

# Codecademy Statistics

Relationele databases 2 & Programmeren 2



## Statistics

### **Groep 15 (PAP7)**

Joey van kuijk | 2136862 | [cam.vankuijk@student.avans.nl](mailto:cam.vankuijk@student.avans.nl)

Joy Boellaard | 2354217 | [je.boellaard@student.avans.nl](mailto:je.boellaard@student.avans.nl)

Tim van Vliet | 2155294 | [t.vanvliet4@student.avans.nl](mailto:t.vanvliet4@student.avans.nl)

Lisa Tyem | 2116056 | [lj.tyem@student.avans.nl](mailto:lj.tyem@student.avans.nl)

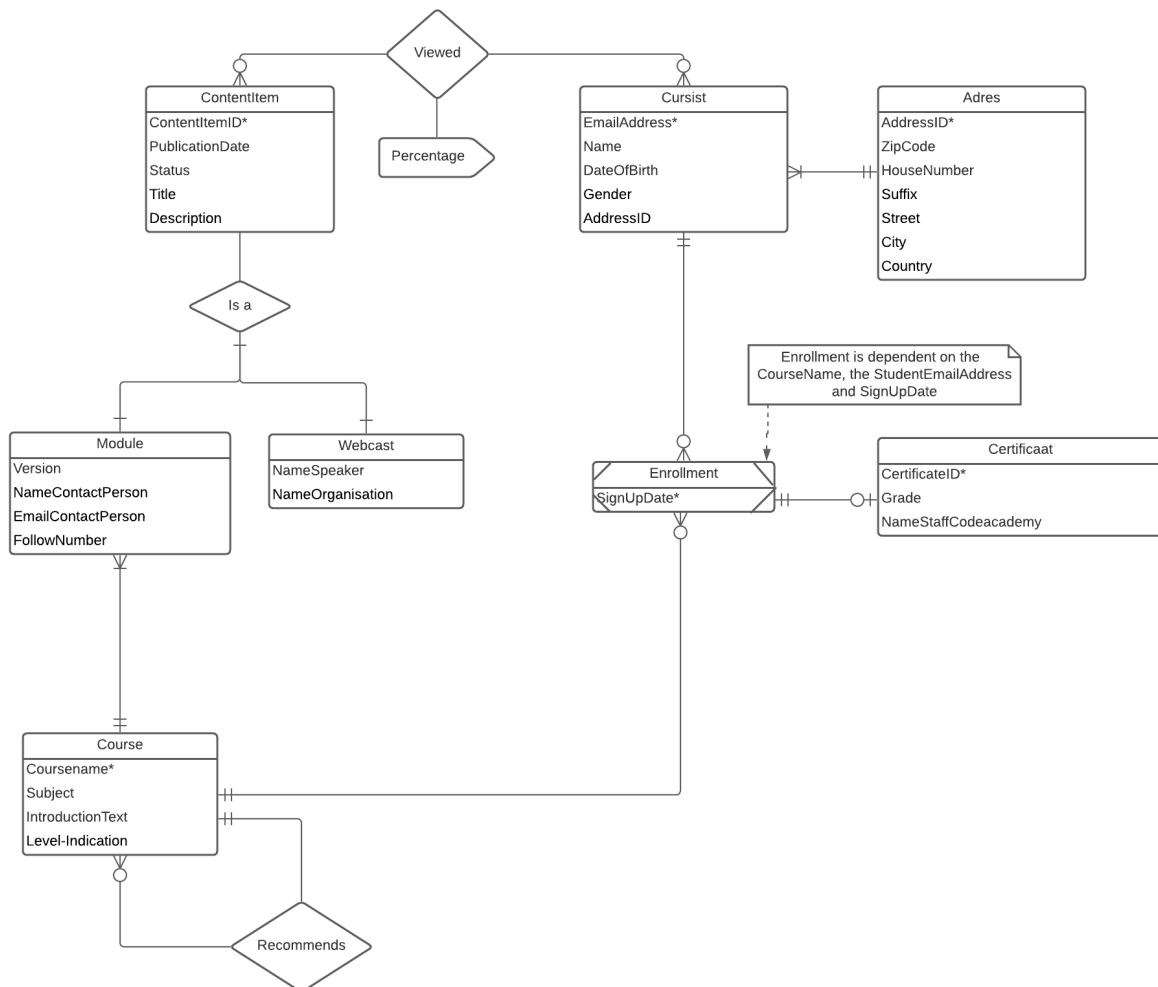
# Inhoudsopgave

## **Inhoudsopgave**

<b>Database ontwerp</b>	<b>1</b>
1.1 ERD	1
1.2 RDO	2
1.3 De 3e normaalvorm	2
<b>Class diagram</b>	<b>3</b>
<b>Handleiding voor VSCode</b>	<b>4</b>
3.1 Benodigdheden	4
3.2 Installatie Java sdk's	5
3.3 Installatie database	6
3.4 Opstarten applicatie	8
<b>Reflectie</b>	<b>9</b>
4.1 Joey's reflectie	9
4.2 Tim's reflectie	9
4.3 Joy's reflectie	10
4.4 Lisa's reflectie	10

