

6.5 Relatieve datamodel

Database

Een database, gegevensbank of databank is een digitaal opgeslagen archief. De gegevens zijn opgeslagen op een dusdanige manier dat flexibele raadpleging en gebruik eenvoudig zijn. Databases spelen een belangrijke rol voor het archiveren en actueel houden van gegevens. Ze zijn een essentieel onderdeel van de informatiemaatschappij, steeds meer gegevens worden in een database opgeslagen. Het functioneren van de overheid, financiële instellingen, bedrijven en wetenschap is tegenwoordig zonder databases ondenkbaar.

Steeds meer gegevens worden ook via internet bereikbaar gemaakt. Zowel commercieel als in de wetenschap worden databases veel gebruikt. In de wetenschap gebeurt dat om meetgegevens of experimentele gegevens in op te slaan. Commercieel gaat het vaak om klant- en betalingsgegevens.

'A World of Internet of Things' is een nieuwe realiteit die het vergaren en beheren van gegevens nog intenser maakt. De volgende clip geeft een idee van waar we over een paar jaar kunnen zijn.



Databases worden steeds belangrijker. Een goede database moet aan de volgende minimale CRUD- eisen voldoen om als zodanig te worden gezien:

- Create: gegevens moeten eenvoudig duurzaam kunnen worden opgeslagen.
- Read: gegevens moeten eenvoudig kunnen worden opgezocht en doorzocht.
- Update: gegevens moeten onderhouden kunnen worden.
- Delete: gegevens moeten verwijderd kunnen worden zonder dat dat de werking van het systeem nadelig beïnvloedt.

Met andere woorden: *de database moeten integer zijn*. Dat wil zeggen dat:

- gegevens niet dubbel mogen worden opgeslagen (redundant);
- de samenhang of relatie met andere gegevens moet kloppen (consistent).

Relationele database

Een relationele database is een database die is opgebouwd volgens een relationeel model.

De gegevens worden opgeslagen in tabellen waarin de rijen de soortgelijke groepen informatie (de records) vormen en de kolommen de informatie die voor elk record moet worden opgeslagen.

Als voorbeeld gebruiken we de inkoop van een elektronicawinkel.

Artikel

ARTIKEL	OMSCHRIJVING	STUKPRIJS
VID34	DVD Arphils	1250
TFN3110	Telefoon India	199
RDO24	Radio type 24	20

Leverancier

CODE	LEVERANCIER	ADRES	WOONPLAATS
PH	Phillips	Blauwgeel	Eindhoven
LG	LG	Dalton Street	Londen

Verschillende tabellen kunnen met elkaar worden verbonden door een kolom toe te voegen waarin een verwijzing naar een record in een andere tabel wordt opgenomen.

ARTIKEL	OMSCHRIJVING	STUKPRIJS	LEV_CODE
VID34	DVD Arphils	1250	PH
TFN3110	Telefoon India	199	LG
RDO24	Radio type 24	20	PH

CODE	LEVERANCIER	ADRES	WOONPLAATS
PH	Phillips	Blauwgeel	Eindhoven
LG	LG	Dalton Street	Londen

Wanneer de gegevens in een relationele database goed gestructureerd zijn, wordt duplicatie van gegevens tot een minimum beperkt en fouten in de gegevensverwerking worden voorkomen.

Normaliseren

Waarom normaliseren?

Een database is een verzameling gegevens. Zonder structuur is het niet meer dan een opeenhoping van data. Zonder structuur zal het bijna zeker zo zijn dat dezelfde gegevens meerdere keren in de database opgenomen zijn. Hierdoor zullen fouten ontstaan tijdens invoegen, updaten of verwijderen van gegevens. De database

is dan *niet integer*. Dat is niet wat we willen. Het kan ook worden voorkomen door relaties in de database aan te brengen en ervoor te zorgen dat gegevens niet redundant, maar eenmalig worden opgenomen. Een database goed ontwerpen doen we via een normalisatieproces.

