**Advanced Databases**

**Cassandra CA**

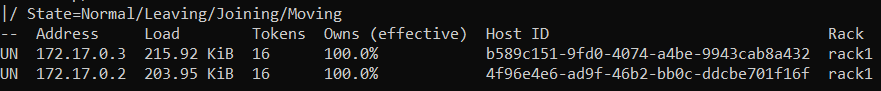
**Student Number:** D20125299

**Student Name:** Luke Hallinan

**Programme Code:** TU856

1. Setting up the cluster and keyspace

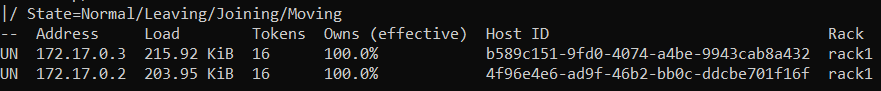
We have a cluster with two nodes.

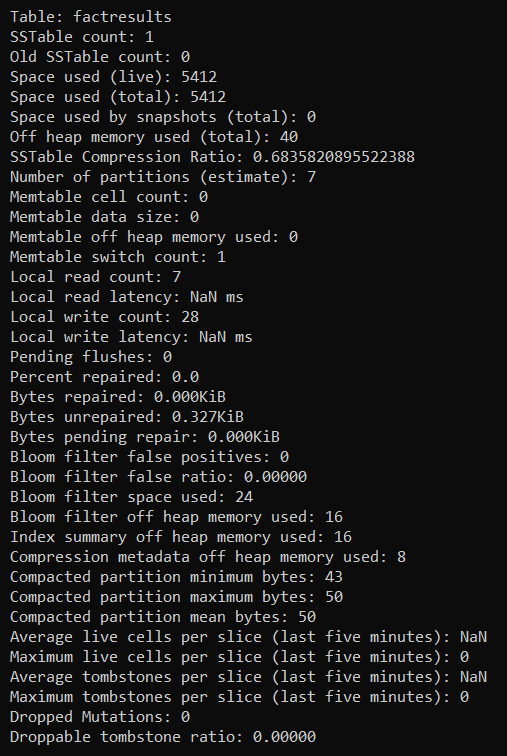


Both nodes are up and functioning normally. Very little load on either only about 200 Kib and each has a unique host ID should It be needed

We have made a keyspace with a simple strategy that has 2 copies for backup and restore purposes.

