**Technisch Ontwerp**

Webshop NerdyGadgets

|  |  |
| --- | --- |
| Versie: | 1.0 |
| Datum: | 10-12-2020 |

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 |
| 1.0 | 10-12-2020 | Begin gemaakt concept | Onze eigen toegevoegde tabellen ingevuld en gebruikers database |

Distributielijst

|  |  |  |
| --- | --- | --- |
| **Aan** | **Versie** | Datum |
| Leden groep 4 | 1.0 | 10-12-2020 |
|  |  |  |

Inhoudsopgave

[1. Inleiding 4](#_Toc58668033)

[1.1. De casus 4](#_Toc58668034)

[1.2. Missie en visie 4](#_Toc58668035)

[1.3. Project 4](#_Toc58668036)

[1.4. Over het document 4](#_Toc58668037)

[2. Huidig ERD 5](#_Toc58668038)

[2.1. Gebruikte tabellen 5](#_Toc58668039)

[3. Aanpassingen aan tabelstructuur 6](#_Toc58668040)

[3.1. Link naar filmpje 6](#_Toc58668041)

[3.2. Afbeeldingen producten 7](#_Toc58668042)

[3.3. Afbeeldingen categorieën 8](#_Toc58668043)

[3.4. Bestelling 9](#_Toc58668044)

[3.5. Auto-Increment 10](#_Toc58668045)

[4. Performance 11](#_Toc58668046)

[5. Autorisatie 11](#_Toc58668047)

[5.1. Gebruikers, rollen en privileges 11](#_Toc58668048)

[5.2. Views 11](#_Toc58668049)

[6. Regels in de database 12](#_Toc58668050)

[6.1. Gedragsregels 12](#_Toc58668051)

[6.2. Procedures en functies 12](#_Toc58668052)

[6.3. Triggers 13](#_Toc58668053)

[7. Storage engines 15](#_Toc58668054)

[8. Transacties 15](#_Toc58668055)

# 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.

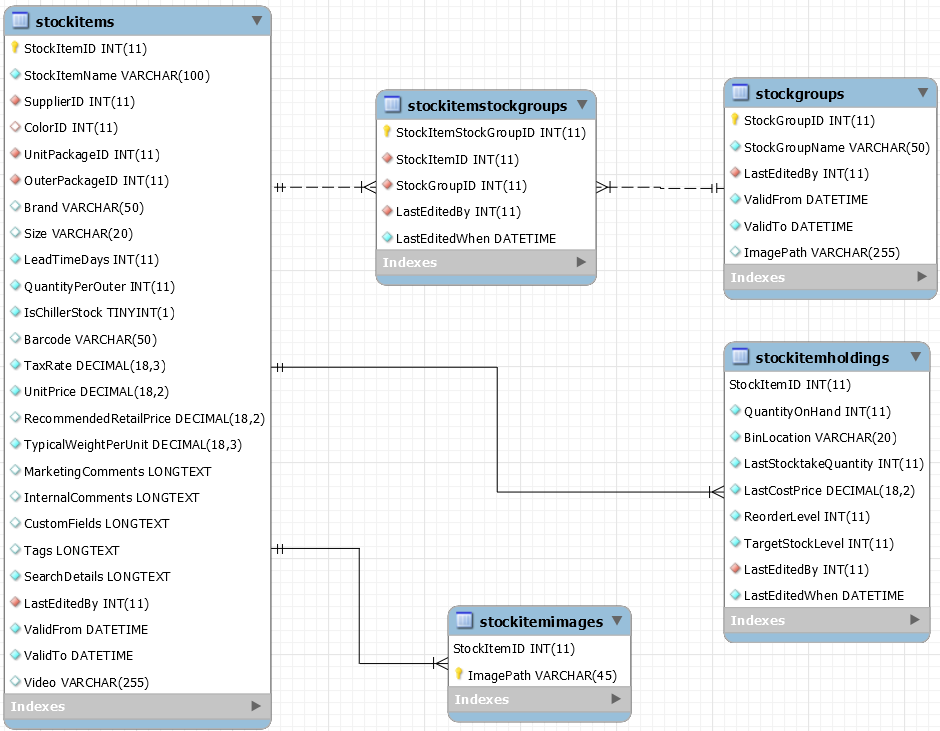
## Over het document

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.