Attributen en entiteiten

In een database gebruiken we een aantal benamingen voor de verschillende onderdelen. Om met beroepsgenoten op gelijk niveau te kunnen spreken is het belangrijk dat jij die termen ook begrijpt en gebruikt. Redundant en consistent zijn net al langsegekomen. Een object wordt beschreven door zijn kenmerken. Een mes zou uit de volgende kenmerken kunnen bestaan:

Merk: Sola

Type: Broodmes

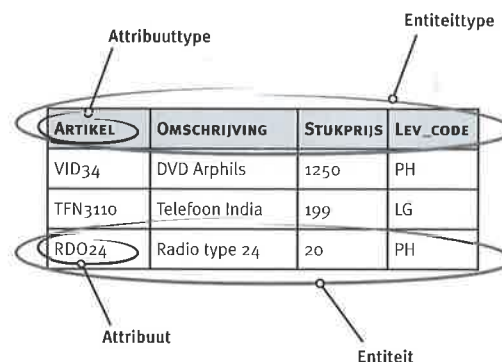
Lemmetlengte: 32 cm

Attribuut en attribuuttype

- Merk is het attribuuttype
- Sola is het attribuut

Entiteiten en entiteittype

- De groep: Merk, Type en Lemmetlengte noemen we het entiteittype
- Sola, Broodmes, 32cm zijn de entiteiten



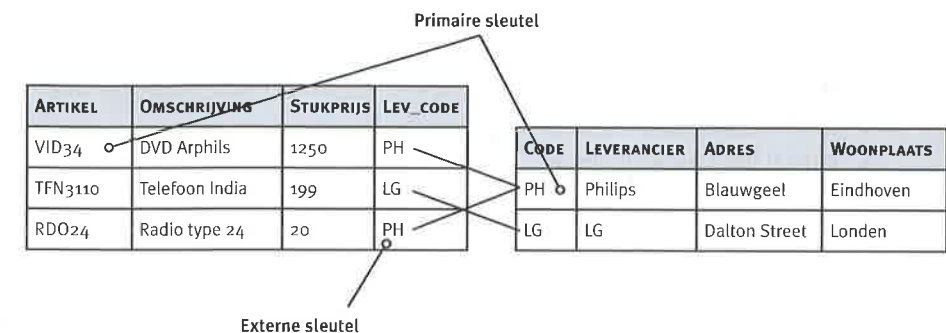
Primaire en externe sleutels

Verschillende entiteiten (tabellen) kunnen een relatie met elkaar hebben. Deze relatie wordt via een sleutel aangegeven. Om elke entiteit te kunnen aanwijzen is een primaire sleutel nodig. Elke entiteit heeft precies één primaire sleutel. Een primaire sleutel kan bestaan uit een speciaal attribuut, het sleutelattribuut, of uit een verzameling van sleutelattributen. Kijken we nog een keer naar de relatie tussen de apparatuur en de leverancier dan zie je dat de leveranciercode PH in het Artikeloverzicht twee keer voorkomt. In het leveranciersoverzicht is PH echter uniek. In het leveranciersoverzicht is de Code een primaire sleutel. De Lev_code in het

artikeloverzicht wijst naar de sleutel in het leveranciersoverzicht en wordt externe sleutel genoemd.

ARTIKEL	OMSCHRIJVING	STUKPRIJS	LEV_CODE	CODE	LEVERANCIER	ADRES	WOONPLAATS
VID34	DVD Arphils	1250	PH	PH	Philips	Blauwgeel	Eindhoven
TFN3110	Telefoon India	199	LG	LG	LG	Dalton Street	Londen
RDO24	Radio type 24	20	PH	LG	LG	Dalton Street	Londen

Een externe sleutel is een attribuut uit een entiteit die zelf niet een sleutelattribuut is maar wel verwijst naar een sleutelattribuut in een andere entiteit. Een externe sleutel koppelt dus alleen entiteiten met elkaar.



Een externe sleutel wordt ook wel vreemde sleutel of foreign key genoemd. Het veld Lev_code in de tabel Artikel wijst naar de primaire sleutel in de tabel Leverancier en heet daarom externe sleutel.

Normaliseren volgens Codd

r. Edgar F. Codd kwam in 1970 met een methode om relationele databases te ontwerpen. De methode van Codd voor het normaliseren van gegevens is een soort standaard geworden. De methode bestaat uit een voorbereidende fase, de zogenaamde 'nulte normaalvorm', en drie stappen om uiteindelijk in de 'derde normaalvorm' te komen. Heel erg kort samengevat zijn er de volgende normaalvormen te onderscheiden:

- Nulte normaalvorm (0NV): inventariseer de attributen;
- Eerste normaalvorm (1NV): splits repeterende groep af;
- Tweede normaalvorm (2NV): splits groepen met gedeeltelijk afhankelijke attributen af;
- Derde normaalvorm (3NV): splits de groepen met onafhankelijke attributen af.