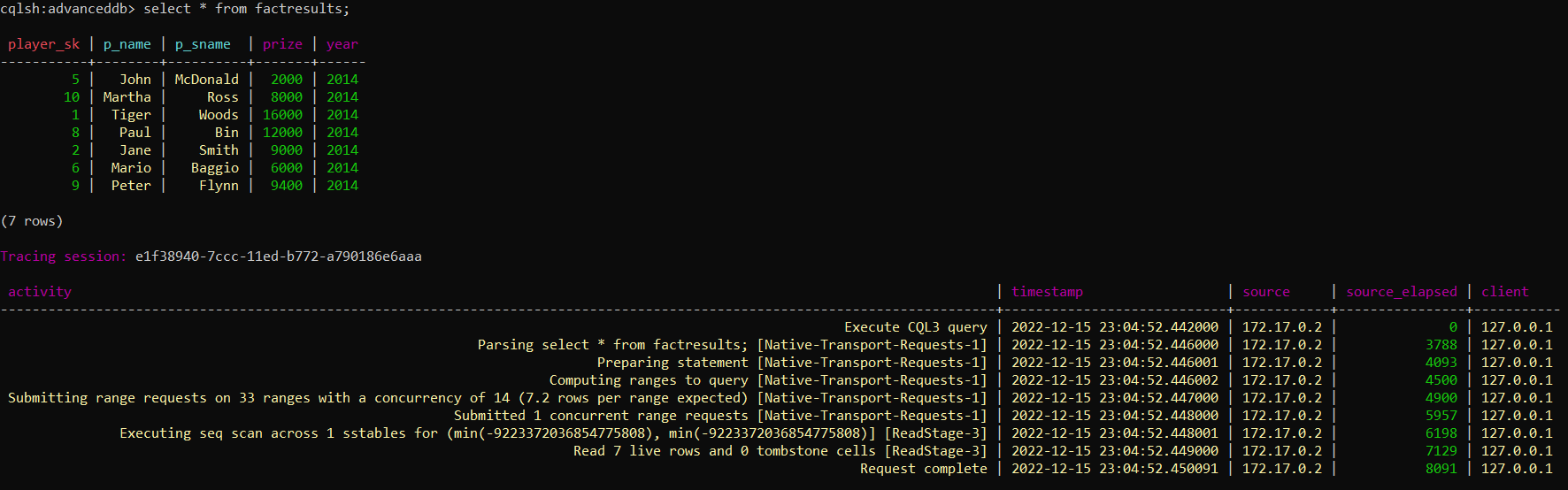
1. Porting the data to Cassandra





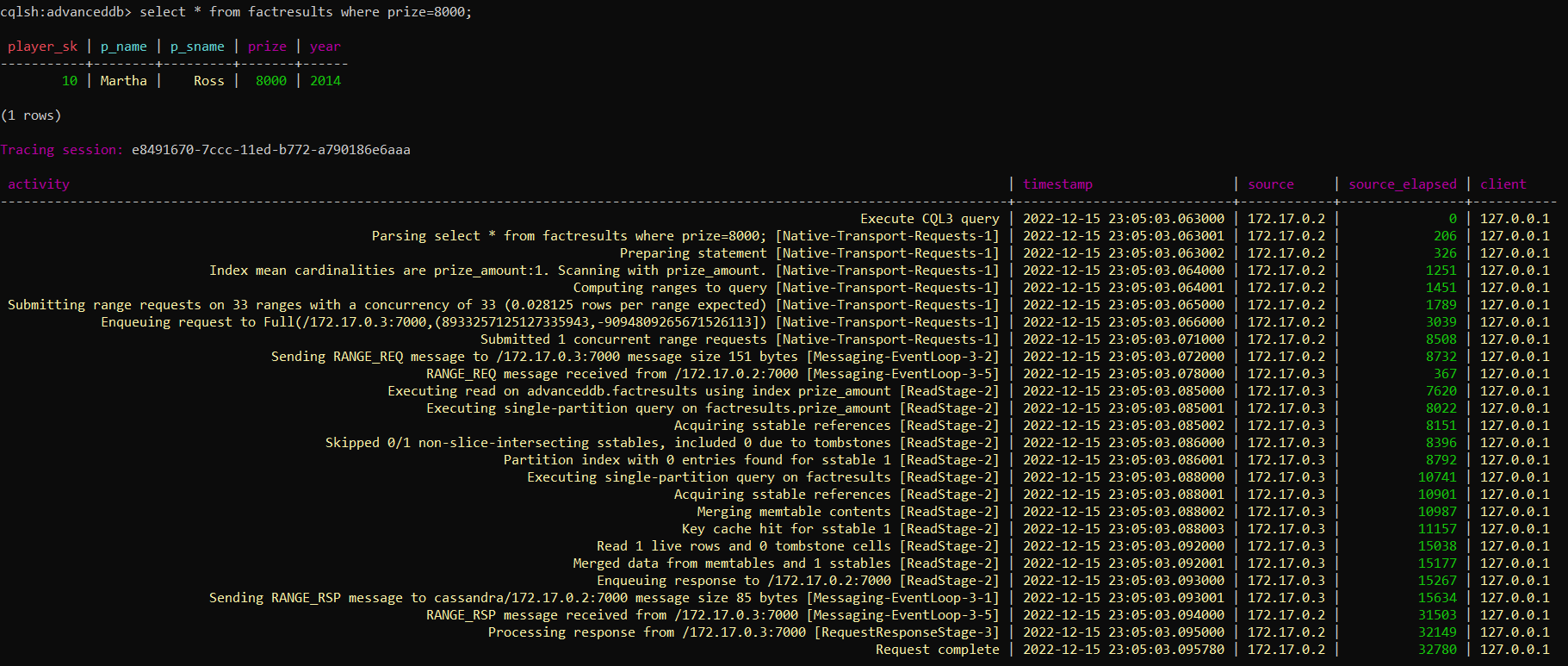
Data shows that there are 7 partitions and the read/write count. Also shoes the space used by the tables.

