BRABOTICA WEBSHOP

PLAN VAN AANPAK - Versie 1.3

Auteurs  
Guus Veen & Erwin van Tiel

Inhoudsopgave

[1. Algemene beschrijving 3](#_Toc71196511)

[2. Aanpak 5](#_Toc71196512)

[2.1. Planning 5](#_Toc71196513)

[2.2. Afspraken 5](#_Toc71196514)

[2.3. Taakverdeling 6](#_Toc71196515)

[2.3.1 Rollen 6](#_Toc71196516)

[2.3.2 Taken 6](#_Toc71196517)

[2.4. Eisen 7](#_Toc71196518)

[2.4.1. Algemeen 7](#_Toc71196519)

[2.4.2. Klanten 7](#_Toc71196520)

[2.4.3. Brabotica admins 7](#_Toc71196521)

[2.4.4. Producten 7](#_Toc71196522)

[3. Functionele benodigdheden 8](#_Toc71196523)

[4. Relationeel model & beschrijving (RD) 10](#_Toc71196524)

[4.1. Normalisatie vorm 0 10](#_Toc71196525)

[4.2. Normalisatie vorm 1 10](#_Toc71196526)

[4.3. Normalisatie vorm 2 11](#_Toc71196527)

[4.4. Normalisatie vorm 3 11](#_Toc71196528)

[5. Database Realisatie (DDL) 12](#_Toc71196529)

[5.1. Database aanmaken 12](#_Toc71196530)

[5.2. Rollentabel aanmaken 12](#_Toc71196531)

[5.3. Gebruikerstabel aanmaken 12](#_Toc71196532)

[5.4. Adressentabel aanmaken 13](#_Toc71196533)

[5.5. Orderstabel aanmaken 13](#_Toc71196534)

[5.6. Betalingentabel aanmaken 13](#_Toc71196535)

[5.7. Categorieëntabel aanmaken 14](#_Toc71196536)

[5.8. Productentabel aanmaken 14](#_Toc71196537)

[5.9. Winkelwagenstabel aanmaken 14](#_Toc71196538)

[5.10. Orderregelstabel aanmaken 15](#_Toc71196539)

[5.11. Zoektermentabel aanmaken 15](#_Toc71196540)

[6. Security (DCL) 16](#_Toc71196541)

[6.1. DCL importeren 16](#_Toc71196542)

[6.2. Code van het DCL bestand: 17](#_Toc71196543)

[7. Installatie 19](#_Toc71196544)

[7.1. Stappenplan installatie Database 19](#_Toc71196545)

[7.2. Stappenplan gegevens importeren in de database 20](#_Toc71196546)

[8. Uitwerking van user stories (DML) 22](#_Toc71196547)

[8.1. Registreren 22](#_Toc71196548)

[8.2. Inloggen 22](#_Toc71196549)

[8.3. Persoonsgegevens wijzigen 22](#_Toc71196550)

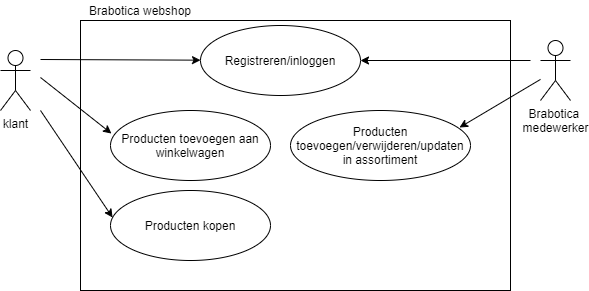
[8.4. Producten toevoegen aan winkelwagen 23](#_Toc71196551)

[8.5. Afrekenen 23](#_Toc71196552)

[Literatuurlijst 24](#_Toc71196553)

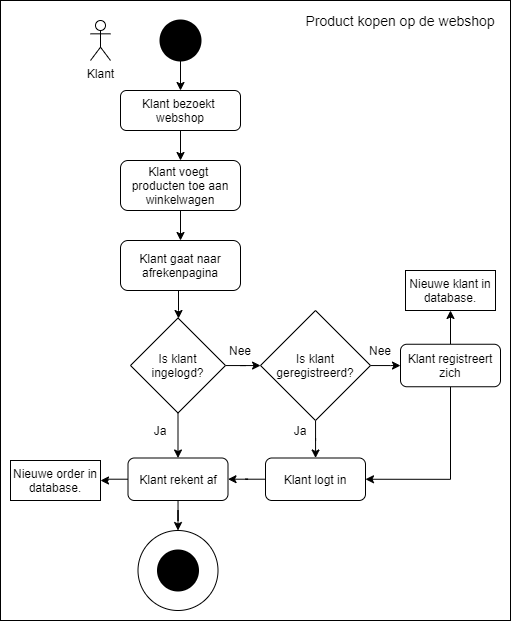
# Algemene beschrijving

Met behulp van diagrammen wordt in dit hoofdstuk een kort overzicht gegeven van de functionaliteit en het doel van de Brabotica-webshop. Er wordt uitgelegd hoe het koopproces er   
voor de klant uitziet en wat de Brabotica-medewerker nodig heeft om de webshop bij te werken.   
Het doel van de webshop van Brabotica is om het voor klanten mogelijk te maken om online producten te kopen.



Figuur 1: Use-case diagram webshop

De klant kan zich op de webshop registeren, inloggen en producten kopen. De Brabotica-medewerker kan zich registeren, inloggen en het assortiment bijwerken.



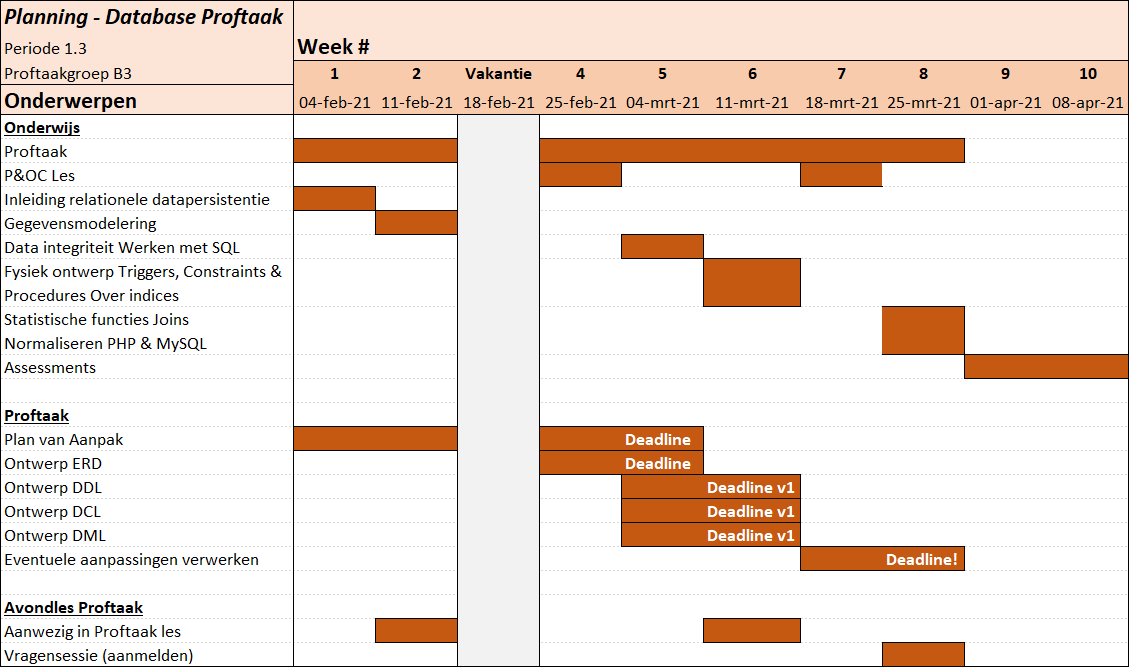
Figuur 2: Activteitendiagram webshop

Voor de eerste versie van de webshop, zullen de klanten alleen producten kunnen kopen als ze zich registreren. In de volgende hoofdstukken wordt toegelicht hoe dit gerealiseerd wordt.