Aan de hand van een voorbeeld voor de inschrijving op een cursus door een cursist gaan we de stappen doornemen.

ATV sch 701		INSCHRIJVING		14FEB16	
STUDENTNR: 1234		Naam:		Rheenen J.C. van	
		Adres:		Hooicamp 9	
		Woonplaats:		Houten	
CURSUSCODE	CURSUSNAAM	STARTDATUM	GEBOUW	LOCATIE	GEBOUWNAAM
SO	Systeemontw.	12 mei 16	2	Utrecht	Paviljoen
SA	Systeemanalyse	21 jun 16	1	Utrecht	Toren
PR	Programmeren	11 aug 16	7	Enschede	Bastille
IA	Informatieanalyse	16 okt 16	4	Zwolle	Tuinschuur

Dit formulier geeft de informatie die we moeten ontleiden.

Nulde normaalvorm (oNV)

e nulde normaalvorm krijg je door alle attributen op te schrijven (inventariseren) die uit de analyse van de informatie behoefte komt. De te nemen stappen zijn:

- Inventariseer alle mogelijke attribuuttypen
- Bepaal de constanten en procesgegevens, streep de procesgegevens weg
- Geef aan het entiteitstype een betekenisvolle naam
- Bepaal het sleutelattribuut
- Bepaal of er een repeterende groep voorkomt

Inventariseer de attribuuttypen

ATV sch 701		INSCHRIJVING		14FEB16	
STUDENTNR: 1234		Naam:		Rheenen J.C. van	
		Adres:		Hooicamp 9	
		Woonplaats:		Houten	
CURSUSCODE	CURSUSNAAM	STARTDATUM	GEBOUW	LOCATIE	GEBOUWNAAM
SO	Systeemontw.	12 mei 16	2	Utrecht	Paviljoen
SA	Systeemanalyse	21 jun 16	1	Utrecht	Toren
PR	Programmeren	11 aug 16	7	Enschede	Bastille
IA	Informatieanalyse	16 okt 16	4	Zwolle	Tuinschuur

ATV sch701
 14FEB16
 STUDENTNR
 Naam
 Adres
 Woonplaats
 CURSUSCODE
 CURSUSNAAM
 STARTDATUM
 GEBOUW
 LOCATIE
 GEBOUWNAAM

Bepaal de constanten en procesgegevens, streep de procesgegevens weg

In de bovenstaande lijst zijn ATV sch 701 en 14FEB16 doorgehaald. De schoolcode zal voorgedrukt zijn op het formulier en niet in de database staan. De datum zal door het systeem worden gegenereerd en is in die hoedanigheid een procesgegeven. Deze gegevens doen verder ook niet meer mee.

Geef aan het entiteitstype een betekenisvolle naam

Het gaat om een Student met zijn inschrijving voor cursussen. De naam Student ligt voor de hand.

STUDENT
Studentnr
Naam
Adres
Woonplaats
Cursuscode
Cursusnaam
Startdatum
Gebouw
Locatie
Gebouwnaam

Bepaal het sleutelattribuut

Het sleutelattribuut maakt een entiteit uniek. Het gegeven dat dit formulier uniek maakt in de stapel formulieren is het studentnummer. Daarom is het Studentnr aangeduid als sleutel door het te onderstrepen.

Bepaal of er een repeterende groep voorkomt

We zien dat student Van Rheenen meerdere cursussen volgt. Deze starten op verschillende dagen in verschillende gebouwen. De Attribuuttypen CURSUSCODE, CURSUSNAAM, STARTDATUM, GEBOUW, LOCATIE en GEBOUWNAAM worden bij herhaling gebruikt. Daarom noemen we dit een repeterende groep. Die duiden we aan door hem te laten inspringen.