1. Golf data
   1. Basic query on golf data



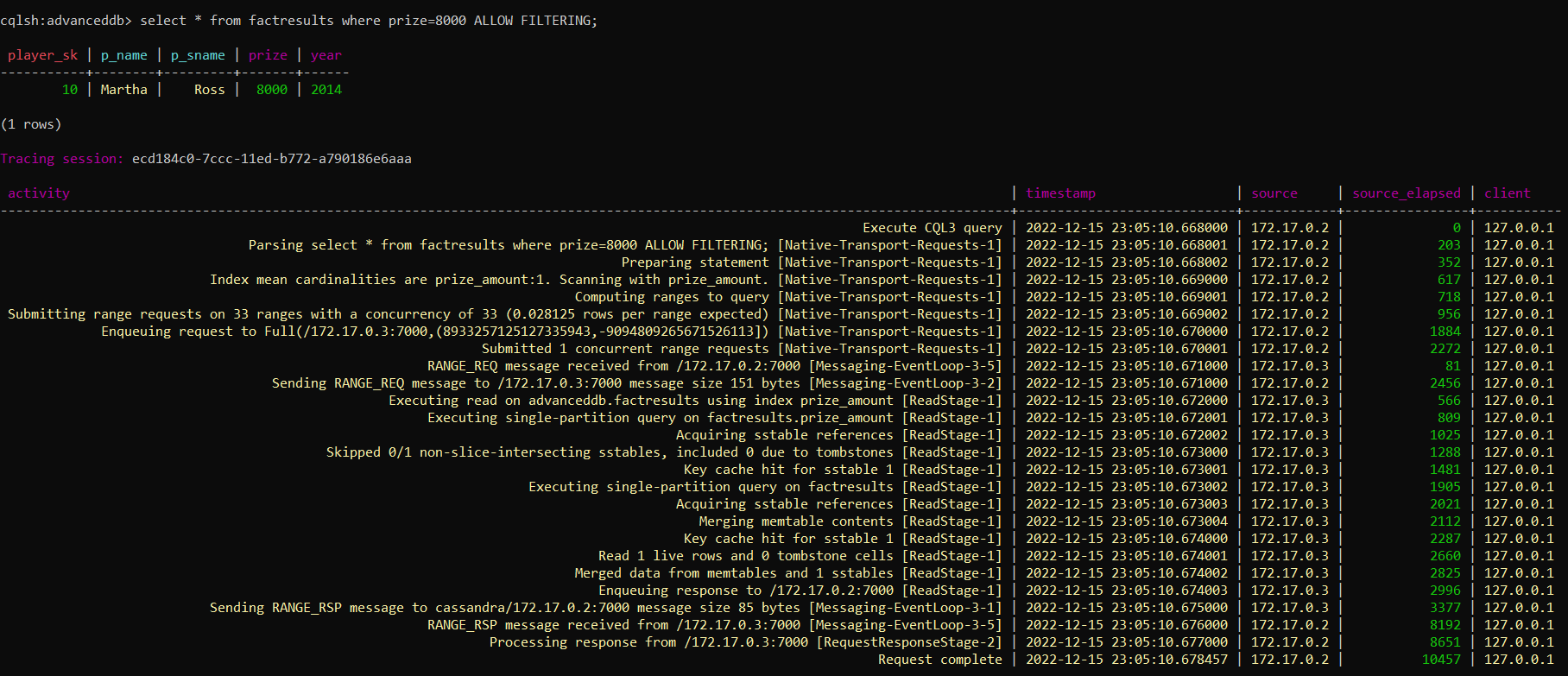
Getting the whole tables took 8091 ms

* 1. Query of non-primary index (without index)



Much longer to execute, 32780 ms. Checks all 33 ranges rather then just the prize column.

* 1. Adding a secondary index to golf data



Much faster then before the index was added only 10457 ms.