## Gebruikte tabellen



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.

# Aanpassingen aan tabelstructuur

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

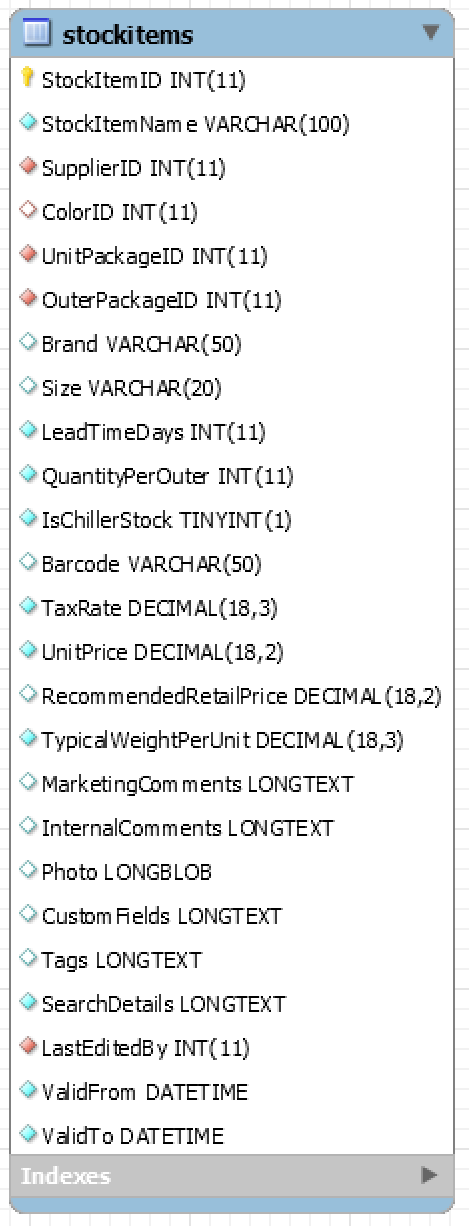
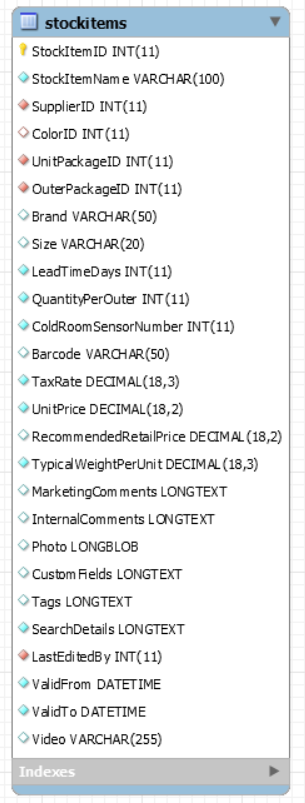
## Link naar filmpje