# 1. Database ontwerp

## 1.1 ERD

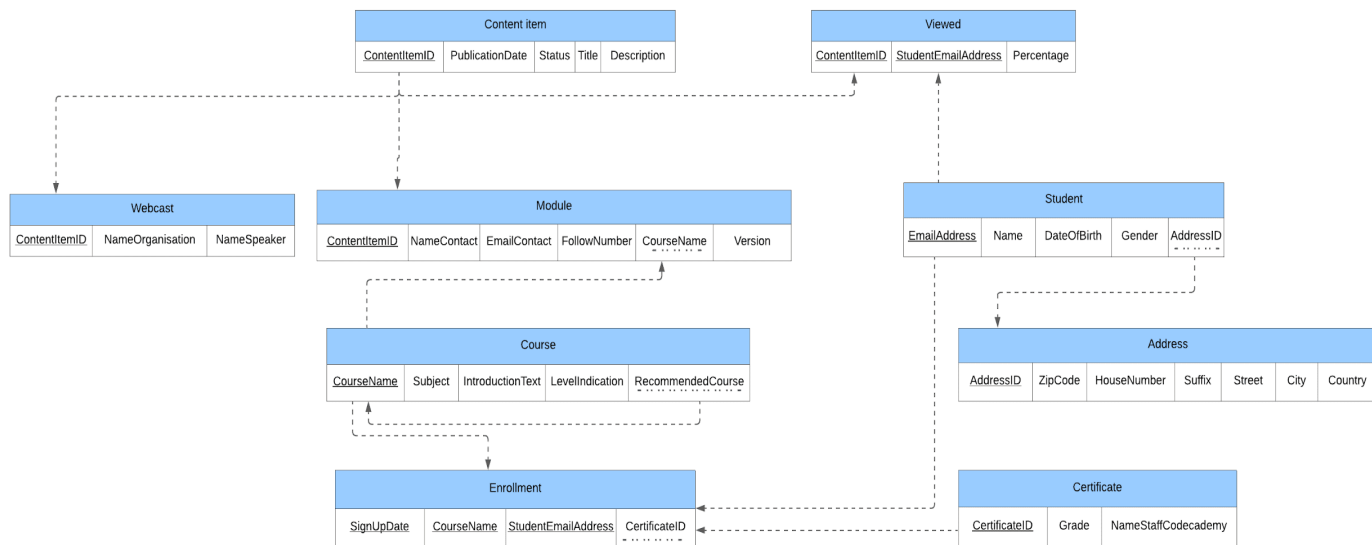


Een cursist kan meerdere content items bekijken en die kunnen ook weer bekeken worden door meerdere cursisten. Bekeken vormt samen met het intersectie gegeven percentage een aparte tabel in de onze database, zodat we bij kunnen houden hoeveel van het content item bekeken is, en door wie.

Content item bestaat uit een module en een webcast, waarbij de modules weer onderdeel uitmaken van de cursussen. Content item is dus een supertype, module en webcast zijn subtypes. Een module is onderdeel van maar één content item, en behoort ook tot maar één cursus. Cursus kan echter wel meerdere modules bevatten.

Een cursist kan zich inschrijven voor een cursus, wat maakt dat inschrijving afhankelijk is cursist, en dus een zwakke entiteit is. Als een cursus is behaald wordt een certificaat toegekend, deze behoort maar tot één inschrijving.