# Aanpak

 In dit hoofdstuk worden de planning, afspraken en Eisen voor het team opgesteld.

### 2.1. Planning



Figuur 3: planning periode 1.3

### 2.2. Afspraken

Hier komen de gemaakte afspraken die we tijdens de meetings hebben besproken.

|  |  |
| --- | --- |
| **Afspraken** | **Motivatie** |
| DrawIO gebruiken voor model | Iedereen heeft DrawIO tot zijn/haar beschikking. Zo kan iedereen de modellen openen en blijven de ontwerpen tot een stijl |
| De naam van de entiteit | De naam van de entiteit in de ERD's wordt in hoofdletters boven de scheidingslijn genoteerd |
| Entiteit meervoud | De naam van de entiteit in de ERD's worden in meervoud genoteerd |
| Attributen | De attributen van de entiteit worden onder de scheidingslijn genoteerd |
| Primaire sleutel | Met PK wordt in de ERD's aangegeven welke attributen de unieke sleutels zijn |
| * Secundaire sleutel | Met FK wordt in de ERD's aangegeven welke attributen de secundaire sleutels zijn |
| Apache-webserver | Iedereen gebruikt XAMPP om de Apache-webserver te gebruiken |
| Versiebeheer | Voor versiebeheer gebruiken we GitHub https://github.com/guusveen/BraboticaDatabasePlan |

Tabel 1: gemaakte afspraken

### 2.3. Taakverdeling

#### 2.3.1 Rollen

|  |  |  |
| --- | --- | --- |
|  | Erwin | Guus |
| teamleider | X |  |
| Devops Engineer | X | X |
| software architect | X | X |
| Notulist |  | X |
| secretaris | X |  |

#### 2.3.2 Taken

**Algemene taken en subtaken**

* Update DB model en documenten(Guus) (Les 3);
  + Update diagrammen;
  + Update plan van aanpak.
* Update functioneel en technisch ontwerp(Erwin) (Les 3) .
  + Maak nieuwe wireframes;
  + Tekst aanpassen zodat het klopt met de huidige modellen;
  + ERD verplaatsen naar technisch ontwerp.

**Webshop taken en subtaken**

* Registratiepagina (Erwin) (Les 5);
  + Maak klasse gebruiker;
  + Maak form met alle nodige velden.;
  + Maak Php file of class die de formdata verwerkt naar db.
* LoginPagina (Erwin) (Les 6) ;
  + Maak form met email en wachtwoord;
  + Maak php file of class die kijkt of user bestaat en wachtwoord klopt en doorstuurt naar de webshop.
* Profielpagina (optioneel);
  + Maak pagina waar gebruiker zijn gegevens kan zien;
  + Maak pagina waar gebruiker zijn bestellinggeschiedenis kan zien (optioneel).
* Productpagina (Guus) (Les 5);
  + Maak klasse product;
  + Maak producten uit db zichtbaar op pagina;
  + Maak filterering met categorieën;
  + Maak zoekfunctie.
* Winkelwagenpagina (?) (Les 6);
  + Maak klasse winkelwagen;
  + Laat alle winkelwagen entries uit db voor gebruiker zien;
  + Laat totaalprijs zien;
  + Maak functionaliteit om meer of minder van een product in je winkelwagen te doen.
* Maak Beheerpagina voor producten (?) (Les 7);
  + Maak functionaliteit om producten toe te voegen in de productentabel;
  + Maak functionaliteit om producten te updaten;
* Beheerpagina voor gebruikers (optioneel);
  + Maak functionaliteit om gebruikers te zoeken;
  + Maak functionaliteit om gebruikers te promoten of demoten (admin / klant).
* Beheerpagina voor zoektermen (?) (Les 7);
  + Maak lijst met laatste 50 zoektermen;
  + Maak filterfuncties (datum of zoekterm bijvoorbeeld).
* Maak Orderpagina (optioneel);
* Contactpagina (optioneel).
  + Maak backendpagina voor ContactPagina;
  + Activeer emailing;
  + Maak opmaak voor te sturen email;
  + Stuur email naar gebruiker on submit.

### 2.4. Eisen

#### 2.4.1. Algemeen

Er moet een productpagina zijn;   
Er moet een beheerpagina zijn.

#### 2.4.2. Klanten

Klanten moeten zich kunnen registreren;   
Klanten moeten in kunnen loggen;   
Klanten moeten productpagina’ s kunnen zien;   
Klanten moeten producten in een winkelwagen kunnen doen;   
Klanten moeten af kunnen rekenen;   
Na afrekenen moet er een order worden geplaatst.

#### 2.4.3. Brabotica admins

Admins moeten in kunnen loggen;   
Admins moeten gebruikers aan kunnen maken;   
Admins moeten gebruikers kunnen promoveren tot admin;   
Admins moeten toegang hebben tot de beheerpagina;   
Admins moeten het assortiment aan kunnen passen op de beheerpagina.

#### 2.4.4. Producten

Producten moeten een prijs hebben;   
Producten moeten een beschrijving hebben;

# Functionele benodigdheden

In dit hoofdstuk worden de meest voorkomende processen van het systeem uitgelegd met behulp van use-case tabellen.

|  |  |
| --- | --- |
| **Naam** | **Registreren** |
| Actor | Klant  Braboticamedewerkers |
| Aannamen | Gebruiker is nog niet geregistreerd. |
| Beschrijving | 1. De gebruiker vult zijn gegevens in 2. Het systeem controleert of het emailadres al in gebruik is en of de rest van de gegevens geldig zijn. 3. Als alles geldig is dan worden de gegevens opgeslagen zodat de gebruiker in kan loggen. Anders komt er een uitzondering [gegevens kloppen niet] of [email in gebruik]. |
| Uitzonderingen | [Gegevens kloppen niet] gebruiker krijgt een melding dat gegevens niet kloppen.  [Email in gebruik] gebruiker krijgt een melding dat email in gebruik is. |
| Resultaat | De inlog-gegevens zijn opgeslagen in de database en de klant kan inloggen met zijn emailadres en wachtwoord. |

