2ID70 - Milestone 1 - Documentation - Group 23

Optimization process

1. Create primary keys conform an unique index themselves. These primary keys are added in the tables Degrees, Students, StudentRegistrationsToDegrees, Courses and CourseOferrs on DegreeId, StudentId, StudentRegistrationId, CourseId and CourseOfferId respectively.
2. Index on CourseOfferId from CourseRegistrations.
3. Index on CourseOfferId from StudentAssistants table.
4. Index on StudentRegistrationId from CourseRegistrations table.
5. Create a materialized view PassedCreditsPerRegistration that stores all the passed credits for every StudentRegistrationId.
6. Create an index on StudentRegistrationId on the previous materialized view.
7. Create an auxiliary view CompletedRegistrations.
8. Create a materialized view ActiveRegistrations.
9. Create an index on StudentRegistrationId from the materialized view in point 8.
10. Create a materialized view MaxGradesOfferAndStudent that stores the maximum grades obtained by course offer and the StudentId of who obtained that max grade.
11. Create an index on StudentId from the materialized view in 10.
12. Create an auxiliary view studentsCourseOfExcellence that references all the number of courses where every student was excellent.
13. Create an auxiliary view aux1 that computes the number of registrations by CourseOfferId.
14. Create an auxiliary view aux2 that computes the number of student assistants by CourseOfferId.
15. Create an auxiliary view CourseOffersRegsAss that computes the triples (CourseOfferId, numberOfRegistrations, numberOfStudentAssistants).

Performance:

1. The total size of the optimizations is 4440MB and it takes less than 6 minutes in our machines to create all the structures.
2. The primary keys improved the performance when searching for a record by its id in tables mentioned in 1 of the previous section. E.g. searching an student in table Students went from 15 seconds to less than one second with the primary key.
3. The index created in 4 in the previous section improves extremely the performance of query 1, going from 15 seconds to less than 3 in the worst of the cases.
4. The structures from 5 and 6 in the previous section are needed for query 2 and 7 and they indeed improve the performance.
5. The structures created in 7, 8 and 9 from the previous section improve the performance and the readability of queries 3 and 6.
6. The structures created in 8, 9 and 10 are helpers for query 6. Apart from improving its performance, they make the query more readable.
7. For query 8, the indices in 2 and 3 (used in views 11, 12 and 13) in previous section improve its performance especially for the group by clauses.