## 1.2 RDO

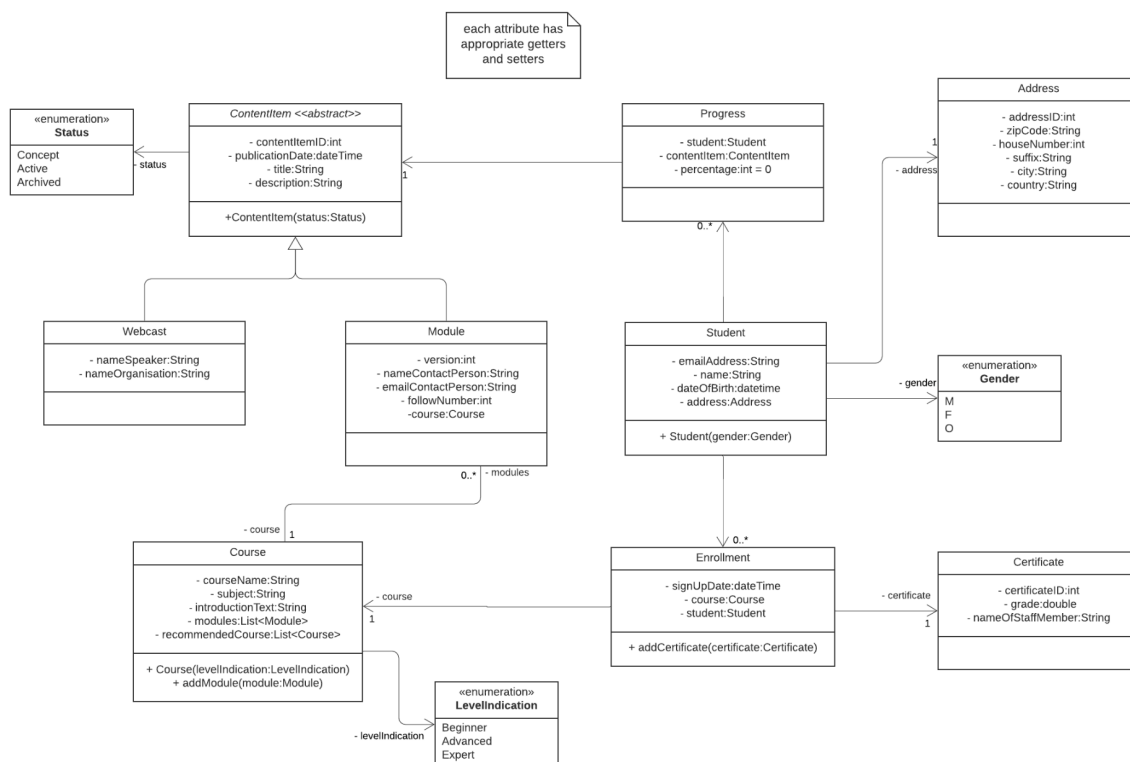


Een belangrijke verandering in het database script vergeleken met het RDO is dat er een nieuwe entiteit is gemaakt voor RecommendedCourses. In het RDO hierboven zou het niet mogelijk zijn om meerdere RecommendedCourses op te slaan bij een Course. Ook is Viewed opgesplitst in twee entiteiten, zie onderbouwing bij het hoofdstuk Class diagram. Als laatste hebben we de Title en Description uit ContentItem gehaald. Hoewel beide Webcast en Module een Title hebben, was het bij Module van belang dat de Title samen met de Version uniek was.

## 1.3 De 3e normaalvorm

Zoals in het RDO te zien is, zijn er geen transitieve afhankelijkheden meer. Oftewel: kolommen die afhankelijk zijn van een niet-sleutel. Zie hieronder de voorbeelden die hier dus juist niet aanwezig zijn. Ook zijn er geen functionele afhankelijkheden meer: kolommen die afhankelijk waren van een deel van de primaire sleutel zijn afgesplitst. Dit zijn de voorwaarden voor de derde normaalvorm, en betekent dus dat het RDO hieraan voldoet.

## 2. Class diagram



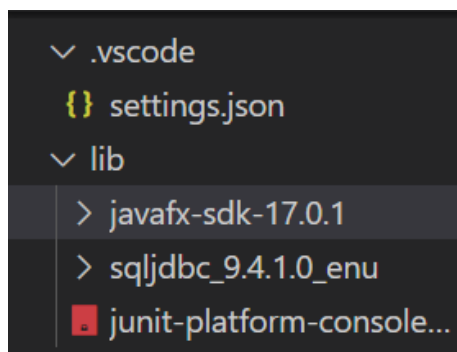
Dit is de domain class diagram, gebaseerd op het database ontwerp van het vorige hoofdstuk. Een ContentItem is een abstracte klasse omdat deze alleen als Module of als Webcast aangemaakt mag worden. Certificaat is verbonden aan Enrollment omdat een Student meerdere keren kan inschrijven voor een Course en dus ook meerdere certificaten zou mogen behalen. In dit model is het nog mogelijk om een Course aan te maken zonder modules, maar dit hebben we later in de code aangepast omdat een Course minimaal één module moet hebben. Verder hebben we ook in de code (en het database script) de Progress van een Webcast en de Progress van een Module uit elkaar gehaald, omdat eenzelfde module meerdere Progress entities kan hebben van dezelfde student bij verschillende inschrijvingen, maar voor een Webcast kan niet worden ingeschreven.