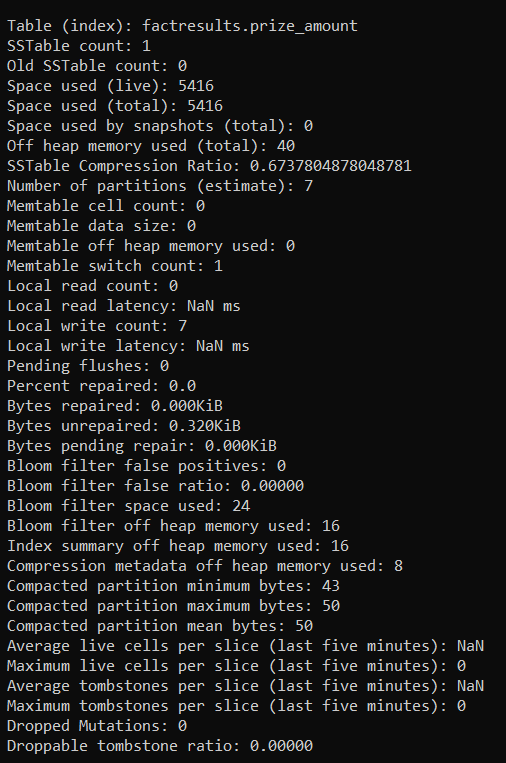
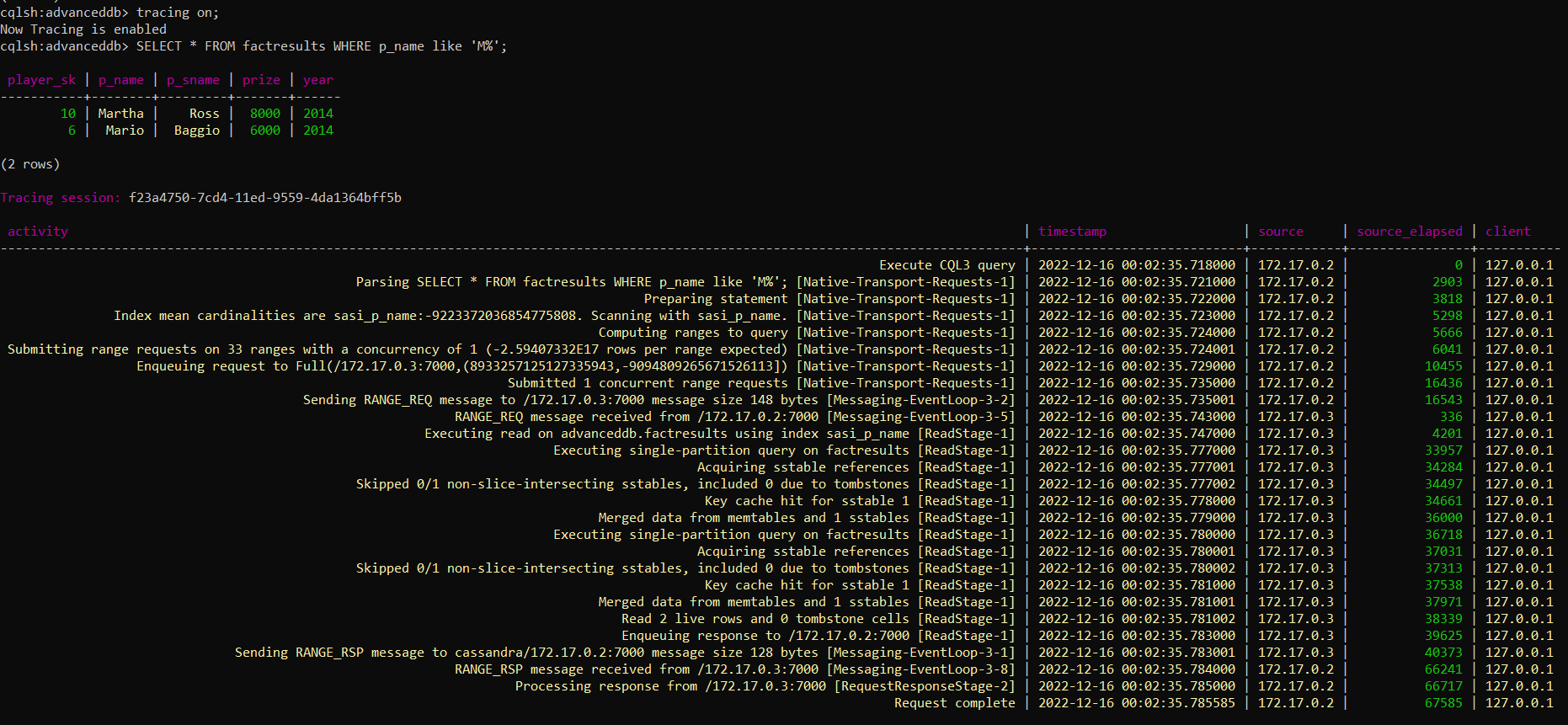


Table index now shows in stablestats. Takes up 5416 bytes so more space is needed for each index the database has.

