Onderbouwing Normalisatie Database

## Omgezette recursieve relatie

In ons Relationeel Database Ontwerp staat een extra tabel ‘RelatedCourses’. Deze tabel is de verwerking van de recursieve relatie tussen courses. Een Course heeft namelijk meerde ‘Suggested Courses’, deze worden gekoppeld door middel van de koppel tabel ‘RelatedCourses’.

## Omgezette zwakke entiteit

De zwakke entiteit register is omgezet in een tabel register, met hierin een eigen unieke ID en de waarden voor StudentEmail, CourseName en CertificateID. Wij hebben deze ID meegegeven om te voorkomen dat we in alle gerelateerde tabellen ook de andere kolommen van de meervoudige Primary Key moesten zetten. Wanneer we probeerden om een PK\_Constraint (Primary Key Constraint) aan te maken voor meerder kolommen moesten we deze extra kolommen ook in de gerelateerde tabellen zetten omdat ander niet de hele PK (Primary Key) beschikbaar was vanuit die andere tabellen. Nu kunnen we makkelijk een nieuwe RegisterID aanroepen en krijgen we van die registratie de benodigde informatie.

## Omgezet attribuut

Het attribuut ‘Progress’ voor een student die een content item bekijkt is verwerkt in de tabel ‘Viewing’, deze tabel heeft als primary key een student en een ContentID op deze manier kan voor een student voor een specifiek content item progressie bijgehouden worden. In ons systeem kan een Student zich maar één keer inschrijven voor dezelfde cursus zoals beschreven in de opdracht. Dit komt ook overeen met deze tabel omdat je maar één keer de progressie bij kunt houden voor een combinatie van een student en content item.

## ON Update en ON Delete constraints

In onze database hebben we op de juiste manieren de eisen voor de referentiële integriteit verwerkt.

* De email van student is een heeft verschillende Foreign Keys (Viewing.StudentEmail, Register.StudentEmail) en wanneer deze aangepast wordt worden ook de Foreign Keys aangepast, hetzelfde geldt voor het verwijderen.
* Een cursus kan alleen verwijderen wanneer deze geen inschrijvingen heeft. Dit is verwerkt in het programma zelf. Er wordt in de validatie methode gekeken of een cursus inschrijvingen heeft wanneer dit het geval is kan de cursus niet verwijderd worden.

## Onderbouwing 3e normaalvorm

In onze database wordt de 3e normaalvorm compleet toegepast. Dit betekent dat er binnen een tabel geen kolommen zijn die maar afhankelijk zijn van een deel van de primary key voor die tabel.

In de tabel Register hebben we een ID gebruikt. In deze tabel hangen de StudentEmail en CourseName af van de RegisterID en aan de hand van een RegisterID wordt alle informatie opgehaald.

De tabellen webcast en module zijn gesplitst van content item om te voorkomen dat het type content item opgeslagen moet worden (Webcast of Module).

# 