De nulde normaalvorm ziet er dan als volgt uit:



Eerste normaalvorm (1st NV)

De te nemen stappen zijn:

- Bepaal of de attribuuttypen nog verder opgesplitst moeten worden
- Splits de repeterende groep af
- Geef aan het afgesplitste entiteitstype een betekenisvolle naam
- Maak van de sleutel een primaire sleutel

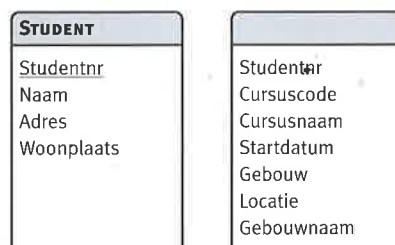
Bepaal of de attribuuttypen nog verder opgesplitst moeten worden

Als we naar de nulde normaalvorm kijken zouden we de attribuuttypen adres, woonplaats en datum nog kunnen splitsen in adres+huisnummer, postcode+woonplaats en dag+maand+jaar. Is het nuttig om deze attribuuttypen verder te splitsen omdat we de verschillende onderdelen gaan gebruiken? Zo ja, dan kunnen we het attribuuttype verder opdelen.

In ons inschrijvingsformulier levert het verder splitsen van deze attribuuttypen geen voordeel op. Daarom laten we ze staan zoals ze zijn.

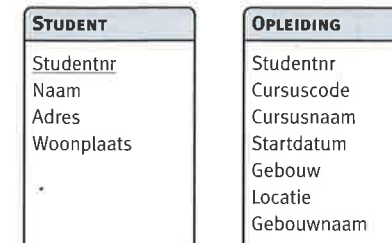
Splits de repeterende groep af

We splitsen de repeterende groep af in een eigen entiteitstype. Omdat in de oorspronkelijke entiteit de sleutel door het sleutelattribuut Studentnr bepaald wordt nemen we deze over. De nieuwe entiteit is op deze manier gekoppeld aan de eerste. Hiermee verzekeren we de integriteit van de database. Alle gegevens zijn door de koppeling nog steeds te bereiken.



Geef aan het entiteitstype een betekenisvolle naam

We bedenken voor deze repeterende groep een goede naam. Omdat het de gegevens bevat die over de te volgen cursussen gaan zullen we deze entiteit de naam Opleiding geven.

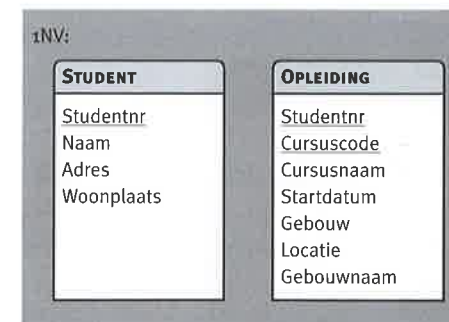


Maak van de sleutel een primaire sleutel

De overgenomen sleutel uit de eerste entiteit is niet voldoende om als primaire sleutel te dienen. Een student kan immers meerdere cursussen volgen. In dat geval zal bij elke cursus hetzelfde studentnummer staan en is er geen sprake meer van een unieke sleutel.

De primaire sleutel zal een samengestelde sleutel moeten zijn bestaande uit meerdere attribuuttypen. We kiezen het attribuuttype Cursuscode als tweede sleutelattribuut. Deze geeft samen met het studentnummer unieke attribuutwaarden. We onderstrepen deze dan ook.

De eerste normaalvorm ziet er dan als volgt uit:



Tweede normaalvorm (2^{de} NV)

De te nemen stappen:

- Bepaal het attribuuttype dat maar voor een deel van de sleutel afhangt
- Splits het attribuuttype af in een eigen entiteitstype met sleutel
- Geef het entiteitstype een betekenisvolle naam

Bepaal het attribuuttype dat maar voor een deel van de sleutel afhangen

Binnen de eerste normaalvorm kunnen attribuuttypen zitten die maar aan een deel van de (primaire) sleutel gekoppeld zijn. Deze groepen zijn dus niet uniek bepaald door de gekozen combinatie van sleutelattributen. Zo is het attribuuttype Cursusnaam niet afhankelijk van Studentnr maar wel weer van het sleutelattribuut Cursuscode. Het attribuuttype Cursuscode en Cursusnaam vormen hiermee een apart entiteittype.

Splits de attributen af in een eigen entiteit

De attribuuttypen Cursuscode en Cursusnaam vormen een eigen entiteittype. Het sleutelattribuut Cursuscode die in de oorspronkelijke entiteit geen primaire sleutel was is dat in de nieuwe wel. Om de koppeling met het oorspronkelijke entiteittype te behouden laten we een kopie van het sleutelattribuuttype Cursuscode achter.

