**Vak:** DMDD

**Docent:** Matthijs de Jonge

**Door:** Lauren Meenhorst SN: 632206 & Martijn Hogendoorn SN: 634495

**Klas:** ITA-ISE-A-F

**Datum:** 12 maart 2021

**Versie:** 1

*Modular synth store*

Technisch ontwerp

Inhoudsopgave

[Inleiding 2](#_Toc66452823)

[1. PDM 2](#_Toc66452824)

[1.1 Tabel en kolom beschrijvingen 2](#_Toc66452825)

[1.2 Gemaakte keuzes 6](#_Toc66452826)

[1.2.1 Subtypen 6](#_Toc66452827)

[1.2.2 Fabrikanten 6](#_Toc66452828)

[1.2.3 Productvarianten 6](#_Toc66452829)

[1.3 Integriteitsregels 7](#_Toc66452830)

[2. Bijlagen 8](#_Toc66452831)

[2.1 Bijlage 1: PDM 8](#_Toc66452832)

[2.2 Bijlage 2: SQL Query’s 9](#_Toc66452833)

# Inleiding

De casus ERM gaat over het bedrijf Modular Synth Store, afgekort MSS.

MSS is een winkel die modular synth artikelen en accessoires verkoopt.

Hierbij is het belangen alleen bij het in winkel verkoop, en niet het online verkoop.

Er is gewenst om een databasesysteem te creëren voor MSS die verschillende informatie vasthoudt.

Zo wil de winkel haar verkoopmedewerkers ondersteunen bij het opzoeken van informatie over de producten, klanten, verkoopgeschiedenis, etc. van de winkel.

Aan ons dus de opdracht om hier een informatiesysteem voor te ontwerpen.

In het Functioneel ontwerp werd beschreven wat het systeem zal gaan inhouden/doen.

In het Technisch ontwerp wordt beschreven hoe de inhoud/functies van het systeem tot stand zullen komen.

Normaliter worden software architectuur en GUI ontwerpen hierin opgenomen. Dit is bij deze casus niet van toepassing. Dit omdat het systeem enkel gebruikt wordt om informatie op te slaan en niet om business processen te automatiseren.

# 1. PDM

In het Functioneel ontwerp kwam uit een feitanalyse een Conceptueel Data Model tot stand. Dit model is hier omgezet naar een Fysiek Data Model (PDM). Vanuit dit model ontstaan scripts die het informatiesysteem implementeren.

Diagram

Description automatically generated

## 1.1 Tabel en kolom beschrijvingen

Een korte omschrijving van de tabellen die gevonden kunnen worden in het CDM

|  |  |
| --- | --- |
| **Klant** | **Een persoon die een aankoop heeft gedaan bij MSS** |
| Klantnummer | Een uniek nummer dat de klant identificeert |
| Voornaam | De voornaam van de klant |
| Achternaam | De achternaam van de klant |
| Telefoonnummer | Het telefoonnummer waarop MSS de klant kan bereiken |

|  |  |
| --- | --- |
| **Factuur** | **Een samenstelling van alle Factuurregels binnen 1 transactie** |
| Factuurnummer | Een uniek nummer dat de transactie identificeert |
| Klantnummer | De klant die de aankoop heeft gedaan |
| Datum | De datum waarop de transactie heeft plaatsgevonden |

|  |  |
| --- | --- |
| **Factuurregel** | **Het aantal eenheden van een product dat tegen een bepaalde prijs deel uit hebben gemaakt van een factuur**  **(Koppeltabel tussen Product en Factuur)** |
| Factuurnummer | Het unieke nummer van de factuur in kwestie |
| Productcode | De code van het aangeschafte product |
| Prijs | De verkoopprijs op het moment van de transactie |
| Aantal | Het aantal aangeschafte eenheden van het product |
| Productvariant | De variant van het aangeschafte product |

|  |  |
| --- | --- |
| **Factuurregel\_Variant** | **Het aantal eenheden van een productvariant dat tegen een bepaalde prijs deel uit hebben gemaakt van een factuur**  **(Koppeltabel tussen Productvariant en Factuur)** |
| Productvariant | De aangeschafte variant van het product met varianten |
| Productcode | De code van het aangeschafte product met varianten |
| Factuurnummer | Het unieke nummer van de factuur in kwestie |
| Prijs | De verkoopprijs op het moment van de transactie |
| Aantal | Het aantal aangeschafte eenheden van het product |

|  |  |
| --- | --- |
| **Product** | **De gegevens van een product** |
| Productcode | De unieke code van een product dat geen varianten heeft |
| Productnaam | De naam van het product in kwestie |
| Verkoopprijs | De huidige verkoopprijs van het product |
| ProductBeschrijving | De beschrijving van het product in kwestie |
| Inkoopprijs | De huidige inkoopprijs van het product |
| IsSet | Of het product een set is of niet |

|  |  |
| --- | --- |
| **Productvoorraad** | **De voorraad van een product** |
| Productcode | De code van het product in kwestie |
| Aantal | Het aantal beschikbare eenheden van het product |

