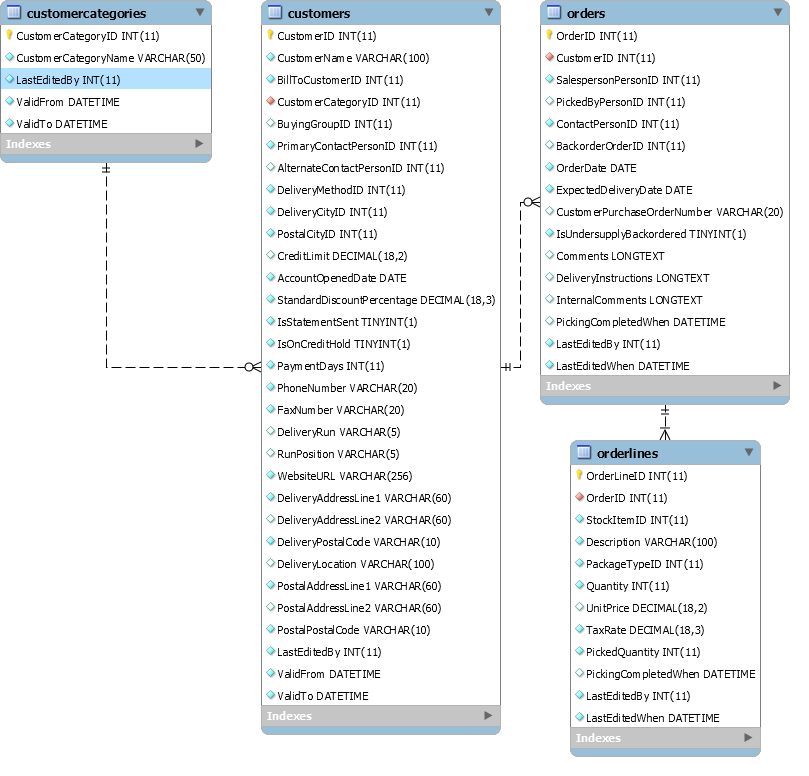
**Oude situatie Nieuwe situatie**



## Bestelling

**Waarom de huidige tabelstructuur niet voldoet**

De huidige situatie richt zich op de grotere bedrijven. Wij richten ons echter op particuliere klanten, en kunnen geen gebruik maken van al bestaande tabellen voor de administratie van de webshop. De tabellen in de oude situatie, customers, customercategories, orders & orderlines zijn de tabellen die voor de NerdyGadgets webshop niet geschikt zijn.

**Oude Situatie**

**Nieuwe situatie**

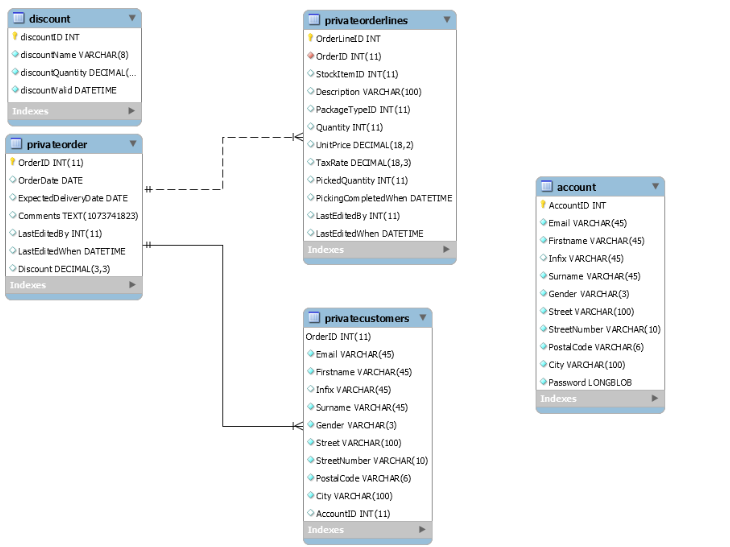
De structuur voor de tabellen customer, order en orderlines zijn overgenomen voor de nieuwe situatie en ingevuld met velden die meer geschikt zijn voor particuliere klanten.

