Stage 3 Revisions:

In stage 3, we evaluated the performance of different indices on our data. For query 1, we initially added an index on Activities(calories_burned) as we were taking the sum of all calories burned and therefore accessing this data numerous times. We then added an index on Goals(calories_goal) because we were grouping by this attribute for each user. We then tried only having an index on calories_goal as well.

Having an index on both attributes was the most time efficient; however, this requires large memory usage. Given that the time speedup was marginal, we felt that the cost of needing more memory was a greater expense than the benefit we were getting from the speedup.

For query 2, we added an index on the Foods(ServSize) because if this value was null, then we would need to look at Foods(Quantity). Because of this double-check, we decided to place an index on ServSize. We then tried only having an index on Goals(calories_goal) because we needed to search for only “Daily” goals. Finally, we tried having an index on Activities(calories_burned) since we are summing this value.

We noticed that the performance did not improve much with the first two indices when comparing time and space. The third index was significantly faster than having no index, and given how frequently we access the calories_burned attribute in other queries, it is practical to have that indexed. Additionally, we can try placing an index on ServSize and Quantity and see if that results in better performance as well.

Revised advanced query for a leaderboard:

```
SELECT user_id, SUM(calories_burned)
FROM Goals LEFT JOIN Activities USING (user_id)
WHERE Goals.timeline LIKE "%Daily"
GROUP BY user_id
ORDER BY SUM(calories_burned)
```