Tabel 2: Use-case tabel: registreren

|  |  |
| --- | --- |
| **Naam** | **Inloggen** |
| Actor | Klant  Braboticamedewerkers |
| Aannamen | Gebruiker is geregistreerd. |
| Beschrijving | 1. De gebruiker vult zijn inlog-gegevens in 2. Het systeem controleert of de email + wachtwoordcombinatie bestaat. 3. Zo ja, dan wordt de gebruiker ingelogd. 4. Anders komt er een uitzondering [Email + wachtwoordcombinatie klopt niet] of [email niet gebruik]. |
| Uitzonderingen | [Email + wachtwoordcombinatie klopt niet] gebruiker krijgt een melding dat inlog-gegevens niet kloppen.  [Email niet gebruik] gebruiker krijgt een melding dat email niet in gebruik is en wordt gevraagd of hij zich wil registreren. |
| Resultaat | Gebruiker is ingelogd en krijgt zijn persoonlijke pagina’s te zien. |

Tabel 3: Use-case tabel: inloggen

|  |  |
| --- | --- |
| **Naam** | **Toevoegen/verwijderen uit winkelwagen** |
| Actor | Klant |
| Aannamen | Gebruiker is aangemeld en op de productenpagina. |
| Beschrijving | 1. De gebruiker geeft aan hoeveel van het gekozen product hij toe wil voegen of verwijderen in zijn winkelwagen en bevestigt dat. 2. De producten worden toegevoegd of verwijdert uit zijn winkelwagen. Als er niet genoeg producten op voorraad zijn, komt er een uitzondering [Producten niet op voorraad]. |
| Uitzonderingen | [Producten niet op voorraad] Gebruiker wordt gemeld dat de producten niet op voorraad zijn en er wordt aangegeven wanneer de producten naar schatting geleverd worden. |
| Resultaat | Producten zijn opgeslagen/verwijdert uit winkelwagen van klant. |

Tabel 5: Use-case tabel: toevoegen/verwijderen uit winkelwagen

|  |  |
| --- | --- |
| **Naam** | **Afrekenen** |
| Actor | Klant |
| Aannamen | Gebruiker is aangemeld en heeft producten opgeslagen in zijn winkelwagen. |
| Beschrijving | 1. Gebruiker kan kwantiteit van gekozen producten veranderen voordat hij naar de volgende stap gaat. 2. Gebruiker krijgt details van de factuur te zien en controleert en wijzigt deze waar nodig. 3. Gebruiker kiest betaaloptie. 4. Wordt doorverbonden naar betaling-service en voltooid betaling. Als betaling niet is voltooid komt er een uitzondering [betaling niet voltooid] 5. Wordt terug naar de orderpagina gestuurd met bevestiging dat de betaling is ontvangen. 6. Krijgt email dat betaling is ontvangen en een factuur pdf erbij. 7. Verkoopafdeling krijgt de order binnen en handelen die af. 8. Order wordt opgeslagen in DB en factuur PDF wordt opgeslagen op server. |
| Uitzonderingen | [Betaling niet voltooid] Klant wordt teruggeleid naar de orderpagina met een melding dat de betaling niet is gelukt. |
| Resultaat | Geld is overgemaakt van klant naar Brabotica en verkoopafdeling heeft melding van een bestelling gekregen |

Tabel 6: Use-case tabel: afrekenen

# Relationeel model & beschrijving (RD)

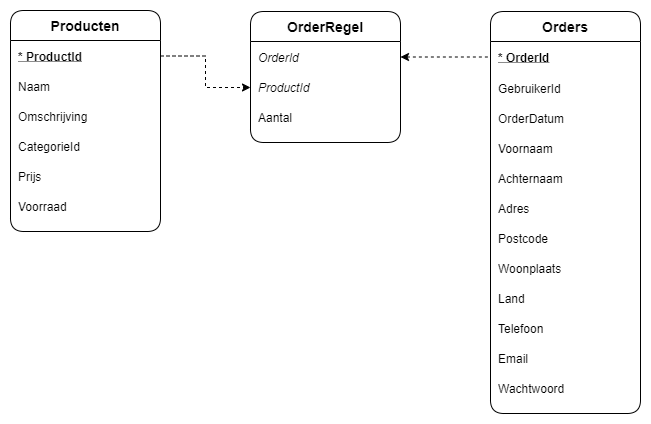
4.1. Normalisatie vorm 0



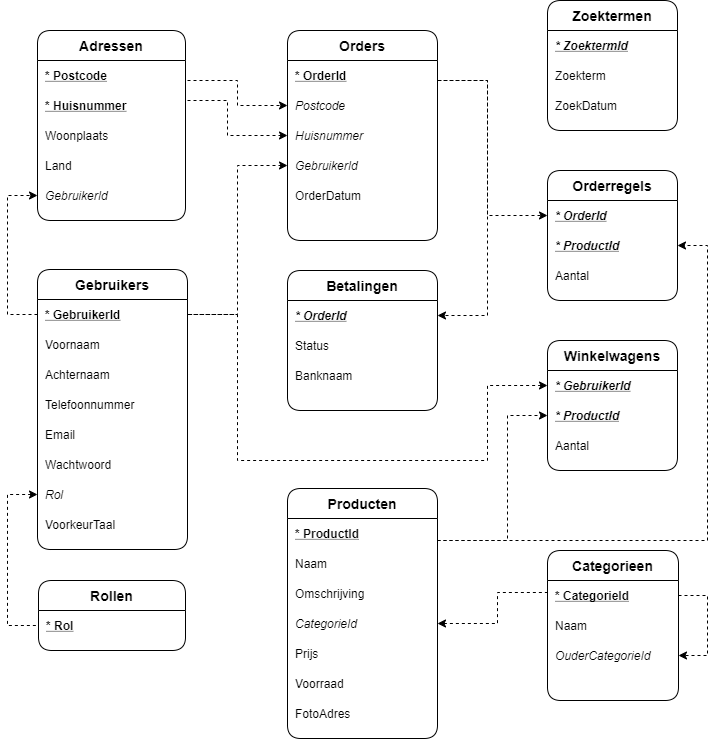
## 4.2. Normalisatie vorm 1



## 4.3. Normalisatie vorm 2



## 4.4. Normalisatie vorm 3



# Database Realisatie (DDL)

In dit hoofdstuk wordt de SQL behandeld die zorgt voor het aanmaken van de database en tabellen.

## 5.1. Database aanmaken

Voordat de tabellen in de database komen, moet er een database zijn. Deze kunt u aanmaken door in PhpMyAdmin aan de linkerkant “New” aan te klikken en dan in de SQL-tab de volgende code in te typen.

**CREATE DATABASE** BraboticaDB;

## 5.2. Rollentabel aanmaken

Deze tabel moet als eerst aangemaakt worden omdat de gebruikerstabel gebruik maakt van de rollentabel. De SQL is als volgt.

**CREATE TABLE** Rollen (  
Rol VARCHAR(255) **NOT** **NULL**,  
**PRIMARY** **KEY** (Rol)  
);

Rollen zijn altijd uniek en worden gebruikt als primary key.