Customercategories is in de nieuwe situatie niet nodig, en dus afwezig.

Een voorbeeld van een aanpassing is het overstappen van het enkele CustomerName veld in customers naar Firstname, Infix en Surname in privatecustomers, wat de webshop de mogelijkheid geeft om enkel de voornaam of achternaam weer te geven.

De centrale tabel in het bestellingsproces is privateorder. Iedere bestelling wordt beginnend in deze tabel opgenomen en krijgt een uniek OrderID. Deze tabel bevat de informatie over de bestelling als geheel, in dit geval de datum van de bestelling en wanneer van toepassing de kortingscode die op de bestelling is toegepast.

Verbonden aan privateorder zijn privateorderlines en privatecustomers, verantwoordelijk voor het opslaan van de informatie van ieder individueel product in de bestelling, en de informatie van de klant die de bestelling geplaatst heeft. Deze drie tabellen zijn via de key OrderID met elkaar verbonden.

**Nieuwe Situatie**

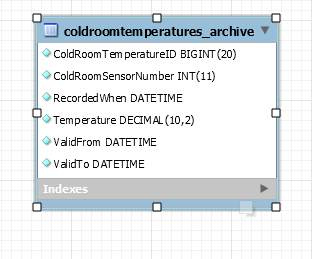
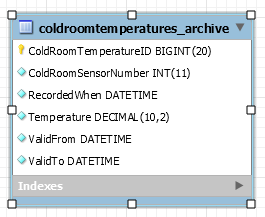
## Archief ColdRoomTemperatures

**Waarom de huidige tabelstructuur niet voldoet**

De oude situatie voldeed niet aan de eisen, omdat het ID in het archief niet automatisch geüpdatet werd. Van ColdRoomTemperatures nemen wij niet het ID van de regel mee naar het archief, maar het archief maakt een eigen ID per regel.

**Nieuwe situatie**

In de nieuwe situatie hebben we een Primaire sleutel aan ColdRoomTemperatureID in het archief toegevoegd. Hierdoor wordt het ID in het archief, bij het invoegen van nieuwe data, met 1 meer dan het laatst toegevoegde ID verhoogd.

Oude situatie Nieuwe situatie

## Auto-Increment

In de tabellen people, customers, orders, orderlines wordt geen gebruik gemaakt van Auto-Increment (afgekort naar AI). Bij nieuwe records zonder AI moet er een primary\_key aangegeven worden. Om dit goed te implementeren moeten er bij elke INSERT eerst een query uitgevoerd worden om de nieuwe primary\_key vast te kunnen stellen. Wanneer er gebruikt wordt van AI hoeft dit niet, dan doet SQL dit voor je.

ALTER TABLE `people`

CHANGE COLUMN `PersonID` `PersonID` INT(11) NOT NULL AUTO\_INCREMENT ;

LOCK TABLES `customers` WRITE;

ALTER TABLE `customers`

CHANGE COLUMN `CustomerID` `CustomerID` INT(11) NOT NULL AUTO\_INCREMENT ;

LOCK TABLES `orders` WRITE;

ALTER TABLE `orders`

CHANGE COLUMN `OrderID` `OrderID` INT(11) NOT NULL AUTO\_INCREMENT ;

SET max\_statement\_time=0;

LOCK TABLES `orderlines` WRITE;

ALTER TABLE `orderlines`

CHANGE COLUMN `OrderLineID` `OrderLineID` INT(11) NOT NULL AUTO\_INCREMENT ;

# Performance

De database gebruikt op dit moment geen indextabellen. Ook heeft de database minder dan 0,4s nodig voor iedere query die vanuit we webshop uitgevoerd, en voldoet daarmee aan de eisen die aan de snelheid van de database gesteld worden. Indextabellen zijn daarom ook niet nodig.