|  |  |
| --- | --- |
| **Productvariantvoorraad** | **De voorraad van een productvariant** |
| Productvariant | De variant van het product met varianten |
| ProductMetVarCode | Het product met varianten |
| Aantal | Het aantal beschikbare eenheden van de variant |

|  |  |
| --- | --- |
| **ProductSetRegel** | **Het aantal eenheden van een product dat deel uitmaakt van een bepaalde product set.**  **(Koppeltabel tussen Product en Product)** |
| ProductSetCode | De productcode van het product dat een product set is |
| Productcode | De code van het product dat deel uitmaakt van de product set |
| Aantal | Het aantal eenheden van product dat deel uitmaakt van de set |

|  |  |
| --- | --- |
| **Productvariant** | **Varianten van een Product met varianten** |
| ProductMetVarCode | De unieke code van een product dat varianten heeft |
| Productvariant | De variant van het product met varianten |
| Productnaam | De naam van de variant |
| Inkoopprijs | De huidige inkoopprijs van de variant |
| Verkoopprijs | De huidige verkoopprijs van de variant |

|  |  |
| --- | --- |
| **ProductMetHP** | **De horizontal pitch waarde van producten van het type case/module.** |
| Productcode | De code van het product in kwestie |
| HorizontalPitch | De horizontal pitch waarde van het product |

|  |  |
| --- | --- |
| **Module** | **Een product van het type module** |
| Productcode | De code van het product in kwestie |
| FunctieNaam | De naam van de functie van het product |

|  |  |
| --- | --- |
| **Functie** | **Een domein tabel met hierin alle mogelijke functies van een modulle** |
| FunctieNaam | De naam van de functie |

|  |  |
| --- | --- |
| **ProductType** | **Duidt aan van welk type een product is**  **(Koppeltabel tussen Product en Type)** |
| Productcode | De code van het product in kwestie |
| TypeNaam | De naam van het type van het product |

|  |  |
| --- | --- |
| **Type** | **Een domein tabel met hierin alle mogelijke typen van een product** |
| TypeNaam | De naam van het type |

|  |  |
| --- | --- |
| **Fabrikant\_Product** | **De fabrikant van een product**  **(Koppeltabel tussen Product en Fabrikant)** |
| Productcode | De code van het product in kwestie |
| Fabrikantnummer | Het nummer van de fabrikant in kwestie |

|  |  |
| --- | --- |
| **Fabrikant\_Product** | **De fabrikant van een product**  **(Koppeltabel tussen Product en Fabrikant)** |
| Productvariant | De variant van het product met varianten |
| ProductMetVarCode | De unieke code van een product dat varianten heeft |
| Fabrikantnummer | Het nummer van de fabrikant in kwestie |

|  |  |
| --- | --- |
| **Fabrikant** | **Een bedrijf dat producten publiceert en levert aan MSS** |
| Fabrikantnummer | Een uniek nummer waarmee de fabrikant geïdentificeerd wordt |
| Fabrikantnaam | De naam van de fabrikant |
| Straatadres | Het straatadres van de klant |
| Postcode | De postcode van de fabrikant |
| Plaats | De plaats waar het hoofdkantoor van de fabrikant gevestigd is |
| Land | Het land waar het hoofdkantoor van de fabrikant gevestigd is |
| Telefoonnummer | Het telefoonnummer van de fabrikant |
| Faxnummer | Het faxnummer van de fabrikant |
| Mailadres | Het unieke emailadres van een fabrikant |

|  |  |
| --- | --- |
| **Land** | **Een domein tabel met landen in de wereld** |
| Land | De naam van het land in de wereld |

## 1.2 Gemaakte keuzes

Er zijn een aantal keuzes gemaakt bij het omzetten van het CDM naar het PDM. Dit hoofdstuk dient deze keuzes te onderbouwen.

### 1.2.1 Subtypen

Het subtype ProductSet bleek overbodig te zijn omdat het geen attributen had. De overervingsrelatie naar product is verplaats naar de tabel ProductSetRegel en de tabel ProductSet is verwijderd.

Tussen Module en Functie kwam een koppeltabel Module\_Functie. Dit had als resultaat dat Module enkel een Productcode en mogelijke variant als attribuut had. Om ruimte te besparen is ervoor gekozen de functie van een module in de Module tabel te verwerken.

Tot slot is ervoor gekozen de parent en children van subtypen in apparte tabellen op te nemen.

### 1.2.2 Fabrikanten

Er is voor gekozen niet de Boyce-CODD normaalvorm toe te passen op de Fabrikant tabel. Hier is voor gekozen omdat er dan rekening gehouden moet worden met veel buitenlandse manieren om adressen op te nemen. Gezien de tijd was dit niet haalbaar.

Ook is er door een probleem in powerdesigner voor gekozen gekozen met varchars te werken, ondanks dat de fabrikanten in meerdere landen gevestigd kunnen zijn en uni code wenselijk is.

