DJ Wiley
A01099945
CS5800 More-SQLing

Query 1

1. Baseline Cost = 2753.62
2. I optimized this query by creating an index on batting(h). My cost became significantly lower.
3. Optimized Cost = 0.40

Query 2

1. Baseline Cost = 2999.76
2. I tried to optimize this query by creating an index on year id and age. The cost didn’t change at all. I don’t think this query can be optimized any further. The query itself is already pretty fast and has a low cost. Because the query still has to list the min from every year, the query couldn’t be made any faster by trying to create an index
3. Optimized Cost = 2999.76

Query 3

1. Baseline Cost = 3579.87
2. I tried to optimize this query by creating an index on master(masterID). The cost didn’t change at all. This leads me to believe that this query cannot be optimized any further. This query is also a very short query and I couldn’t think of any other way to reduce the cost besides trying to index something.
3. Optimized Cost = 3579.87

Query 4

1. Baseline Cost = 2276.39
2. I optimized this query by matching up the masterID
3. Optimized Cost = blah

Query 5

1. Baseline Cost = 694.66
2. I was able to optimize this query by grouping the i’s and distinct variables.
3. Optimized Cost = blah

Query 6

1. Baseline Cost = 584.54
2. For this query I tried creating an index for birth state and birth country. Creating this index did not speed up the query at all. The cost remained the same.
3. Optimized Cost = 584.54

Query 7

1. Baseline Cost = 5387.77
2. This query was optimized by using UNION ALL.
3. Optimized Cost = 3123.51

Query 8

1. Baseline Cost = 3595.92
2. For this query I noticed that the same from statement was used for three different years. So I rewrote that same part of the query so that it would only be written out once. After the rewrite I did 3 joins, once on master, two to itself. This sped up the query noticeably.
3. Optimized Cost = 1078.64

Query 9

1. Baseline Cost = 5324.65
2. For this query I created on index on jeter. This sped things up immensely. I thought this would be a good idea because the entire query is about the different stats on jeter. By creating an index it would create a shortcut for the query.
3. Optimized Cost = 453.04

Query 10

1. Baseline Cost = 105581.21
2. For this query I created an index on the teams(yearID). After the index was created the query cost became much more efficient.
3. Optimized Cost = 23917.46