**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**

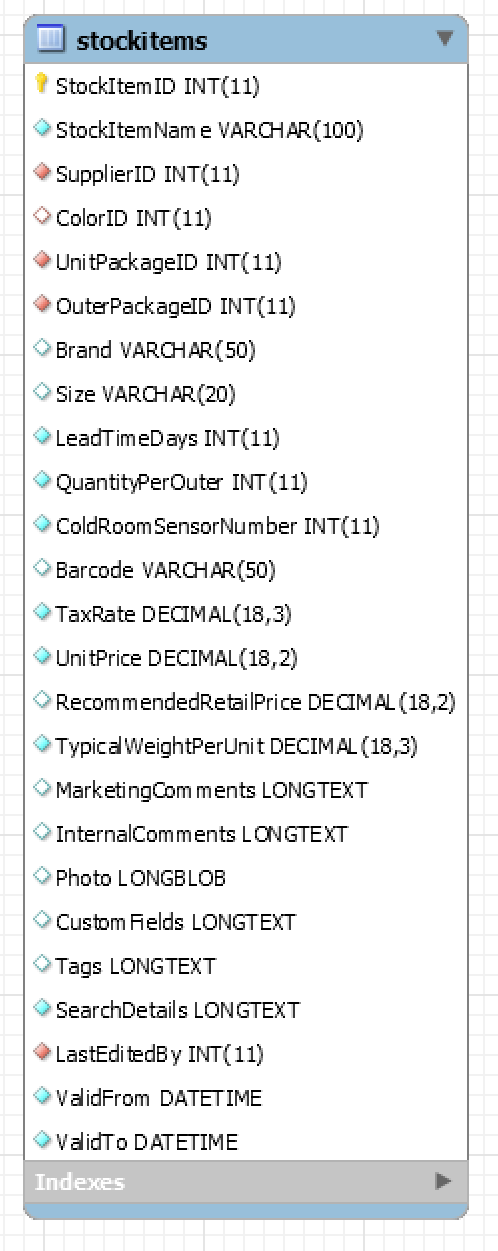
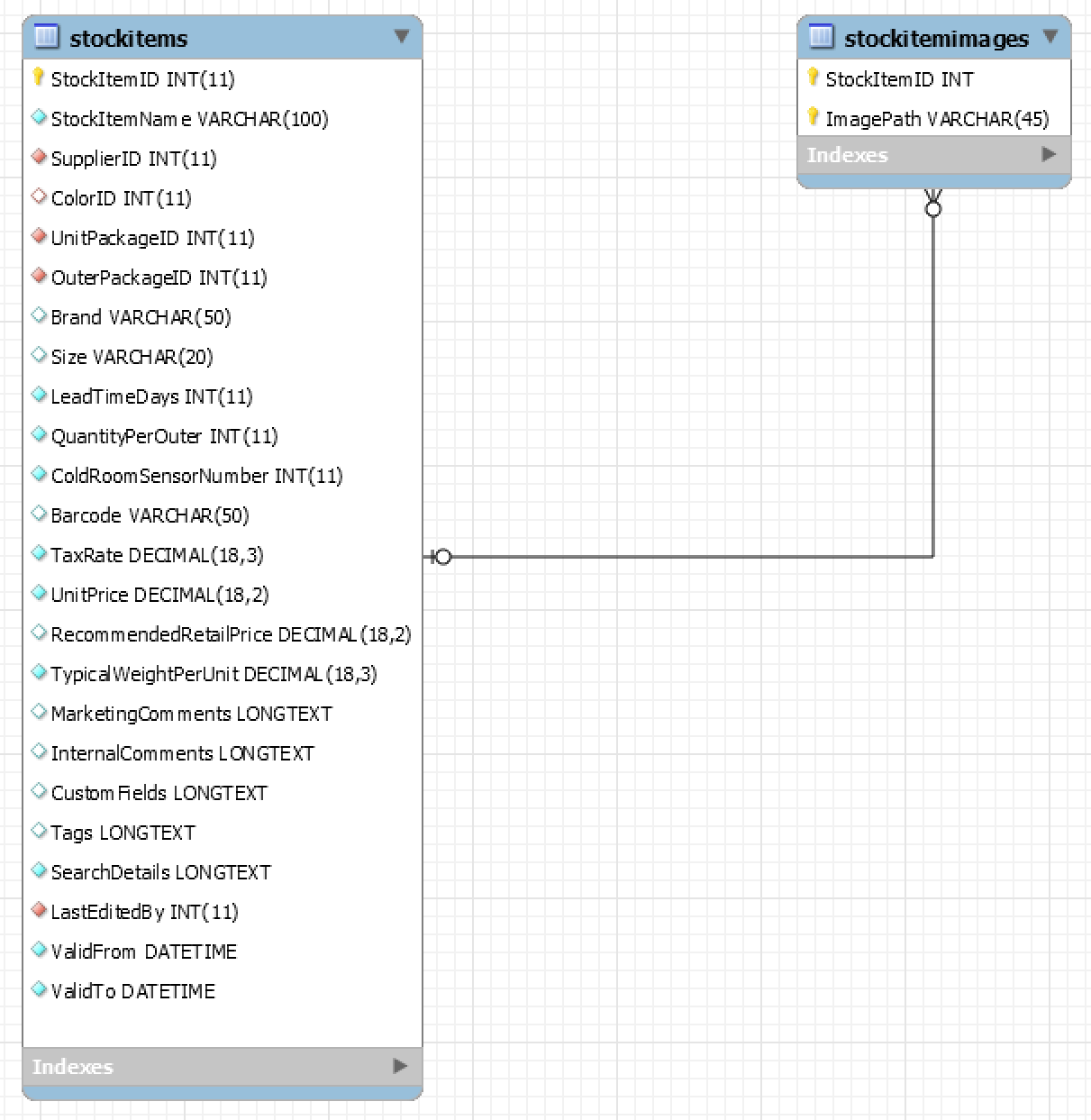
Er kan per product maar één foto toegevoegd worden. Voor de User story Bekijken producten moeten we echter meerdere afbeeldingen laten zien.