STUDENT	OPLEIDING	
<u>Studentnr</u>	<u>Studentnr</u>	<u>Cursuscode</u>
Naam	<u>Cursuscode</u>	Cursusnaam
Adres	Startdatum	
Woonplaats	Gebouw	
	Locatie	
	Gebouwnaam	

Geef het entiteittype een betekenisvolle naam

We geven deze nieuwe entiteit ook de betekenisvolle naam Cursus. De tweede normaalvorm ziet er dan als volgt uit:

2NV:

STUDENT	OPLEIDING	CURSUS
<u>Studentnr</u>	<u>Studentnr</u>	<u>Cursuscode</u>
Naam	<u>Cursuscode</u>	Cursusnaam
Adres	Startdatum	
Woonplaats	Gebouw	
	Locatie	
	Gebouwnaam	

Derde normaalvorm (3NV)

De te nemen stappen zijn:

- Bepaal de attribuuttypen die niet afhankelijk zijn van een sleutelattribuuttype
- Splits het attribuuttype af in een eigen entiteittype
- Bepaal het sleutelattribuut
- Geef aan het entiteittype een betekenisvolle naam
- Geef de externe sleutel aan

Bepaal de attribuuttypen die niet afhankelijk zijn van een sleutelattribuuttype

Er kunnen attribuuttypen bestaan die niet afhankelijk zijn van een sleutelattribuuttype maar van andere gewone attribuuttypen. Zo is in ons voorbeeld het attribuuttype Gebouw, Gebouwnaam en Locatie niet afhankelijk van Studentnr of Cursuscode. Deze vormen daarmee een afzonderlijke entiteittype.

STUDENT	OPLEIDING	CURSUS
<u>Studentnr</u>	<u>Studentnr</u>	<u>Cursuscode</u>
Naam	<u>Cursuscode</u>	Cursusnaam
Adres	Startdatum	
Woonplaats	Gebouw	
	Locatie	
	Gebouwnaam	

Splits de attribuuttypen af in een eigen entiteittype

Deze onafhankelijke attributen worden afgesplitst in een eigen entiteit.

STUDENT	OPLEIDING	CURSUS	
<u>Studentnr</u>	<u>Studentnr</u>	<u>Cursuscode</u>	Gebouw
Naam	<u>Cursuscode</u>	Cursusnaam	Locatie
Adres	Startdatum		Gebouwnaam
Woonplaats			

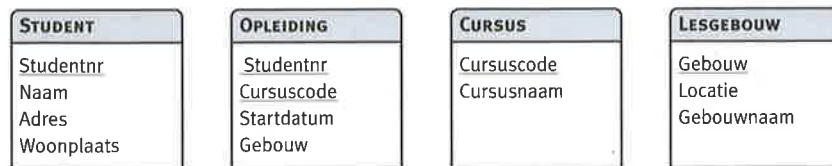
Bepaal het sleutelattribuuttype

Als we naar de attributen en hun attribuutwaarden kijken, dan is het attribuut gebouw uniek. Deze kunnen we gebruiken als primaire sleutel in de nieuwe entiteit.

GEBOUW	LOCATIE	GEBOUWNAAM
2	Utrecht	Paviljoen
1	Utrecht	Toren
7	Enschede	Bastille
4	Zwolle	Tuinschuur

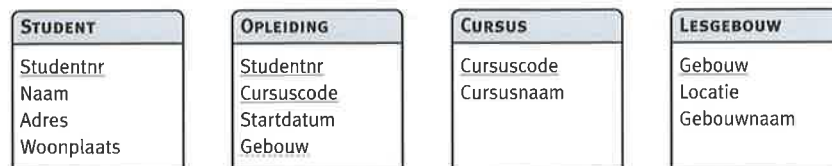
Geef aan het entiteittype een betekenisvolle naam

Ook nu moeten we een betekenisvolle naam bedenken voor het nieuwe entiteittype. Omdat hier bijgehouden wordt in welk gebouw de cursus plaatsvindt geven we het entiteittype de naam Lesgebouw.