## 3. Handleiding voor VSCode

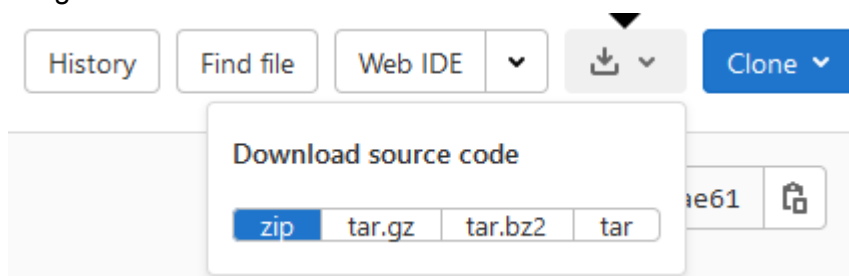
### 3.1 Benodigdheden

De bestanden benodigd om de applicatie te runnen zijn te vinden in de github repository; <https://github.com/2182556/Codecademy-15>. Ze zullen ook zijn toegevoegd in het zip-bestand waar dit document in zit. Het bestaat uit een java project, met daarbij de jar files en het script nodig om de applicatie te kunnen runnen.

Om het database script te runnen hebben wij gebruik gemaakt van de Microsoft SQL Server Management Studio en is dus nodig voor de installatie van de database. Verder hebben we gebruik gemaakt van Visual Studio Code met daarbij de Java en JavaFx devkits. JavaFx en het database script zijn beiden toegevoegd aan de library, dus het downloaden van JavaFx is hierdoor niet nodig.



De bestanden zijn dus te vinden op onze git repository. Download ze door op de download knop te klikken en daar te kiezen voor zip. Extract de file in een map waar alles makkelijk terug kunt vinden.

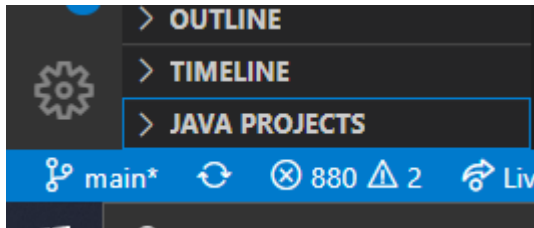


<https://github.com/2182556/Codecademy-15>

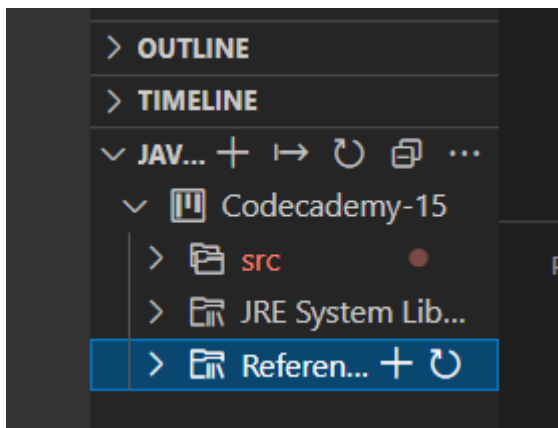
## 3.2 Installatie Java sdk's

Mochten de jar files toch niet blijken te werken, dan zullen ze handmatig toegevoegd moeten worden aan de referenced libraries.