Nu is ook een goed moment om de beschikbare rollen te vullen

**INSERT** **INTO** Rollen Values ('Klant'), ('Medewerker'), ('Admin');

## 5.3. Gebruikerstabel aanmaken

Hier volgt de SQL voor de gebruikerstabel. Omdat de adressentabel de gebruikerstabel nodig heeft, wordt de gebruikerstabel eerst aangemaakt.

**CREATE** **TABLE** Gebruikers (  
GebruikerId INT(10) **NOT** **NULL** AUTO\_INCREMENT,  
Voornaam VARCHAR(255) **NOT** **NULL**,  
Achternaam VARCHAR(255) **NOT** **NULL**,  
Telefoonnummer VARCHAR(15),  
Email VARCHAR(255) **NOT** **NULL**,  
Rol VARCHAR(255) **NOT** **NULL** **DEFAULT** 'Klant',  
Wachtwoord VARCHAR(255) **NOT** **NULL**,  
VoorkeurTaal VARCHAR(10) **NOT** **NULL** **DEFAULT** 'NL',  
**PRIMARY** **KEY** (GebruikerId),  
**FOREIGN** **KEY** (Rol) **REFERENCES** Rollen(Rol),  
**UNIQUE** (Email)  
);

Gebruikers hebben een rol uit de rollentabel en het emailadres moet uniek zijn.

## 5.4. Adressentabel aanmaken

Hier volgt de adressentabel.

**CREATE** **TABLE** Adressen (  
Postcode VARCHAR(10) **NOT** **NULL**,  
Huisnummer INT(5) **NOT** **NULL**,  
Land VARCHAR(255) **NOT** **NULL**,  
GebruikerId INT(10) **NOT** **NULL**,  
**PRIMARY** **KEY** (Postcode, Huisnummer),  
**FOREIGN** **KEY** (GebruikerId) **REFERENCES** Gebruikers(GebruikerId)  
);

De primary key bestaat uit de postcode en het huisnummer omdat die samen uniek zijn. Een adres heeft altijd een gebruiker.

## 5.5. Orderstabel aanmaken

Orders worden aangemaakt zodra een klant aangeeft dat hij de inhoud van zijn winkelwagen af wil rekenen.

**CREATE** **TABLE** Orders (  
OrderId INT(10) **NOT** **NULL** AUTO\_INCREMENT,  
Postcode VARCHAR(10) **NOT** **NULL**,  
Huisnummer INT(5) **NOT** **NULL**,  
GebruikerId INT(10) **NOT** **NULL**,  
Orderdatum DATETIME **NOT** **NULL,**  
**PRIMARY** **KEY** (OrderId),  
**FOREIGN** **KEY** (Postcode, Huisnummer) **REFERENCES** Adressen(Postcode, Huisnummer),  
**FOREIGN** **KEY** (GebruikerId) **REFERENCES** Gebruikers(GebruikerId)  
);

Omdat het adres van een order anders kan zijn dan het adres van de klant, zijn er foreign keys van de adressentabel toegevoegd.

## 5.6. Betalingentabel aanmaken

Zodra een klant zijn betalinggegevens heeft ingevoerd, wordt er een betaling aangemaakt die bij de order hoort. Zodra de betaling binnen is, gaat de status naar ‘voltooid’.

**CREATE** **TABLE** Betalingen (  
OrderId INT(10) **NOT** **NULL,**  
Banknaam VARCHAR(255) **NOT** **NULL**,  
Status VARCHAR(255) **NOT** **NULL**,  
**PRIMARY** **KEY** (OrderId),  
**FOREIGN** **KEY** (OrderId) **REFERENCES** Orders(OrderId)  
);

## 5.7. Categorieëntabel aanmaken

Categorieën kunnen categorieën hebben. Dat wil zeggen dat een categorie verbonden kan zijn aan een 'oudercategorie'.

**CREATE** **TABLE** Categorieen (  
CategorieId INT(10) **NOT** **NULL** AUTO\_INCREMENT,  
Naam VARCHAR(255) **NOT** **NULL**,  
ouderCategorie INT(10),   
**PRIMARY** **KEY** (CategorieId)  
);  
**ALTER** **TABLE** Categorieen   
**ADD** **FOREIGN** **KEY** (ouderCategorie) **REFERENCES** Categorieen(CategorieId);

De foreign key kan tijdens de tabel creatie niet aangemaakt worden omdat de tabel nog niet bestaat. Daarom wordt er na de tabel creatie een alter table operatie uitgevoerd die de foreign key aan zichzelf verbindt.

## 5.8. Productentabel aanmaken

Hier volgt de productentabel.

**CREATE** **TABLE** Producten (  
ProductId INT(10) **NOT** **NULL** AUTO\_INCREMENT,  
Naam VARCHAR(255) **NOT** **NULL**,  
Omschrijving VARCHAR(255) **NOT** **NULL**,  
CategorieId INT(10) **NOT** **NULL**,  
Prijs DECIMAL(10,2) **NOT** **NULL**,  
Voorraad INT **DEFAULT** 0,  
FotoAdres VARCHAR(255),  
**PRIMARY** **KEY** (ProductId),  
**FOREIGN** **KEY** (CategorieId) **REFERENCES** Categorieen(CategorieId)  
);

## 5.9. Winkelwagenstabel aanmaken

Als een klant een nieuw soort product in zijn winkelwagen plaatst, wordt er een rij in deze tabel gevuld. Zodra een klant afrekent, worden alle rijen met de GebruikerId van de klant verwijdert uit deze tabel.

**CREATE** **TABLE** Winkelwagens (  
GebruikerId INT(10) **NOT** **NULL**,  
ProductId INT(10) **NOT** **NULL**,  
Aantal INT(10) **NOT** **NULL**,  
**PRIMARY** **KEY** (GebruikerId, ProductId),  
**FOREIGN** **KEY** (GebruikerId) **REFERENCES** Gebruikers(GebruikerId),  
**FOREIGN** **KEY** (ProductId) **REFERENCES** Producten(ProductId)  
);

Een winkelwagenrij kan uniek geïdentificeerd worden door de GebruikerId en de ProductId.

## 5.10. Orderregelstabel aanmaken

Zodra een klant een order plaatst, worden alle winkelwagen rijen die bij de klant horen, gekopieerd naar de orderregelstabel. Het verschil is dat de GebruikerId in deze tabel wordt vervangen door de OrderId.

**CREATE** **TABLE** Orderregels (  
OrderId INT(10) **NOT** **NULL**,  
ProductId INT(10) **NOT** **NULL**,  
Aantal INT(10) **NOT** **NULL**,  
**PRIMARY** **KEY** (OrderId, ProductId),  
**FOREIGN** **KEY** (OrderId) **REFERENCES** Orders(OrderId),  
**FOREIGN** **KEY** (ProductId) **REFERENCES** Producten(ProductId)  
);

Een OrderRegelRij kan uniek geïdentificeerd worden door de OrderId en de ProductId.

