Optimization process:

First, we … then, we ...

Chosen optimizations:

We have chosen to optimize our dataset by using 4 materialized views and 1 index. We use views in all our queries, these do not result in any performance optimization, but are useful to simplify our queries.

The materialized view *with_grade* is used for query 1 and 5 and for making other (materialized) views that are connected to the grades. The main benefit of this materialized view is that often grades that are 0 or null should not be taken into account in the given queries, so a dataset that excludes those values is considerably faster. The total space for this materialized view is 3209 MB. Constructing this materialized view costs 1 minute and 32 seconds. Our performance gain is 15 seconds per query run, tested on query 5. Without materialized view this query takes 1 minute, with *with_grade* it takes 45 seconds.

The materialized view *active_students_per_degree* is used for queries 3 and 7. Since we needed the active students per degree multiple times in the queries, and making the view was quite costly to perform this per query, 1 minute and 1 second, a materialized view seemed to be the best option. The size of this view is 204 MB. With this view, the running time of query 3 is 6,5 seconds, as opposed to the 1 minute and 32 seconds when not using the view. This is a performance gain of 1 minute and 25 seconds.

The materialized view *GPA* has a size 123 MB, and consists of the gpa of each student per degree. *GPA* is also needed in two queries, 2 and 7, and was executed in 59 seconds. Without this view, our performance would have been 41 seconds and with this view the query takes 600 milliseconds. This is a performance optimization of 40,4 seconds. Like *active_student_per_degree*, the GPA was often needed and calculation per query was not beneficial due to query 2 running ten times.

Our last materialized view *not_passed* requires 100 MB of space, and is used in query 2. Due to query 2 running ten times, a materialized view had a huge benefit. The required time to create this view was 58 seconds. Query two with this materialized view and its index runs in 600 milliseconds. Without the materialized view, the resulting time is 50 seconds. This results in a performance optimization of 49,4 seconds.

We have made one index on our materialized views. We have only used one index, since often the index did not guarantee performance optimization. The index *idx_not_passed* is an index on the materialized view *not_passed*, which can be constructed in 2 seconds and requires 62 MB of space.

The total gain in our workload is 189,8 seconds, which is a little bit more than three minutes. The total space used for the index and materialized views is 3698 MB. The total time required to construct the materialized views and index is 272 seconds, which results in just under five minutes. These values are computed by adding the values of all the materialized views and the index that we created.

Materialized views:

- with_grade with size of 3209 MB and time 1 min 32 sec
  - query 5 with with_grade with index on grade: 55s
  - query 5 with with_grade without index on grade: 45s
  - query 5 without with_grade: 1 min
- active_students_per_degree of size 204 MB and time 1 min 1 sec
  - query 3 with active_studentblabla: 6,5 sec
  - query 3 without: 1 min 38 sec
- gpa of size 123 MB and time 59 sec
  - query 2/7 with and index: 700 ms
  - query 2/7 with and without index: 600 ms
  - query 2/7 without: 41 sec
- not_passed of size 100 MB and time 58 sec
  - query 2 with and index: 600 ms
  - query 2 with and without index: 860 ms
  - query 2 without: 50 sec

Indices:

- idx_not_passed of size 62 MB and time 2 sec

1. We chose an index on XYZ, because …. The expected gain in our workload is …. Before the index, queries A and B and C took X minutes, whereas now they take Y minutes

2. We chose …

3. The total space used for the indices and views was this much. The total time required to construct them was that much.

```
               objectname                | objecttype |   entries    |    size
-----------------------------------------+------------+--------------+------------
 courseregistrations                     | r          | 8.00017e+07  | 3345 MB
 with_grade                              | m          | 7.59987e+07  | 3209 MB
 idx_with_grade                          | i          | 7.59987e+07  | 1628 MB
 students                                | r          | 3.99999e+06  | 386 MB
 studentregistrationstodegrees           | r          | 7.99994e+06  | 338 MB
 active_students_per_degree              | m          |  5.8977e+06  | 204 MB
 studentregistrationstodegrees_pkey      | i          | 7.99994e+06  | 171 MB
 gpa                                     | m          |   2.903e+06  | 123 MB
 not_passed                              | m          | 2.90333e+06  | 100 MB
 active_students                         | m          | 2.62499e+06  | 91 MB
 students_pkey                           | i          | 3.99999e+06  | 86 MB
 idx_not_passed                          | i          | 2.90333e+06  | 62 MB
 idx_gpa                                 | i          |   2.903e+06  | 62 MB
 studentassistants                       | r          |      800000  | 28 MB
 courseoffers                            | r          |      400000  | 17 MB
 teacherassignmentstocourses             | r          |      400000  | 14 MB
 courseoffers_pkey                       | i          |      400000  | 8800 kB
```

```
courses                            | r         |       40000 | 4288 kB
teachers                           | r         |       40000 | 3976 kB
teachers_pkey                          | i         |       40000 | 904 kB
courses_pkey                           | i         |       40000 | 904 kB
degrees                            | r         |        8000 | 816 kB
```

https://dba.stackexchange.com/questions/96534/postgres-check-disk-space-taken-by-materialized-view