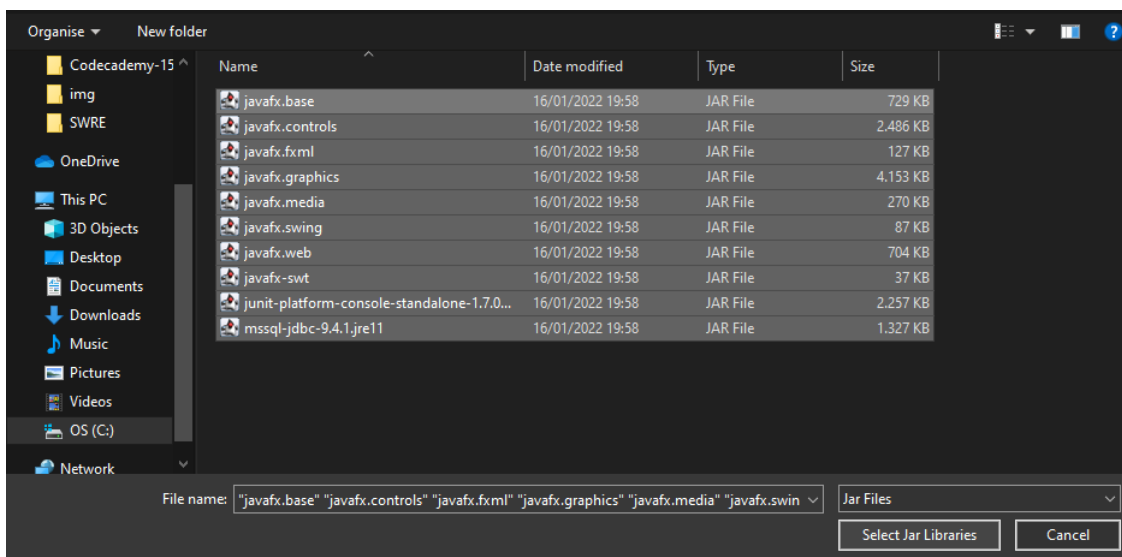
Ga onderaan de explorer naar 'java projects'.



Klik vervolgens bij de optie 'referenced libraries' op het plusje.



Navigeer naar waar de jar files zijn opgeslagen, en selecteer deze. Druk daarna op 'select jar libraries'.

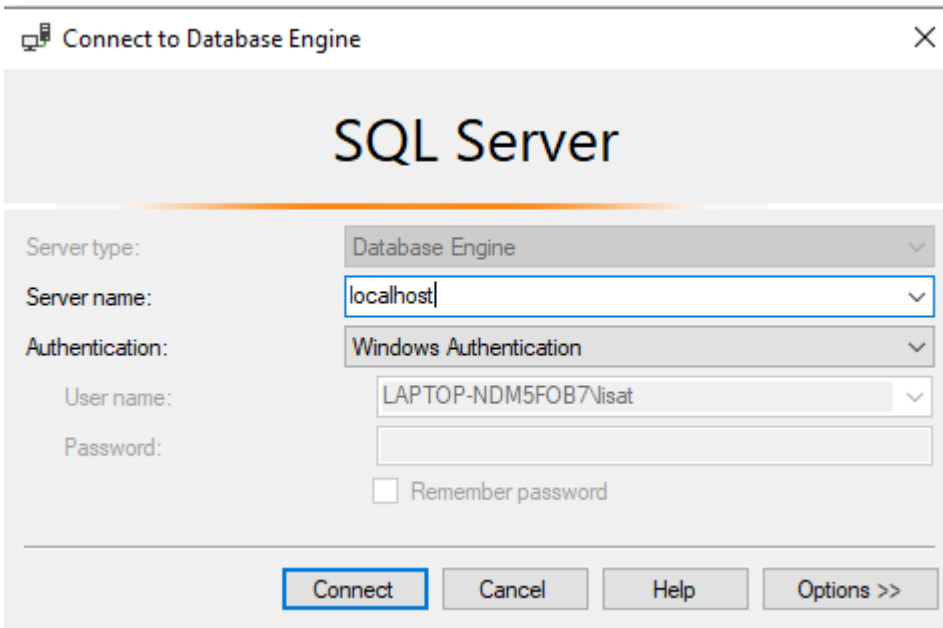


### 3.3 Installatie database

.git	16/01/2022 21:47	File folder	
.idea	16/01/2022 21:39	File folder	
.vs	16/01/2022 21:39	File folder	
.vscode	16/01/2022 21:34	File folder	
bin	16/01/2022 21:34	File folder	
lib	16/01/2022 21:39	File folder	
src	16/01/2022 21:34	File folder	
Class diagram Group 15.png	17/12/2021 12:07	PNG File	193 KB
CodecademyDBQuery.sql	16/01/2022 21:05	Microsoft SQL Server...	10 KB
DomainClass diagram Group 15.png	17/12/2021 12:07	PNG File	435 KB
ERD group 15.png	17/12/2021 12:07	PNG File	65 KB
RDO group 15.png	17/12/2021 12:07	PNG File	50 KB
README.md	21/11/2021 12:28	MD File	1 KB