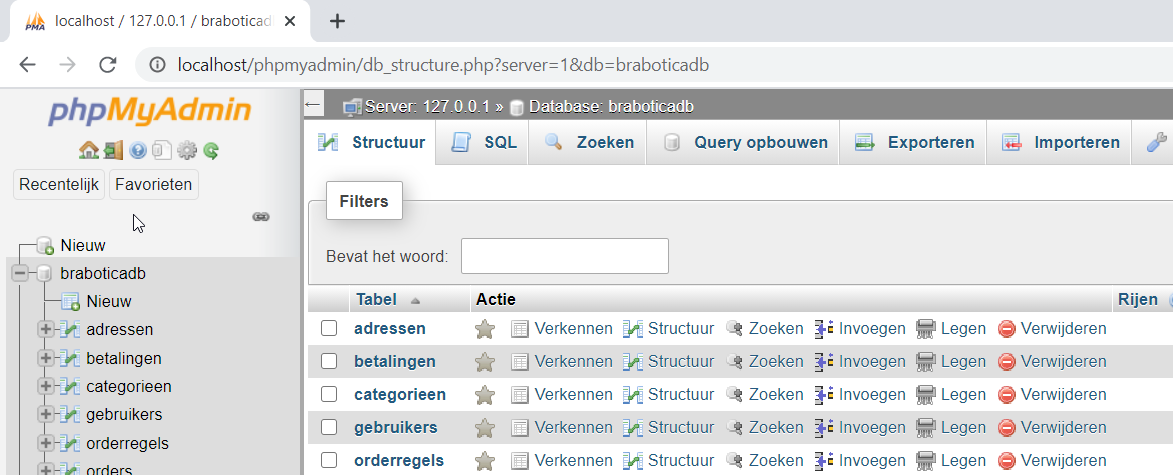
## 5.11. Zoektermentabel aanmaken

Hier volgt de zoektermentabel.

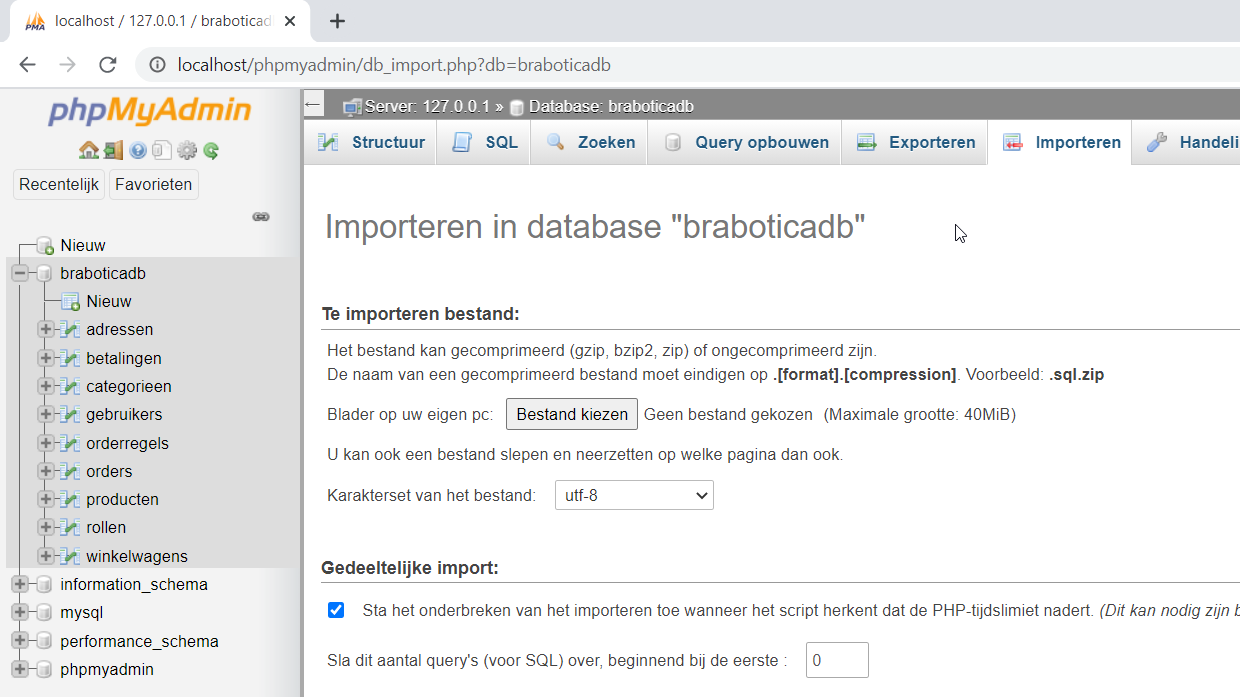
**CREATE** **TABLE** Zoektermen (  
ZoektermId INT(10) **NOT** **NULL**,  
ZoekTerm VARCHAR(255) **NOT** **NULL**,  
Zoekdatum DATETIME **NOT** **NULL**,  
**PRIMARY** **KEY** (ZoekTermId)  
);

# Security (DCL)

6.1. DCL importeren  
De DCL kunt u eenvoudig importeren via het meegeleverde ‘BraboticaDCL.sql’ bestand.

1. Klik op het menu ‘Importeren’.
2. Klik op de button ‘Bestand kiezen’

Selecteer het bestand ‘BraboticaDCL.sql’ en klik op openen.



1. Klik rechts onderin op de button ‘Starten’ om het geselecteerde .sql bestand te

Importeren.

1. Er wordt nu op het scherm aangegeven dat het importeren is geslaagd.

6.2. Code van het DCL bestand:   
  
*--*

*-- Maak gebruiker Klant aan --*

*--*

**drop** **user** if exists 'Klant'@'localhost';

**CREATE** **USER** 'Klant'@'localhost' IDENTIFIED BY 'GeheimeWachtwoord';

*--*

*-- Geef de gebruiker 'klant' lees schrijf en update rechten op bepaalde velden in de tabel adressen --*

*--*

**GRANT** **SELECT** (`Postcode`, `Huisnummer`, `Land`), **INSERT** (`Postcode`, `Huisnummer`, `Land`), **UPDATE** (`Postcode`, `Huisnummer`, `Land`)

**ON** `braboticadb`.`adressen` **TO** 'Klant'@'localhost';

*--*

*-- Geef de gebruiker 'klant' lees schrijf en update rechten op bepaalde velden in de tabel gebruikers --*

*--*

**GRANT** **SELECT** (`Voornaam`, `Achternaam`, `Telefoonnummer`, `Email`, `Wachtwoord`), **INSERT** (`Voornaam`, `Achternaam`, `Telefoonnummer`, `Email`, `Wachtwoord`), **UPDATE** (`Voornaam`, `Achternaam`, `Telefoonnummer`, `Email`, `Wachtwoord`)

**ON** `braboticadb`.`gebruikers`

**TO** 'Klant'@'localhost';

*--*

*-- Geef de gebruiker 'klant' lees rechten op bepaalde velden in de tabel orders --*

