Database

Verantwoording Datatypes

Yakup Küçük -

Procesbegeleider

Maria Boes-Voet -

Professional Skills

Jorg Visch -

Productbegeleider

*Pr-IP-P4/15 2018*

E. Alper 613525

D. Hengeveld 616743

P. Ismaiel 619856

S. Kurtovic 555081

B. Slijkhuis 619105

W. Smeltink 604792

Inhoudsopgave

[Inleiding 3](#_Toc11225199)

[Veranderingen 3](#_Toc11225200)

[Lengte 3](#_Toc11225201)

[Char 3](#_Toc11225202)

[Numeric 3](#_Toc11225203)

[Datum/tijd 4](#_Toc11225204)

[Tabellen 4](#_Toc11225205)

# Inleiding

Bij het maken van de database is gebruik gemaakt van het opgeleverde document “EenmaalAndermaal Appendix D - LRS.doc”. Hierin staat aangegeven hoe de database gebouwd moet worden, er staat onder andere in hoe de tabellen, kolommen, primary keys en foreign keys gemaakt moeten worden. Ook staan de datatypen van elke kolom erbij, deze zijn echter niet ideaal. Wij hebben toestemming van iConcepts om zelf te beslissen welke datatypes veranderd moeten worden. In dit document geven wij onze verantwoording bij elke verandering. Alle veranderingen zijn bijgehouden in een los document “Database veranderde data types.docx”.

# Veranderingen

## Lengte

Bij een groot aantal kolommen vonden wij dat de gegeven maximum lengte te kort zou zijn om nuttig te zijn in gebruik, zo stonden er bijvoorbeeld voor voornaam maar 5 karakters. Om deze reden hebben we bij veel kolommen de maximum lengte verhoogd.

## Char

In de meeste gevallen staat er char() als datatype, dit datatype is bedoelt voor data waarvan je zeker bent hoe lang het is. Denk bijvoorbeeld aan een cel waar enkel een ‘M’ of een ‘F’ in komt te staan. Dit zal altijd 1 karakter zijn en dus kan de database deze cellen sneller lezen door char() te gebruiken.

In de meeste gevallen weet je echter niet hoe lang de inhoud van een cel is, dit is dubbel waar in gevallen waar een end-user zelf data kan invullen. Als iemand zijn telefoonnummer ergens in moet vullen, kan hij dit doen als ‘0612345678’, of ’06-12345678’, of zelfs ‘+31 6 12345678’. In deze gevallen kun je niet gebruik maken van de snellere datatype char() en zul je gebruik moeten maken van da datatype varchar(), dit datatype kan namelijk wél verschillende lengtes tekst bewaren in een kolom.

Om deze redenen zijn er een groot aantal kolommen waarbij wij de datatypen van char() naar varchar() hebben veranderd.

## Numeric

De variabele numeric() kan decimale getallen opslaan. In gevallen waar geen decimalen nodig waren hebben wij het datatype veranderd naar integers, dit zijn waardes zonder decimalen. De kleinste is tinyint, deze gaat van 0 tot 255 en is in de meeste gevallen voldoende. Er zijn ook smallint (-32,768 tot 32,767), int (-2,147,483,648 tot 2,147,483,647) en bigint. Bigint is echter zo groot dat hij niet nodig is geweest.

Behalve integers zijn er ook andere datatypes voor getallen, er zijn datatype money en smallmoney. Deze lijkt op het eerste gezicht perfect voor het opslaan van geld, echter blijkt dat deze datatype niet goed werkt in de praktijk. We hebben voor geld dus gebruik gemaakt van numeric() met 2 decimalen.

Ten slotte is er nog een datatype bit dat gebruikt wordt als boolean, de cel kan alleen een 0 (false) of een 1 (true) bevatten.

## Datum/tijd

Voor datum en tijd stond in de appendix ook char aangegeven, er zijn echter datatypes die speciaal gemaakt zijn om data en tijd in op te slaan, dit zijn ‘date’ en ‘time’. Deze hebben wij dus ook toegepast in de database.

## Tabellen

In de tabel rubriek hebben wij de kolom Rubriek hernoemd naar Hoofdrubriek, deze naam is namelijk duidelijker.