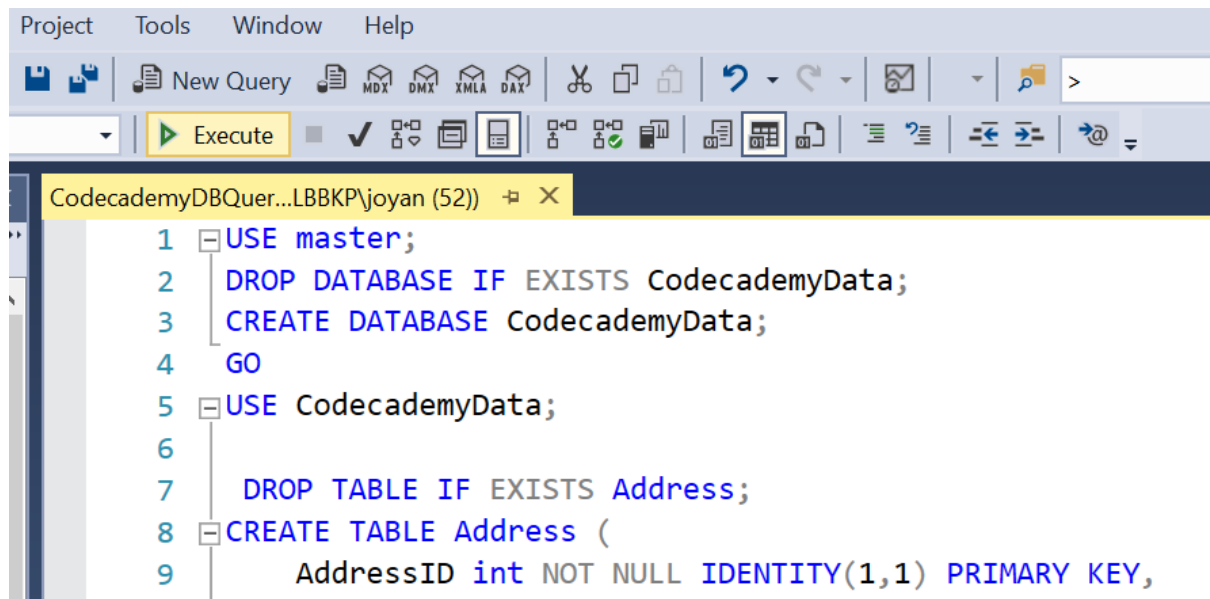
Open het database script vanuit de file waar je de code in hebt geplaatst.

Zorg dat bij het maken van de connectie je op de localhost zit. Selecteer daarbij bij authenticatie 'windows authentication'. Klik vervolgens twee keer achter elkaar op connect.



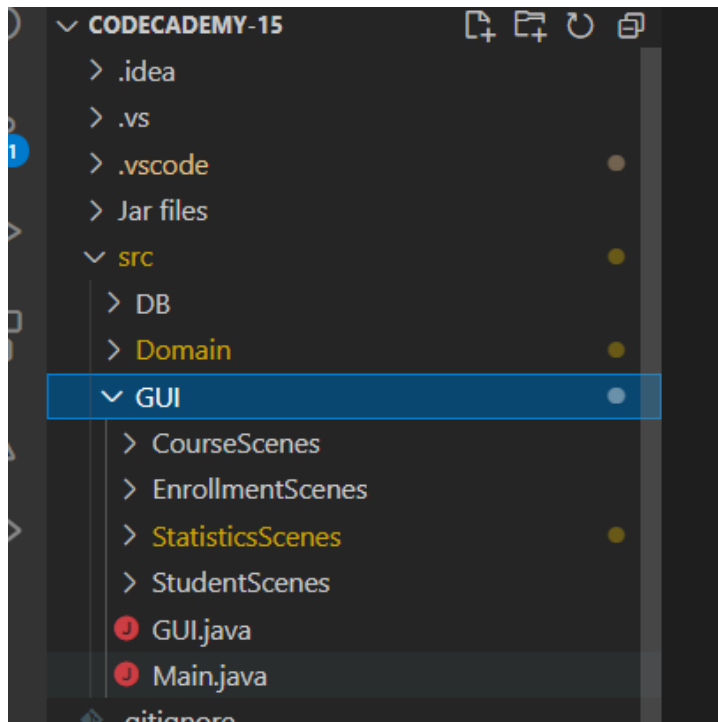
Dan is het alleen nog nodig om het script te executen, en is de database geïnstalleerd.