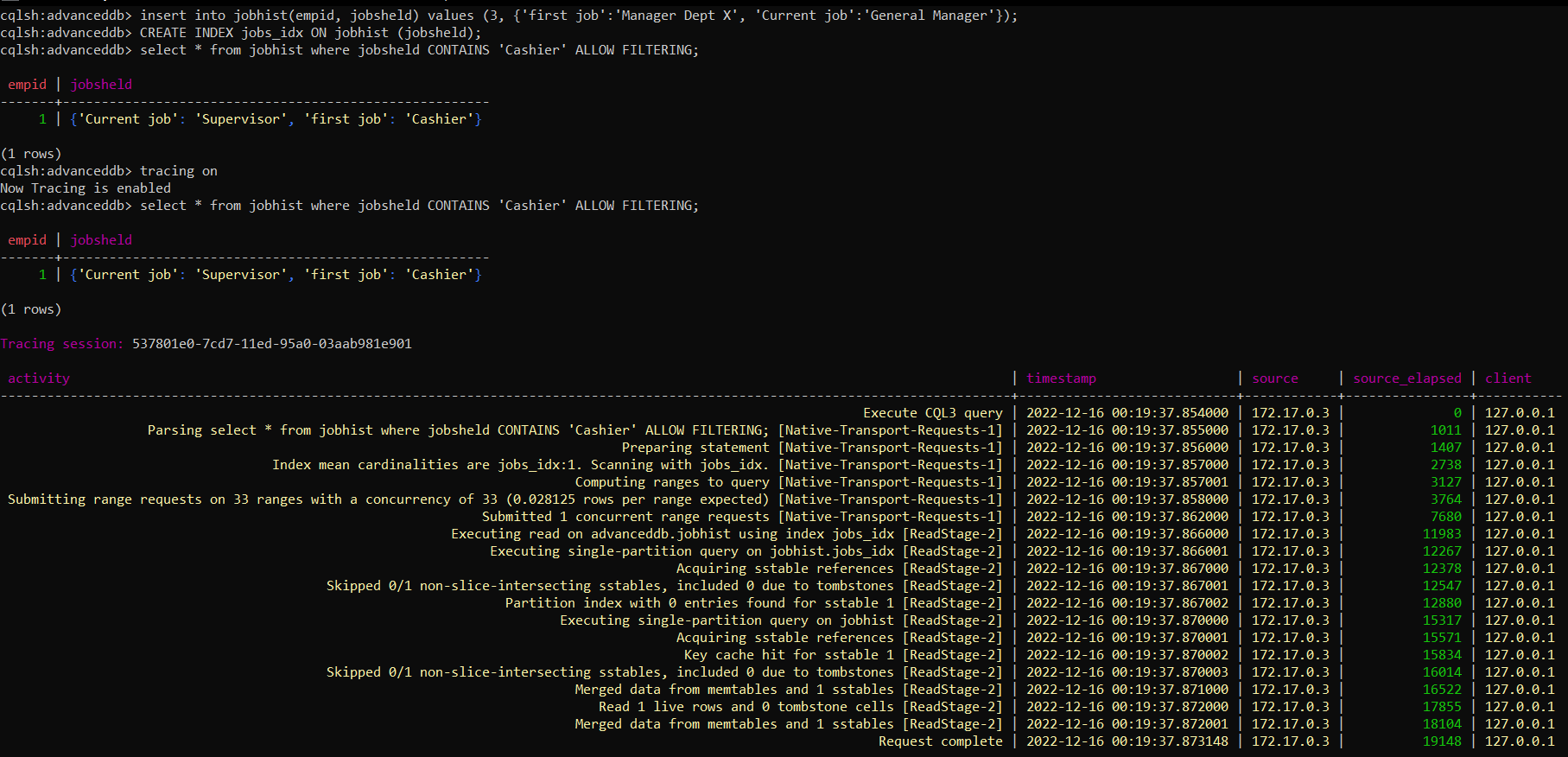
* 1. Adding an SASI index to golf data to facilitate pattern matching



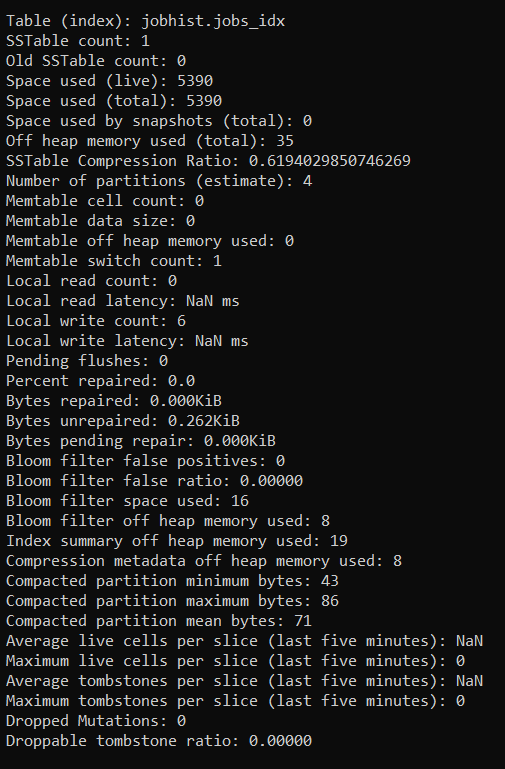
This has allowed us to fins everyone with a name starting with M which would otherwise have been very difficult. It does however take significant amounts of time compared to other searches at 67585ms.

This method does not create a new table index like the normal manual indexing does. This saves on space but still allows like searches.

1. Data including collection data type



This allows us to search more effectively with only 19148ms for the full result.



This method also add a new index table which takes up more space. For a small table like this the index takes up almost as much space as it though it does improve performance.