*--*

**GRANT** **SELECT** (`Postcode`, `Huisnummer`, `Orderdatum`)

**ON** `braboticadb`.`orders`

**TO** 'Klant'@'localhost';

*--*

*-- Geef de gebruiker 'klant' lees schrijf en wijzig rechten op de tabel winkelwagens --*

*--*

**GRANT** **SELECT**, **INSERT**, **UPDATE**

**ON** `braboticadb`.`winkelwagens`

**TO** 'Klant'@'localhost';

*--*

*-- Geef de gebruiker 'klant' lees rechten op de tabel producten --*

*--*

**GRANT** **SELECT**

**ON** `braboticadb`.`producten`

**TO** 'Klant'@'localhost';

*--*

*-- Maak gebruiker Medewerker aan --*

*--*

**drop** **user** if exists 'Medewerker'@'localhost';

**CREATE** **USER** 'Medewerker'@'localhost' IDENTIFIED BY '20Br@MeBrbo';

*--*

*-- Geef de gebruiker 'Medewerker' lees schrijf wijzig en delete rechten op de alle tabellen in de database --*

*--*

**GRANT** **SELECT**, **INSERT**, **UPDATE**, **DELETE**

**ON** `braboticadb`.\*

**TO** 'Medewerker'@'localhost' **WITH GRANT** OPTION;

*--*

*-- Maak gebruiker Admin aan --*

*--*

**drop** **user** if **exists** 'Admin'@'localhost';

**CREATE** **USER** 'Admin'@'localhost' IDENTIFIED BY 'Ad20@MinBro';

*--*

*-- Geef de gebruiker 'Admin' alle rechten voor de database braboticadb --*

*--*

**GRANT** ALL PRIVILEGES

**ON** `braboticadb`.\*

**TO** 'Admin'@'localhost' **WITH** **GRANT** OPTION;

# Installatie

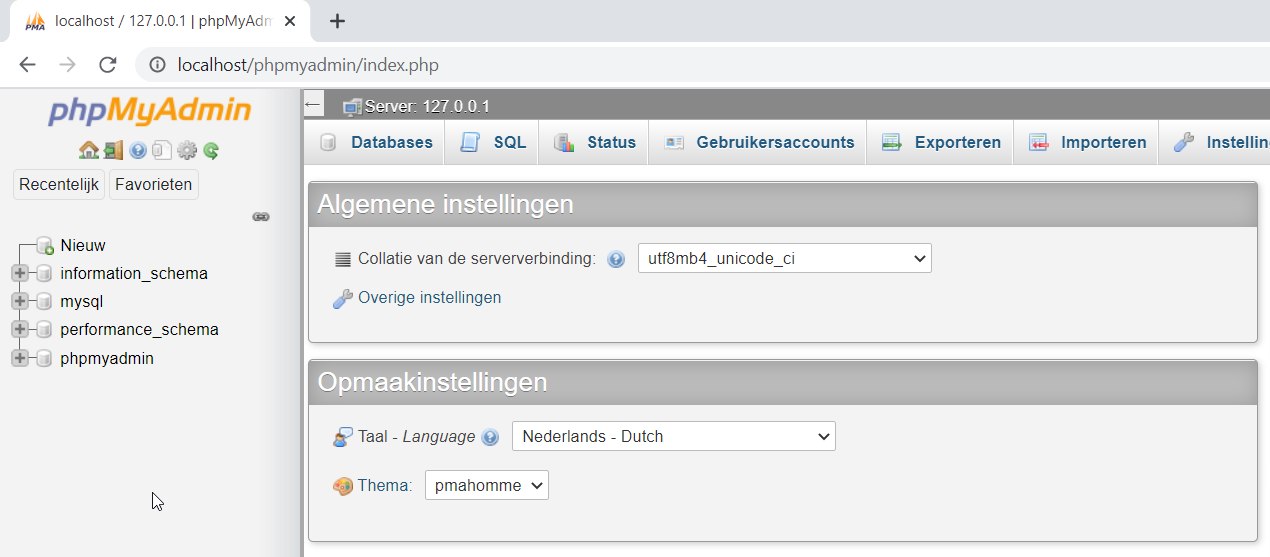
## 7.1. Stappenplan installatie Database

1. Open het bestand ‘Installatie\_Database\_Brabotica.zip’

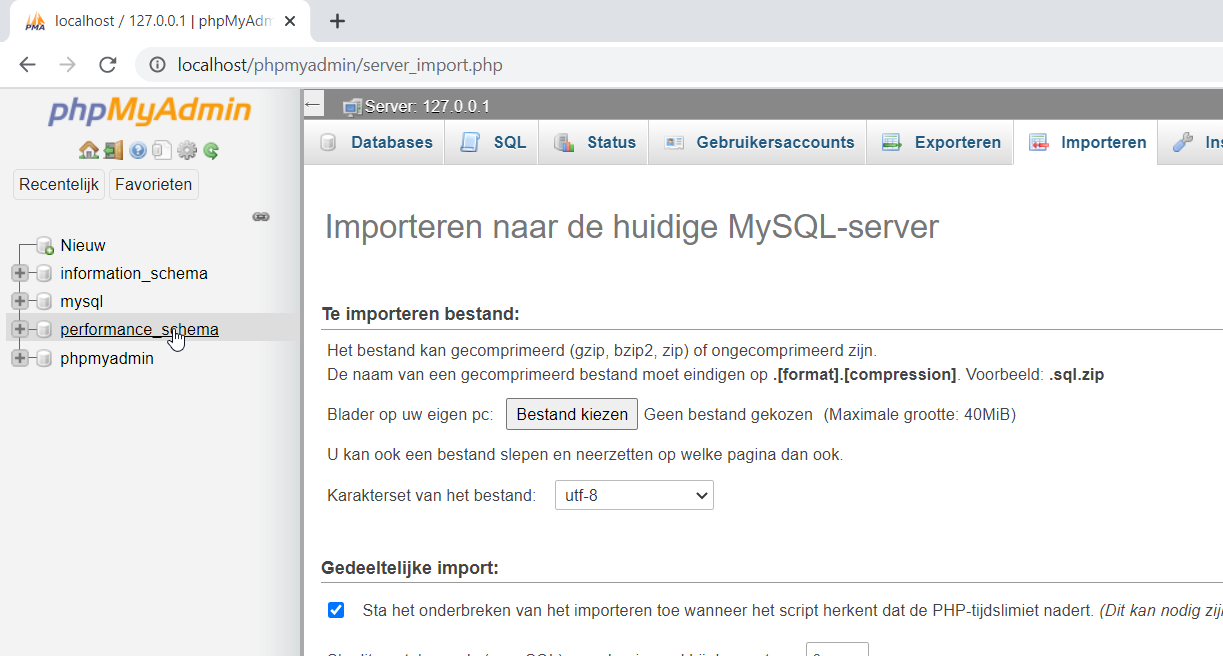
Pak het ingepakte .zip bestand uit naar gewenste locatie en open hierna de locatie.