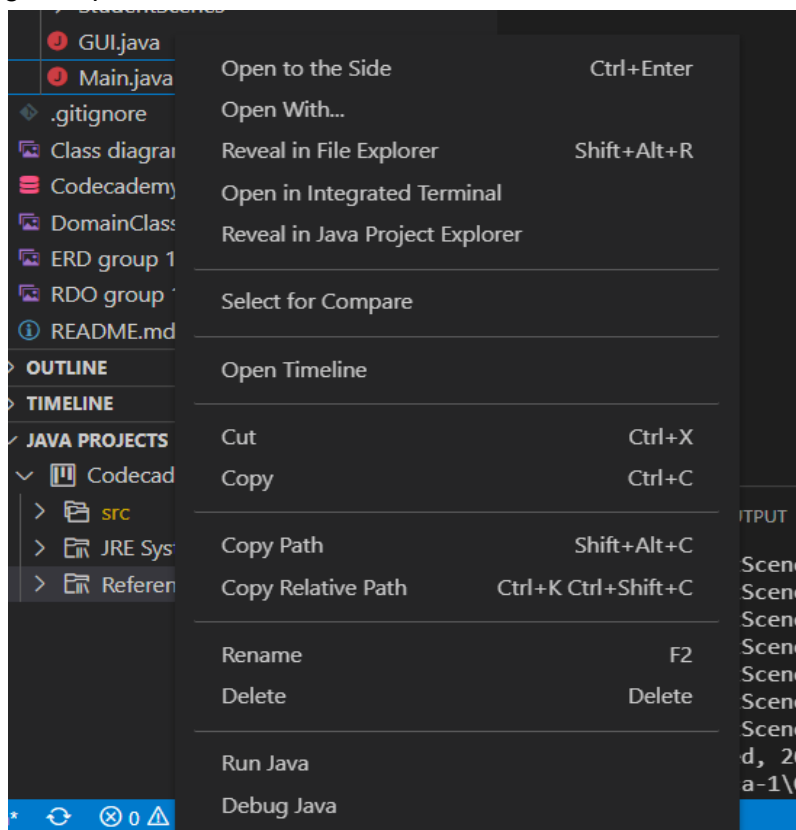


### 3.4 Opstarten applicatie

Om de applicatie te starten navigeer je naar `src` → `GUI` → `Main.java`.



Klik met de rechtermuisknop op `Main.java`, en kies dan voor 'run java'. Als alle code is gecompileerd dan zal de GUI vanzelf in beeld komen.



## 4. Reflectie

### 4.1 Joey's reflectie

Ik vond het een zeer lastig en uitdagend project, maar op een goede manier. Dit project laat mooi zien hoe de verschillende vakken waar wij les in krijgen samenwerken binnen de informatica. Het project ging niet altijd vanzelf, we liepen vaak tegen problemen aan waar we pas na veel overleg met zijn vieren uit kwamen. Persoonlijk vind ik dat wij een goed werkende applicatie op hebben kunnen leveren, ook al was het vrij dichtbij de uiteindelijke deadline.

De planning was aanvankelijk erg goed, zeker aangezien elke milestone ruim op tijd klaar was en wij er ook zeer nuttige feedback op hebben gehad, alleen tegen het einde van het project liep de planning een beetje in de soep, met name omdat wij ons hadden verkeken op hoe veel er moest gebeuren voor deze applicatie. Maar na een paar dagen hard werken was dit tijdsverlies weer goed getrokken.

De samenwerking binnen de groep was prima, iedereen was er vrijwel elke bespreking bij, met uitzondering van hier en daar wat privé zaken die voor gingen. Ik heb hier zelf ook een belangrijke les geleerd over hoe dat iedereen zijn of haar eigen sterke punten heeft, en hoe dan samen werken aan een groepsproject betekent dat je elkaars sterke en zwakke punten aan kan vullen. Het was een erg lastig project maar ik vind dat wij het goed aangepakt hebben.

### 4.2 Tim's reflectie

Ik vond het een uitdagender project dan ik aanvankelijk had verwacht, niet omdat mijn verwachting van de moeilijkheidsgraad niet correct was maar meer over het alleen bijhouden van de milestone's niet genoeg was. In de vakantie is iedereen natuurlijk ook bezig met andere dingen gecombineerd met dat je elkaar niet meer op school ziet en het contact dus gelijk minder is wat resulteerde in een veel hogere werkdruk dichtbij de deadline dat je eigenlijk wilt hebben. Met alle milestone's als deadline ging het samenwerken super goed, we deden dingen samen die niet opgesplitst konden worden en verdeelden het werk wat we wel konden splitsen. Mijn aandeel van het verrichte werk in de vakantie had meer moeten zijn. Het is voornamelijk door Joy dat er toen nog vooruitgang geboekt was.