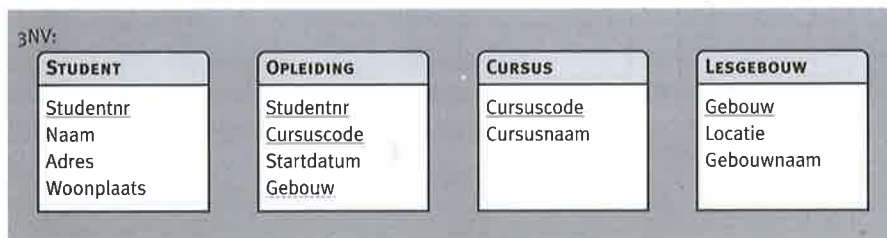


Geef de externe sleutel aan

De koppeling tussen het oorspronkelijke entiteitstype en de nieuwe loopt via het attribuuttype Gebouw. Maar het attribuuttype Gebouw is in deze entiteit geen sleutelattribuut. Om de koppeling aan te geven maken we van Gebouw een externe sleutel. Deze geven we aan door het attribuuttype te onderstrepen met een stippellijn.

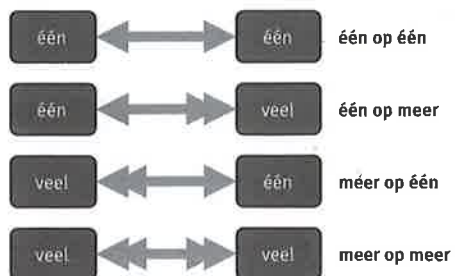


De derde normaalvorm ziet er dan als volgt uit:



Diagrammen

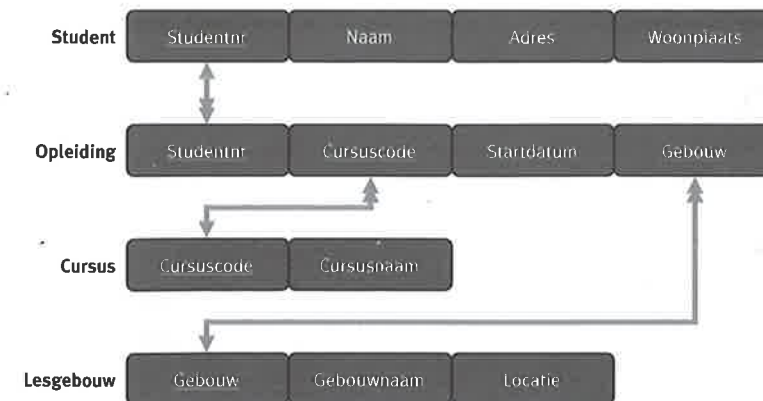
In een Strokendiagram en een Bachman-diagram beschrijf je de relaties tussen de entiteitstypen op een grafische manier. We kennen de volgende relaties:



De relaties één-op-veel en veel-op-één zijn spiegelbeelden van elkaar.

Strokendiagram

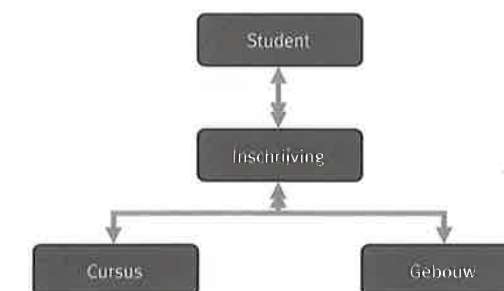
In een strokendiagram worden de verschillende relaties met pijlen grafisch aangegeven:



- Eén student kan vele cursussen volgen.
- Elke cursus heeft maar één cursuscode, maar er kunnen vele cursussen zijn.
- Op een cursus kunnen meerdere studenten inschrijven.
- Er zijn vele gebouwen waarin kan worden lesgegeven, maar slechts één gebouw waar je een bepaalde cursus kunt volgen.

Het Bachman-diagram of Entity-relationship diagram (ERD)

In een Bachman-diagram worden de relaties tussen de entiteitstypen weergegeven. Van het voorbeeld van de normalisatie van het cursusformulier kun je het volgende Bachman-diagram maken:



- Eén student kan vele cursussen volgen.
- Elke cursus heeft maar één cursuscode, maar er kunnen vele cursussen zijn.
- Op een cursus kunnen meerdere studenten inschrijven.
- Er zijn vele gebouwen waarin kan worden lesgegeven, maar slechts één gebouw waar je een bepaalde cursus kunt volgen.