

## Documentation Milestone 1

### Followed optimization processes

First off, we started with thinking about how to translate the queries, and tried to answer the question: 'What do these queries need to run optimally'. With this, we took the material discussed in the lectures into consideration. We used views that will be created before the queries will run, to make it easier for the queries to acquire the correct data from the tables. We tried to minimize the amount of views that we had to create. We did this for multiple reasons namely: To save space. When one would create views for every single query, and the number of queries increase, or the amount of data will increase, the amount of space required will increase. When dealing with huge amounts of data, space is very costly. Another reason is to make the views more dynamic. Using views when creating other views, making the views in such a way such that they can be used for multiple queries. This will also reduce the amount of space we have to use for the views.

### Chosen optimizations

We have created some materialized views to optimize the queries:

**GPA:** This view contains the GPA of each studentregistration ID, which can be linked back to the StudentRegistrationToDegrees table to find the corresponding student ID's. The view contains 2 columns of integers with the number of rows equal to the amount of studentregistration ID's.

**HighestGradeCourseOffers:** This view shows the student ID and the corresponding grade of that student. One more property of this grade is that it is the highest grade that has been achieved for the course by all students who followed to it. The view contains 4 columns with the number of rows equal to the amount or courses there are

**StudentECTS:** This view shows for each studentregistration ID how many credits the student has already obtained. This view contains 4 columns all containing integers. The number of rows are equal to the total amount of StudentRegistration ID's.

**ActiveStudentsPerDegree:** This view shows the student id, the StudentRegistration ID and the Degree ID of the students who did not yet obtain the required number of degrees. The space required are 3 columns, and number of rows equal to the total amount of StudentRegistration ID's.

**DegreeCompleted:** This view contains the studentregistration ID's of those who have already completed their degree. Meaning they obtained at least the required number of ECTS for that degree. This view extends the StudentsECTS view where it checks whether the number of ECTS of that student is greater or equal to the required number of ECTS. This view is only a subset of the StudentECTS view and consists of 2 columns of integers.

**NumAssistentCourseOffer:** This view contains 2 columns of integers. One for the courseID and one for the count of the studentregistrationID where that count is bigger than 0

**NumStudentsCourseOffer:** This view offers a count of student registered for each course. Consisting of 2 columns with the length of the number of courses there are in total. Both columns contain an integer.