Volgende keer wil ik dit natuurlijk voorkomen, en dat ga ik doen door meer initiatief te blijven steken in het contact wanneer er is iets als vrije dagen. Het is logisch dat je wat contact en tijd verliest maar iemand die energie blijft steken in overleg blijven houden lost dit snel op. Verder vind ik dat de samenwerking in onze groep goed verlopen is, buiten de vakantietijd is er in mijn mening door iedereen evenveel moeite in het project gestopt en ook al was de werkdruk hoog op het eind kwam iedereen bij elkaar om het alsnog af te hebben.

Qua samenwerking heb ik vooral geleerd dat contact blijven houden het allerbelangrijkst is en dat ik hierin meer initiatief mag nemen, maar voornamelijk zou ik zeggen dat ik geleerd heb hoe het is om samen te werken aan een codeer opdracht. Werken in groepsverband

was me al wel bekend maar dit is de eerste keer dat ik samen met andere heb gewerkt aan een opdracht dat zowel java als sql koppelt. En dat vond ik heel interessant. Het laat namelijk voor mij echt zien hoe werken via een github is en hoe het zou kunnen zijn als ik dit voor werk zou doen en dat vond ik een leuk onderdeel van dit project.

### 4.3 Joy's reflectie

Voor de vakantie ging het samenwerken erg goed, we hielden alle milestones bij en leken aardig op schema te zitten. We namen allemaal initiatief wanneer dit nodig was voor een milestone en daardoor leek alles goed te gaan. Echter pas aan het eind van de vakantie kwam ik erachter hoeveel werk het nog was om de applicatie volledig werkend te krijgen. Tijdens de vakantie zijn we het contact een beetje verloren en kwamen we er eigenlijk te laat achter hoeveel er nog moest gebeuren een week voor de deadline. Ik heb dus de laatste week voor de inleverdatum ontzettend veel tijd besteed aan het zorgen dat de applicatie aan de minimale vereisten voldeed. Ik vind het zelf erg jammer dat ik er pas zo laat hard aan ben gaan werken, omdat er op veel vlakken duidelijk is dat de applicatie wat is afgeraffeld. Toch denk ik dat het eindproduct niet slecht is, vergeleken met de tijd die uiteindelijk nog over was.

De reden dat het met de planning fout was gegaan, was omdat de vertical slice voor milestone 5 ons erg goed afging, en dus kregen we het idee dat de rest van de applicatie niet ontzettend veel moeite zou kosten. Voor een volgende keer heb ik zeker geleerd om een beter beeld te krijgen van wat er nog moet gebeuren in een vroeger stadium van de opdracht, zodat je later niet tegen verrassingen aanloopt. Ook is het belangrijk om met elkaar in contact te blijven, zodat het gemakkelijker is om dingen ook weer op te pakken na een vakantie en kan het behulpzaam zijn om voor de taken die worden verdeeld deadlines te stellen, zodat er wat meer druk op zit om het op een tijdig moment af te krijgen.

### 4.4 Lisa's reflectie

De milestones hielpen ontzettend goed bij de samenwerking. We hadden elke week één duidelijk doel om naartoe te werken, en dit verliep goed. Echter rond de tijd van de vakantie waren de milestones een greep van wat er verder allemaal nog moest gebeuren, en begonnen we achter te lopen. In de vakantie is er nagenoeg niks gedaan, qua communicatie in ieder geval, en zijn we nog verder achter gaan lopen. Op de laatste dag, de dag van de deadline zelf, zijn we voor het eerst sinds lange tijd weer met zijn allen bij elkaar gekomen, en hebben we gelukkig de achterstand in kunnen halen. Maar dit was niet gelukt als Joy op de achtergrond niet ontzettend hard door bleef werken.

Ik doe dit project nu voor de tweede keer, en net als de eerste keer is mijn gebrek aan contributie deels te wijten aan het niet goed geleerd te hebben. Door dingen niet of niet goed te begrijpen, laat je ze makkelijk links liggen, of duurt iets wat een kwartier had kunnen duren 2 uur. Om samen te kunnen werken is het belangrijk dat je je eerst je eigen zaken (kennis) op orde hebt, anders valt er lastig bij te dragen aan een groter geheel.

Wat ik geleerd heb is dat communicatie altijd de grootste valkuil blijft. Duidelijke afspraken, zelf deadlines inplannen, korte meetings houden en elkaar controleren op het werk. Dat in combinatie met tijdig beginnen en taken verdelen in hapklare stukjes, had ervoor kunnen

zorgen dat we allemaal evenredig hadden kunnen bijdragen, en niet op het laatste moment alles op alles hadden moeten zetten. En specifiek voor mijzelf dat ik ook aandacht moet blijven besteden aan het huiswerk.