1. Open de webbrowser en ga naar phpMyAdmin.

<http://localhost/phpmyadmin/index.php>

1. Klik in het menu op Importeren.

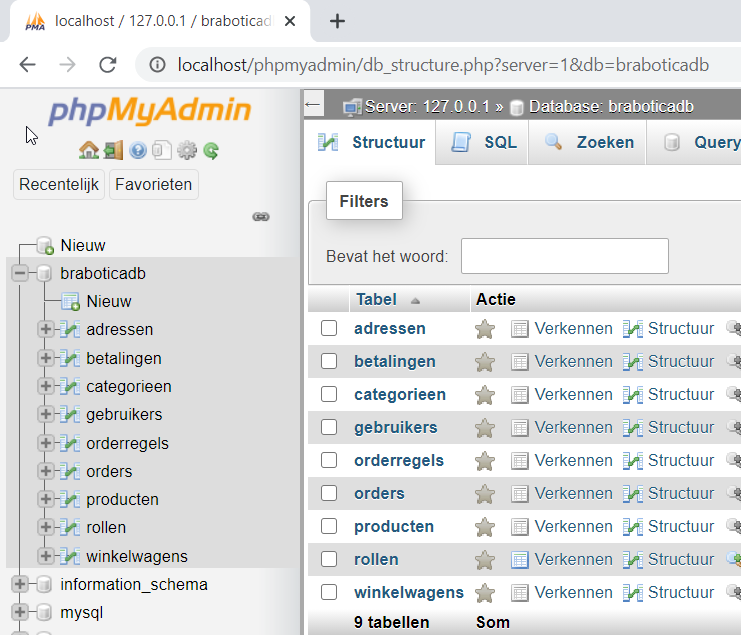
1. Klik op de button ‘Bestand kiezen’

Er wordt nu een nieuw venster geopend, ga in dit venster naar de map waar het zip bestand bij stap 1 is uitgepakt. Selecteer het bestand ‘BraboticaDDL.sql’   
en klik op openen.  


1. Klik rechts onderin op de button ‘Starten’ om het geselecteerde .sql bestand te

Importeren.

1. Er wordt nu op het scherm aangegeven dat het importeren is geslaagd.

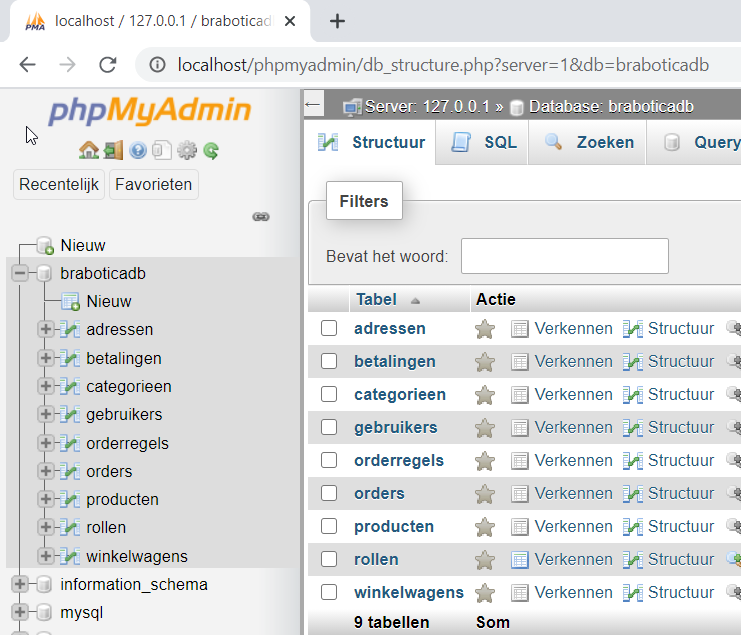
Aan de linkerzijde wordt de database ‘Braboticadb’ weergeven.  


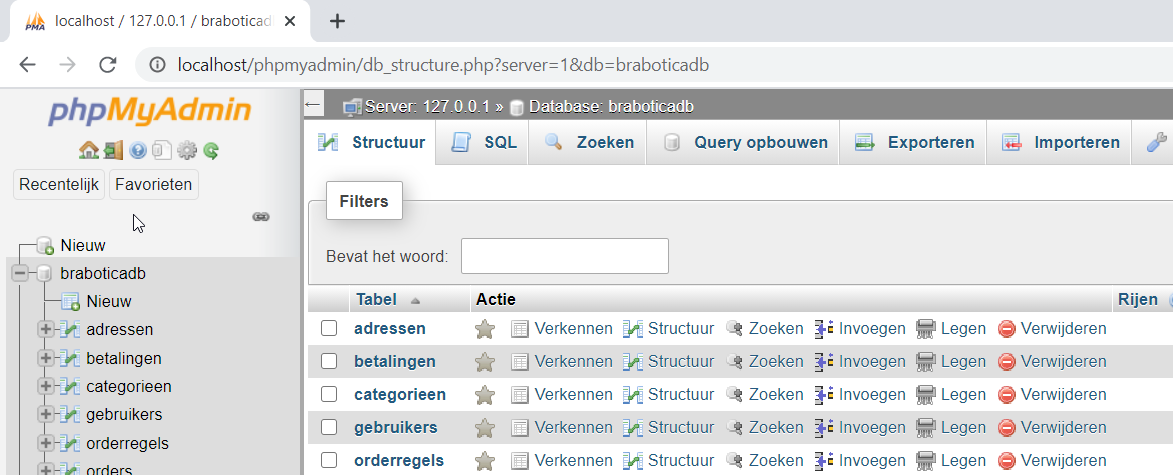
## 7.2. Stappenplan gegevens importeren in de database