Ditzelfde geldt voor artificiële sleutels. De database is snel genoeg om hier geen gebruik van hoeven te maken.

# Autorisatie

## Gebruikers, rollen en privileges

De database gaat twee gebruikers kennen: de klant, administrator en de pi.

Om te voorkomen dat een kwaadwillend individu met toegang tot de root gebruiker complete controle over de database zou kunnen krijgen wordt de root gebruiker uitgeschakeld.

De gebruiker klant kan heeft de volgende permissies:

* SELECT bij de volgende tabellen:
  + Stockitemimages
  + Stockitems
  + Stockitemholdings
  + Stockitemstockgroups
  + Stockgroups
  + Discount
  + Account
* UPDATE bij de volgende kolom in de tabel stockitemholdings:
  + QuantityOnHand
* INSERT in de tabel account
* INSERT in de tabellen privateorder, privateorderlines, privatecustomers

De gebruiker pi heeft de volgende permissies:

* UPDATE bij de kolom Temperature in de tabel coldroomtemperatures
* INSERT de verouderde temperatuur in de tabel coldroomtemperatures\_archive.

De gebruiker administrator heeft alle rechten die root ook heeft. Mocht er iets fout gaan, kan de root geactiveerd worden. De functie van de administrator is hetzelfde als de root, maar is gemakkelijker te monitoren.

We hebben op dit moment voor iedere gebruiker een aparte rol.

## Views

Views zijn handig voor het laten zien van specifieke informatie die de gebruiker mag zien.

Deze maatregelen is echter niet geschikt om SQL-injecties tegen te gaan, omdat een kwaadwillende gebruiker alsnog andere tabellen naar voren zou kunnen halen.

Voor ieder query die op de database afgevuurd wordt, is een view geschreven. Hiermee zorgen we ervoor dat het lastiger wordt om de database te analyseren en te exploiteren.

# Regels in de database

## Gedragsregels

Op dit moment hebben wij geen CHECKs of CONSTRAINTs geïmplementeerd. Er is geen data toegevoegd aan de tabellen die afhankelijk zijn van andere data. Op dit moment wordt alles geregeld met TRIGGERs en PROCEDUREs.

## Procedures en functies

De onderstaande procedure hebben we toegevoegd, om het archiveren van verouderde data volledig te automatiseren op database niveau.

Tabel : Insert\_Temperatuur\_Archief

|  |
| --- |
| Er wordt procedure aangemaakt voor het automatiseren van het importeren van de oude gegevens van de temperatuursensoren naar het archief.  Om het automatisch importeren succesvol te laten lopen, hebben we in overleg de tabel *coldroomtemperatures\_archive een primaire sleutel toegevoegd (ColdRoomSensorID). Dit is in overleg gegeaan met de heer Wouter Keuning.*  *In ColdRoomTemperutures wordt de ID niet door ons gebruikt, vandaar dat een update voldoende zou moeten zijn.* |
| *DELIMITER //*  *CREATE OR REPLACE PROCEDURE Insert\_Temperatuur\_Archief()*  *MODIFIES SQL DATA*  *BEGIN*  *INSERT INTO coldroomtemperatures\_archive SELECT ColdRoomSensorNumber, RecordedWhen, Temperature, ValidFromValidFrom, ValidTo FROM coldroomtemperatures WHERE ColdRoomSensorNumber = 5;*  *END*  *//*  *DELIMITER ;* |

## Triggers

Tijdens het ontwikkelen hebben we de onderstaande triggers toegevoegd. Deze zorgen ervoor dat de juiste data in de juiste tabellen komen.