**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

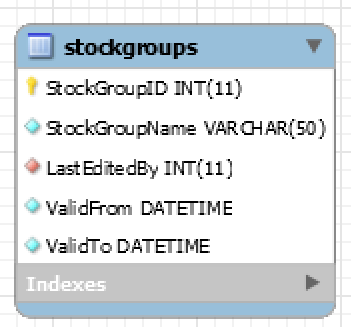
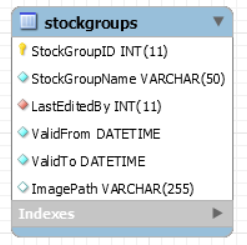
**Waarom de huidige tabelstructuur niet voldoet**

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**



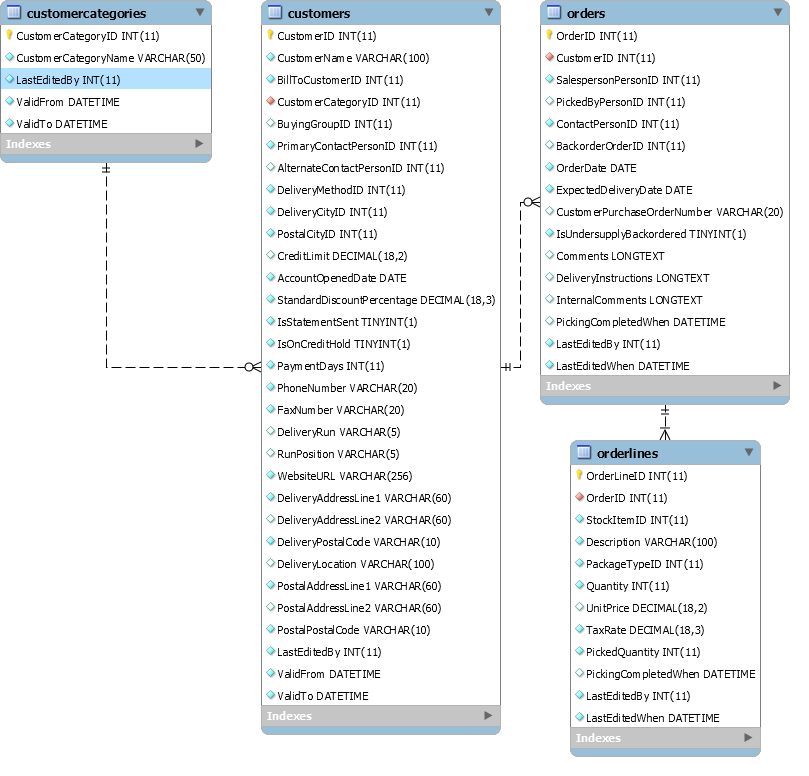
## Bestelling

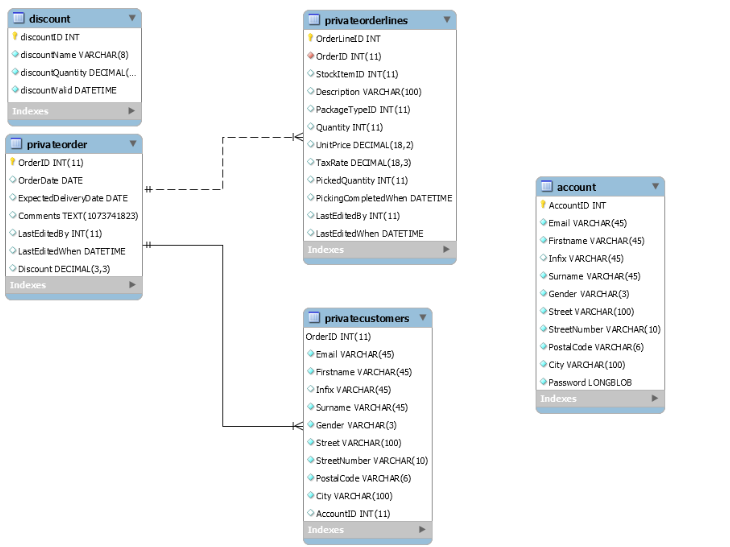
**Waarom de huidige tabelstructuur niet voldoet**

De huidige situatie focust zich op de grotere bedrijven. Wij focussen ons op de particuliere klanten.

**Nieuwe situatie**

Voor de nieuwe situatie hebben wij ongeveer dezelfde tabellen overgenomen en er een paar extra aangemaakt. Deze tabellen hebben wij zo aangepast dat de gegevens kloppen bij onze bestellingen. De oude tabellen bestaan dus nog wel.

**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. De database is op dit moment ook snel genoeg om nog geen gebruik te hoeven maken van indextabellen.

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.

De gebruiker root gaat gedeactiveerd worden, in verband met beveiliging.

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.

Dit is echter niet iets om SQL-injecties tegen te gaan, omdat je alsnog andere tabellen tevoorschijn kan 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

|  |
| --- |
| *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.* |
| *ALTER TABLE account*  *ADD CONSTRAINT Check\_Op\_Postcode CHECK(PostalCode LIKE '[1-9][0-9][0-9][0-9][A-Z][A-Z]' OR PostalCode LIKE '[1-9][0-9][0-9][0-9] [A-Z][A-Z]')*  *ALTER TABLE Check\_Op\_Postcode\_priv\_cust*  *ADD CONSTRAINT Check\_Op\_Postcode CHECK(PostalCode LIKE '[1-9][0-9][0-9][0-9][A-Z][A-Z]' OR PostalCode LIKE '[1-9][0-9][0-9][0-9] [A-Z][A-Z]')* |

## Procedures en functies

Tabel 1: Insert\_Temperatuur\_Archief

|  |
| --- |
| Er wordt procedure aangemaakt voor het automatiseren van het importeren van de oude gegevens van de temperatuursensoren naar het archief. |
| *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

*Beschrijf hier welke triggers zijn toegevoegd. Geef de reden om de trigger aan te maken en het CREATE-statement. De reden kan zijn dat het gaat om een geïmplementeerde maatregel vanuit de securityscan.*

*DWAIJAWD*

Tabel : 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.  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] |
| *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 ;* |

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

# 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.