1. Open de webbrowser en ga naar phpMyAdmin.

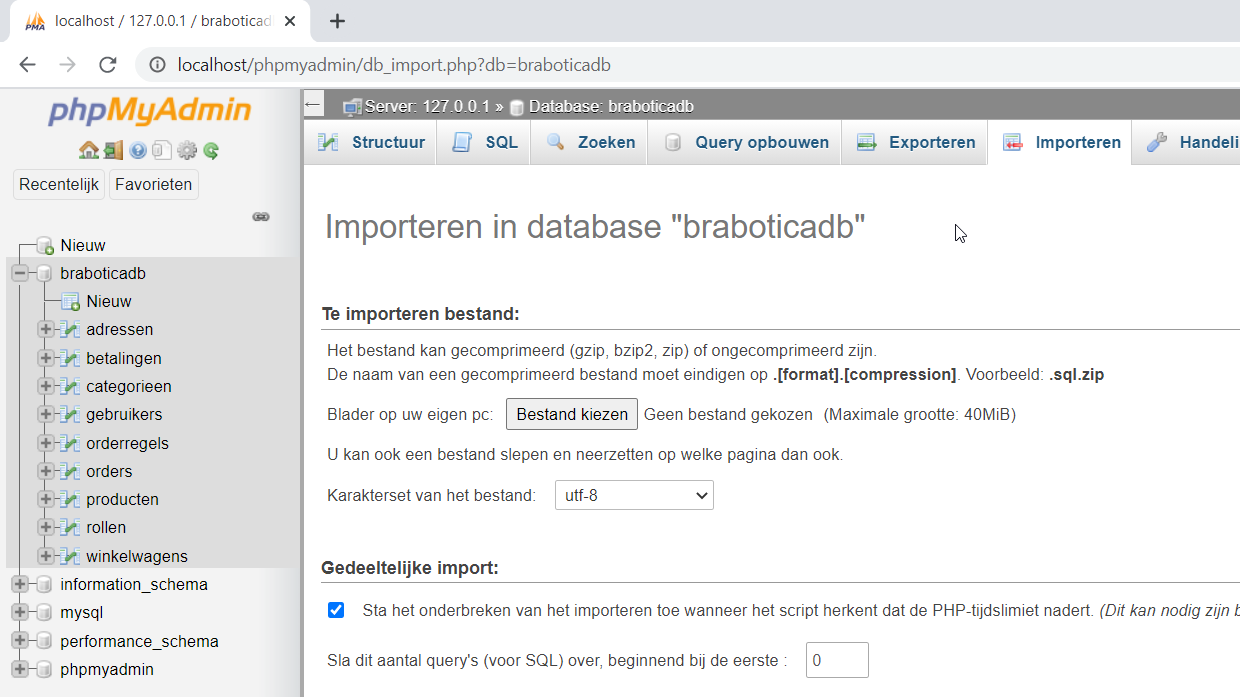
<http://localhost/phpmyadmin/index.php>

1. Klik aan de linkerzijde op de database ‘BraboticaDB’ zodat u de database opent.



1. Klik op het menu ‘Importeren’.
2. Klik op de button ‘Bestand kiezen’

Er wordt nu een nieuw venster geopend, ga in dit venster naar de map waar het zip bestand bij stap 1 is uitgepakt. Selecteer het bestand ‘Data\_Import\_BraboticaDB.sql’   
en klik op openen.



1. Klik rechts onderin op de button ‘Starten’ om het geselecteerde .sql bestand te

Importeren.

1. Er wordt nu op het scherm aangegeven dat het importeren is geslaagd.

Aan de linkerzijde wordt de database ‘Braboticadb’ weergeven.

# Uitwerking van user stories (DML)

In dit hoofdstuk worden de SQL-operaties die bij de user stories horen, gedemonstreerd.

## 8.1. Registreren

Voordat een klant een product kan bestellen op de webshop, moet hij zich registreren. Een gebruiker registreert zich door zijn gegevens in te vullen in de form op de registratiepagina. De SQL-query die nodig is om de ingevulde gegevens op te slaan ziet er als volgt uit.

**INSERT** **INTO** Gebruikers (   
Voornaam, Achternaam, Telefoonnummer, Email, Wachtwoord)  
**VALUES** (  
'John', 'Doe', '0612345678', 'JohnDoe@voorbeeld.nl', 'test');

De default rol van gebruikers is ‘klant’. De rol kan zelf aangegeven worden door de Rol column bij de query toe te voegen.

**INSERT** **INTO** Gebruikers (  
 Voornaam, Achternaam, Telefoonnummer, Email, Wachtwoord, Rol)  
**VALUES** (  
 'John', 'Doe', '0687654321', 'JohnDoe@voorbeeld.nl', 'test', 'Admin');

## 8.2. Inloggen

Nadat een gebruiker geregistreerd is, kan hij inloggen. Op de inlogpagina vult de gebruiker zijn email en wachtwoord in. Het wachtwoord moet gelijk zijn aan het wachtwoord dat bij het ingevulde emailadres hoort in de Gebruikerstabel. De query om het wachtwoord bij het bijbehorende emailadres op halen ziet er als volgt uit.

**SELECT** Email, Wachtwoord **FROM** Gebruikers  
**WHERE** Email='janedoe@voorbeeld.nl';

De wachtwoorden zijn in deze versie nog niet gehasht.

## 8.3. Persoonsgegevens wijzigen

Het kan voorkomen dat de gebruiker verhuist of om een andere reden zijn persoonsgegevens aan wil passen. De gebruiker doet dit door op de persoonsgegevenspagina een form in te vullen met de gewenste gegevens. De SQL-query ziet er als volgt uit.

**UPDATE** Gebruikers  
**SET** Voornaam = 'Jane',   
Achternaam = 'Smith',   
Telefoonnummer = '0612345678',   
Email = 'janedoe@voorbeeld.nl',   
Wachtwoord = 'test'  
**WHERE** GebruikerId = 1;

In de webshopapp wordt de waarde van GebruikerId vervangen met de GebruikerId uit de sessie van de gebruiker.

## 8.4. Producten toevoegen aan winkelwagen

Klanten kunnen producten toevoegen aan hun winkelwagen als ze op een knop drukken. De query ziet er als volgt uit.

**INSERT** **INTO** Winkelwagens (GebruikerID, ProductId, Aantal)  
**VALUES** (1, 1, 1);

De GebruikerId wordt vervangen door de GebruikerId in de sessievariable en de ProductId door de ProductId van het geklikte product. Er is ook een optie om het aantal mee te geven.

Als een klant het aantal van een gekozen product wil wijzigen ziet de query er als volgt uit.

**UPDATE** Winkelwagens  
**SET** Aantal = 2  
**WHERE** GebruikerID = 1 **and** ProductId = 1;

## 8.5. Afrekenen

Zodra een klant de inhoud van zijn winkelwagen af wil rekenen, wordt er eerst een order aangemaakt. De OrderId van de gemaakte entry wordt opgeslagen in de PHP-code voor later gebruik.

**INSERT** **INTO** Orders (GebruikerID, Postcode, Huisnummer)  
**VALUES** (1, '5600aa', 1);

Daarna worden de producten uit de winkelwagen naar de orderregelstabel gekopieerd. Hiervoor moeten eerst de producten uit de Winkelwagens worden opgehaald.

**SELECT** ProductId, Aantal **FROM** Winkelwagens  
**WHERE** GebruikerID = 1;

Nu kunnen de orderregels gevuld worden met de opgehaalde rijen uit Winkelwagens.

**INSERT** **INTO** Orderregels(OrderId, ProductId, Aantal)  
**VALUES** (1, 1, 1);

Nu kunnen de klanten hun bank kiezen en de betalinglink volgen om af te rekenen. Als alles goed is verlopen, wordt de betaling aangemaakt met ‘voltooid’ status, anders wordt de betaling ook aangemaakt maar met de 'geen betaling' status.

**INSERT** **INTO** Betalingen (OrderId, Banknaam, Status)  
**VALUES** (1, 'Rabobank', 'voltooid');

Zodra de status van de betaling op 'voltooid' staat, is de bestelling compleet en wordt het product verstuurd naar de klant.

Er is nog één ding dat moet gebeuren en dat is de winkelwagen leegmaken. Alle Winkelwagen entries van de klant worden verwijderd.

**DELETE** **FROM** Winkelwagens  
**WHERE** GebruikerID = 1;

# Literatuurlijst

Gillenson, M. L. (2004). Inleiding Database managementsystemen (1ste editie). Academic Service.  
  
<https://www.w3schools.com>