Wij raden aan in deze tabel nvarchars te gebruiken.

### 1.2.3 Productvarianten

Eerder in het FO werd al vermeld dat de productvarianten een obstakel zijn geweest binnen ons team.

Er is op het laatste moment een relatie ontstaan tussen Module en Productvariant. Er is echter geen relatie tussen productvarianten en andere types. Dit bedenkend dat een Productvariant enkel een module kan zijn. Er is op dit moment geen communicatie meer mogelijk met de domeinexpert. Vandaar dat wij de aanname maken dat de relatie zo klopt.

Ook kunnen alle varianten van een product met varianten momenteel maar door 1 fabrikant geleverd worden. Dit is nogmaals een aanname.

Indien er meer tijd was geweest voor overleg zouden wij hier graag meer aandacht aan besteden.

## 1.3 Integriteitsregels

Om te zorgen dat het informatiesysteem schoon blijft zullen een aantal business rules verwerkt moeten worden.

**IR-1:**

**Correspondeert met BR1**

Specificatie: Als een product van het type case of module is, moet er voor dit product een Horizontal Pitch waarde worden opgenomen.

Implementatie: stored proc sp\_ModOrCaseThenHP

**IR-2:**

**Correspondeert met BR2**

Specificatie: Als een product van het type module is moet de functie van de module worden opgenomen.

Implementatie: stored proc sp\_FunctieAlsModule

**IR-3:**

**Correspondeert met BR3**

Specificatie: Een product kan een Productvariant, een ProductMetHP/Productset of een Product zijn, nooit iets tegelijk.

Implementatie: check ck\_NieuwProduct

**IR-4:**

**Correspondeert met BR4**

Specificatie: Een product dat een variant heeft kan nooit een losstaand product zijn.

Implementatie: stored proc sp\_InsProdInFac

**IR-5:**

**Correspondeert met BR**

Specificatie: Een factuur zal altijd minimaal 1 regel moeten hebben van of Factuurregel\_Product of Factuurregel\_Variant

Implementatie: stored proc sp\_InsFactuur

# 2. Bijlagen

## 2.1 Bijlage 1: PDM

Diagram

Description automatically generated

## 2.2 Bijlage 2: SQL Query’s

**Integrity rules MSS TO/FO**

Dit bestand is er voor bedoeld om de integrity rules die in het TO staan te testen, wanneer deze business rules daadwerkelijk geimplementeerd worden.

Integrity rule 1: Als een product van het type case of module is, moet er voor dit product een Horizontal Pitch waarde worden opgenomen.

Exec sp\_ModOrCaseThenHP

@Productcode = 123123,

@Productnaam = 'Oscillator mod',

@inkoopprijs = 50.20

@verkoopprijs = 75.00

@Beschrijving = 'Mooie oscillator',

@Inset = 0

@Type = 'Module',

@HPWaarde = 0

Uitkomst:

Er wordt gecontroleerd op type van het product, wanneer deze type case of module is, zal de HP waarde opgenomen moeten worden.

De HP waarde wordt gecontroleerd en krijgt foutmelding omdat deze niet de juiste waarde heeft.

Integrity rule 2: Als een product van het type module is moet de functie van de module worden opgenomen.

exec sp\_FunctieAlsModule

@Productcode = 123123,

@Functie = 'case'

Uitkomst:

Toegevoegd product wordt gecontroleerd op type product. Type product is module, dus zal er een valide functie in meegegeven moeten worden.

Functie is niet valide dus verschijnt er een foutmelding.

Integrity rule 3: Een product kan een Productvariant, een ProductMetHP/Productset of een Product zijn, nooit iets tegelijk.

insert into TYPE(

1,

'Oscillator')

insert into TYPE(

2,

'Case'

)

insert into PRODUCT values(

123123,

'Oscillator mod',

50.20,

75.00,

'Mooie oscillator',

0)

insert into PRODUCT\_TYPE(

123123,

1)

insert into PRODUCT\_TYPE(

123123,

2

)

Uitkomst:

Product wordt toegevoegd, product wordt neer gezet als ProductMetHP.

Foutmelding ontstaat wanneer product nog een producttype krijgt.

Integrity rule 4: Een product dat een variant heeft kan nooit een losstaand product zijn.

exec ps\_InsProdInFac

@Oroductcode = 123123,

@ProductVariant = '',

@Factuurnummer = 1 ,

@Aantal = 1

Uitkomst:

Product wordt toegevoegd aan factuur, krijgt foutmelding als het product een variant heeft maar deze niet benoemd wordt.

Integrity rule 5: Een factuur zal altijd minimaal 1 regel moeten hebben van of Factuurregel\_Product of Factuurregel\_Variant

exec sp\_InsFactuur

@Factuurnummer = 1

Uitkomst:

Factuur wilt toegevoegd worden maar heeft geen regel van of Factuurregel\_Product of Factuurregel\_Variant

Er verschijnt een foutmelding dat het minimaal 1 regel moet bevatten.