Tabel 1: Geïmplementeerde trigger

|  |
| --- |
| Er wordt een trigger toegevoegd voor het veld EmailAddress van de tabel People. Dit wordt gedaan om te voorkomen dat een ongeldig emailadres wordt ingevoerd. Dezelfde trigger wordt ook voor de tabellen account en privatecustomers gebruikt.  Een emailadres moet voldoen aan het volgende format:  [één of meerdere letters of cijfers]@[één of meerdere letters of cijfers]. [één of meerdere letters of cijfers] |
| **Correcte email people** |
| *DELIMITER //*  *CREATE TRIGGER correcte\_email*  *BEFORE INSERT ON people*  *FOR EACH ROW*  *BEGIN*  *IF NEW.EmailAddress NOT LIKE '\_%@\_%.\_\_%' THEN*  *SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Email field is not valid';*  *END IF;*  *END;*  *//*  *DELIMITER ;* |
| **Correcte email privatecustomers** |
| *DELIMITER $$*  *CREATE TRIGGER `correcte\_email\_privatecustomers`*  *BEFORE INSERT ON `privatecustomers` FOR EACH ROW*  *BEGIN*  *IF NEW.Email NOT LIKE '\_%@\_%.\_\_%' THEN*  *SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Email field is not valid';*  *END IF;*  *END*  *$$*  *DELIMITER ;* |
| **Correcte email account** |
| *DELIMITER $$*  *CREATE TRIGGER `correcte\_email\_account`*  *BEFORE INSERT ON `account` FOR EACH ROW*  *BEGIN*  *IF NEW.Email NOT LIKE '\_%@\_%.\_\_%' THEN*  *SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Email field is not valid';*  *END IF;*  *END*  *$$*  *DELIMITER ;* |

Tabel 2: Trigger\_Archief\_Temperatuur

|  |
| --- |
| Voor de database willen wij een trigger maken, die vóór de UPDATE in de coldroom\_temperatures tabel een procedure aanroept.  Voordat er nieuwe temperaturen in de database gezet worden, worden de oude gegevens in de archieftabel gezet. Dat gaat met onderstaande trigger. |
| *DELIMITER //*  *CREATE OR REPLACE TRIGGER Insert\_Temperatuur\_Archief\_Trigger*  *BEFORE UPDATE ON coldroomtemperatures*  *FOR EACH ROW*  *BEGIN*  *CALL Insert\_Temperatuur\_Archief();*  *END //*  *DELIMITER ;* |

Tabel 3: Check\_op\_Postcode

|  |
| --- |
| *De afdeling logistiek heeft als belang dat de postcode en huisnummer juist is. Controleer elke INSERT of UPDATE van een postcode voor het (aflever)adres van de klant. Het patroon voor de check moet: <cijfer><cijfer><cijfer><cijfer><letter><letter>  OF  <cijfer><cijfer><cijfer><cijfer><spatie><letter><letter> zijn. Logistiek eist een controle op het patroon. Mocht je een check ontwikkelen die verder gaat dan patroonherkenning dan is dit erg mooi! Je mag andere handige checks bedenken en doorvoeren.* |
| DELIMITER //  CREATE OR REPLACE TRIGGER insertAddress  BEFORE INSERT ON account  FOR EACH ROW  BEGIN  IF NEW.PostalCode NOT LIKE '[1-9][0-9][0-9][0-9][A-Z][A-Z]'  AND NEW.PostalCode NOT LIKE'[1-9][0-9][0-9][0-9] [A-Z][A-Z]' THEN  SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'postalCode field is not valid';  END IF;  END;  //  DELIMITER ; |

# Storage engines

De archief tabellen (Eindigend met \_archive) worden van engine veranderd. Deze gaan van INNODB naar archive. Dit is handig, omdat archive alle data zoveel mogelijk comprimeert, wat ervoor zorgt dat de bestanden zo klein mogelijk worden. Hierdoor wordt de opslagruimte van de database sterk verminderd.

Verdere tabellen worden niet gewijzigd, omdat ze zoals ze momenteel zijn effectief zijn.

# Transacties

Een TRANSACTION is een query die uitgevoerd wordt op de database. In deze query zitten andere query’s die niet zonder elkaar uitgevoerd mogen worden.

Op dit moment gebruiken wij geen TRANSACTIONs, omdat wij op het moment geen data hebben dat te maken heeft met geld.