# Assignment 2 Cassandra Document (ReadMe)

## Harry O’Donnell C20379081

1. **Setup and populate a Cassandra cluster implementing replication and partitioning**

CQL to create the KEYSPACE with replication factor of 3.

**A black screen with white text

Description automatically generated**

CQL to create the table with portioning and clustering.

The county\_id is the partitioning key and the fact\_key is the clustering key.

**A computer screen with white text

Description automatically generated**

**A screenshot of a computer

Description automatically generated**I exported the data from Cassandra into a Json file so that it can be loaded (collection\_data.json), there’s also a CSV of the data as per the submission.

1. **Implement and tune the performance of query against this data using indexes.**

****

Based on the output of this query before and after the index was implemented with tracing enabled, the performance has vastly improved in terms of latency and the number of row read is also vastly decreased. The index enables a single partition query which is the main reason for the vast increase in speed.

The scan before indexing took approx. 13ms seconds but after indexing it only took 4ms and the rows per range searched was reduced from 10933.493 to 0.6375.

**A screenshot of a computer

Description automatically generatedBefore Indexing**

**A screenshot of a computer

Description automatically generatedAfter Indexing**

1. **Create a new table including a collection datatype.**

**A black and white screen

Description automatically generatedA black screen with green text

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

1. **Implement and tune the performance of materialized view against this table using indexes and where clause.**

**A screenshot of a computer

Description automatically generatedA screen shot of a computer code

Description automatically generated**A screenshot of a computer

Description automatically generatedThe primary key is made up of the partition key and the clustering key. When the WHERE clause includes the partition key, it can quickly locate which partition the data belongs to. If it also includes the Clustering key, the search can be even faster as it can narrow down the data within the partition.